UNIVERSITY OF CALIFORNIA SAN DIEGO

Computational and Statistical Complexity of Learning in Sequential Models

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Gaurav Mahajan

Committee in charge:

Professor Sanjoy Dasgupta, Co-Chair
Professor Shachar Lovett, Co-Chair
Professor Kamalika Chaudhuri
Professor Daniel Kane
Professor Arya Mazumdar

2023

The Dissertation of Gaurav Mahajan is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

DEDICATION

To my wife, who constantly supported my desire to work on challenging problems
in spite of how much she hated me working on them

TABLE OF CONTENTS

Dissertation Approval Page iii

Dedication iv

Table of Contents v

List of Figures vii

List of Tables viii

Acknowledgements ix

Vita xi

Abstract of the Dissertation xiii

Chapter 1 Introduction 1
1.1 Our results 2
1.1.1 Computational-statistical gaps in reinforcement learning 2
1.1.2 Computationally efficient algorithms for learning HMMs 3
1.1.3 Understanding algorithms in practice 4

Chapter 2 Computational-statistical gaps in reinforcement learning 6
2.1 Preliminaries 6
2.1.1 Markov Decision Process (MDP) 6
2.1.2 Computational problems 7
2.2 Our results 8
2.3 Proof of the main result 10
2.3.1 From 3-CNF formulas to 3-action MDPs 12
2.3.2 From RL algorithms to 3-SAT algorithms 15
2.3.3 Setting of Parameters 19

Chapter 3 Computationally efficient algorithms for learning HMMs 21
3.1 Preliminaries 21
3.1.1 Hidden Markov Models and low rank distributions 22
3.1.2 Learning models 23
3.2 Our results 24
3.3 Technical overview 26
3.3.1 Background: Observable operators and hard instances 26
3.3.2 Efficient representation 27
3.3.3 Error propagation 30
3.3.4 Estimating operators 32
3.3.5 Finding the basis 34

3.4 Learning with conditional probabilities 36
3.4.1 Algorithm 40
3.4.2 Analysis 40
3.5 Learning with conditional samples 43
3.5.1 Algorithm 44
3.5.2 Analysis 45
3.6 Discussion 47

Chapter 4 Understanding algorithms in practice 49
4.1 Preliminaries 49
4.2 Our results 54
4.3 Related work 59
4.4 Warmup: Constrained tabular parameterization 63
4.4.1 Gradient domination 64
4.4.2 Convergence rates for projected gradient ascent 66
4.4.3 Lower bound: Vanishing gradients and saddle points 68
4.5 Softmax tabular parameterization 70
4.5.1 Asymptotic convergence, without regularization 70
4.5.2 Polynomial convergence with log barrier regularization 72
4.5.3 Dimension-free convergence of Natural Policy Gradient Ascent 76
4.6 Discussion 80

Bibliography 83

LIST OF FIGURES

Figure 2.1. Example construction of 3-action MDP M_φ from a 3-CNF formula $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_3 \vee \bar{x}_3 \vee \bar{x}_3) \wedge (x_1 \vee x_1 \vee x_1)$. The only satisfying assignment for this formula is $(1,1,-1,1)$. 11

Figure 3.1. Schematic of the circulant structure relating the $\Pr[F_t|H_t]$ and $\Pr[F_{t+1}|H_{t+1}]$ matrices. Columns of $\Pr[F_t \mid H_t]$ can be represented linearly in basis B_t using coefficients $\beta(\cdot)$. The blocks $\Pr[oF_{t+1} \mid B_t]$ appear in the next matrix $\Pr[F_{t+1} \mid H_{t+1}]$ (up to scaling), so they can be represented in basis B_{t+1}. . . 28

Figure 4.1. (Non-concavity example) A deterministic MDP corresponding to Lemma 24 where $V^{\pi_\theta}(s)$ is not concave. Numbers on arrows represent the rewards for each action. 52

Figure 4.2. (Vanishing gradient example) A deterministic, chain MDP of length $H+2$. We consider a policy where $\pi(a|s_i) = \theta_{s_i,a}$ for $i = 1,2,\dots,H$. Rewards are 0 everywhere other than $r(s_{H+1},a_1) = 1$. See Proposition 28. 52

LIST OF TABLES

Table 4.1. **Iteration Complexities with Exact Gradients for the Tabular Case:** A summary of the number of iterations required by different algorithms to find a policy π such that $V^\star(s_0) - V^\pi(s_0) \leq \varepsilon$ for some fixed s_0, assuming access to *exact policy gradients*. 54

Table 4.2. **Overview of Approximate Methods:** The suboptimality, $V^\star(s_0) - V^\pi(s_0)$, after T iterations for various approximate algorithms, which use different notions of approximation error (sample complexities are not directly considered but instead may be thought of as part of ε_1 and $\varepsilon_{\text{stat}}$). 57

ACKNOWLEDGEMENTS

First and foremost, I am immensely grateful to my advisors, Shachar Lovett and Sanjoy Dasgupta. I started working with both of them in the middle of my PhD (Winter 2020). Even though, changing areas this late into my PhD could have been stressful, it in fact turned out to be the most fulfilling period of my research life.

Shachar introduced me to many interesting problems, gave me many enlightening ideas and even guided me on problems he initially had no interest in. Meetings with him always filled me with hope. This allowed me to work on interesting problems at the boundary of ones I could possibly solve and constantly push this boundary.

Sanjoy has always been a great friend to me. I remember in my early PhD years (when I was advisor-less), sitting outside a classroom contemplating my life, and Sanjoy offering to hear my thoughts. I can not imagine anyone else being so humble and welcoming. He has always had this faith in me, and my best interests have always been his primary concern. Sanjoy taught me how to find new interesting problems and techniques to solve them. Research with Sanjoy has been a lesson in looking at the bigger picture of life and research.

Outside of research, both Sanjoy and Shachar, always gave me great lessons: where I should go for postdoc, where to hike in Zion National Park, how to lead a happy family life, and so on. I will forever be grateful to them for their generosity.

Other than my advisors, Daniel Kane, Sham Kakade and Jason Lee have been great collaborators and mentors. Their ability to solve problems continue to amaze me and a lot of techniques used in my research, were developed in discussions with them and my advisors. Also, I am grateful to Kamalika Chaudhuri and Arya Mazumdar for reading my thesis and serving on my committee. Arya especially has been encouraging, and my only regret is not being able to find a problem in coding theory to work with him. Hopefully this will change in the future.

I also thank my student collaborators at UCSD: Robi Bhattacharjee, Max Hopkins, Geelon So and Sihan Lui. Robi is especially brilliant at solving problems and I enjoyed discussing problems with him. I thank him for showing me how being laid-back is an option. Max is meticulous and was the driving force in all the work I did with him. He rewrote and made beautiful most of the early writing I did with him and in general did most of the writing for our works. I thank him for all this hard work and constant flow of good problems to work on. Geelon is a budding mathematician, great at working on really hard problems. I thank him for not giving up and teaching me about Stochastic Processes.

Thank you to all my collaborators and supporters- especially Akshay Krishnamurthy, Jason Lee, Simon Du, Ruosong Wang, Wen Sun, Cyril Zhang, Gellért Weisz, Alekh Agarwal and Csaba Szepesvári.

Thank you to all my friends during graduate school - especially Matt Zhang, Aditi Mavalankar, Geelon So, Rex Lei, Sophia Sun, Mark Schultz, Jessica Sorrell, Ken Hoover, Sankeerth Rao and Marco Carmosino. Your friendship has made my PhD a fantastic experience.

Most importantly, I would like to thank my wife: Nirjhar Kabery. Your love and support has made this thesis possible. I am truly lucky to have found someone with such tremendous amount of honesty and trust. Walking with you has been the constant source of happiness. I dedicate this thesis to you, Kabery.

Chapter 2 contains a reprint of the material as it appears in Conference on Learning Theory (COLT 2022). Daniel Kane, Sihan Liu, Shachar Lovett, Gaurav Mahajan. *Computational-statistical gaps in reinforcement learning*. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part is currently being prepared for submission for publication of the material. Sham M. Kakade, Akshay Krishnamurthy, Gaurav Mahajan and Cyril Zhang. *Learning Hidden Markov Models Using Conditional Samples*. The dissertation author was the primary investigator and author of this material.

Chapter 4 contains a reprint of the material as it appears in Conference on Learning Theory (COLT 2020). Alekh Agarwal, Sham M. Kakade, Jason D. Lee, Gaurav Mahajan. *Optimality and approximation with policy gradient methods in markov decision processes*. The dissertation author was the primary investigator and author of this paper.

VITA

2013	BS in Mathematics, Indian Institute of Technology Delhi
2023	PhD in Computer Science, University of California San Diego

PUBLICATIONS

- Daniel M. Kane, Sihan Liu, Shachar Lovett, Gaurav Mahajan, Csaba Szepesvári and Gellért Weisz. *Exponential Hardness of Reinforcement Learning with Linear Function Approximation*. Preprint [57]

- Sham M. Kakade, Akshay Krishnamurthy, Gaurav Mahajan and Cyril Zhang. *Learning Hidden Markov Models Using Conditional Samples*. Preprint [55]

- Max Hopkins, Daniel M. Kane, Shachar Lovett and Gaurav Mahajan. *Do PAC-Learners Learn the Marginal Distribution?*. Preprint [45]

- Max Hopkins, Daniel M. Kane, Shachar Lovett and Gaurav Mahajan. *Realizable learning is all you need*. Conference on Learning Theory (COLT 2022) [43]

- Geelon So, Gaurav Mahajan and Sanjoy Dasgupta. *Convergence of online k-means*. International Conference on Artificial Intelligence and Statistics (AISTATS 2022) [89]

- Daniel M. Kane, Sihan Liu, Shachar Lovett and Gaurav Mahajan. *Computational-statistical gaps in reinforcement learning*. Conference on Learning Theory (COLT 2022) [56]

- Robi Bhattacharjee and Gaurav Mahajan. *Learning what to remember*. International Conference on Algorithmic Learning Theory (ALT 2022) [20]

- Simon S. Du, Sham M. Kakade, Jason D. Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun and Ruosong Wang. *Bilinear classes: A structural framework for provable generalization in rl*. International Conference on Machine Learning (ICML 2021) [31]

- Alekh Agarwal, Sham M. Kakade, Jason D. Lee and Gaurav Mahajan. *On the theory of policy gradient methods: Optimality, approximation, and distribution shift*. Journal of Machine Learning Research (JMLR 2021) [6]

- Max Hopkins, Daniel M. Kane, Shachar Lovett and Gaurav Mahajan. *Noise-tolerant, reliable active classification with comparison queries*. Conference on Learning Theory (COLT 2020) [42]

- Alekh Agarwal, Sham M. Kakade, Jason D Lee and Gaurav Mahajan. *Optimality and approximation with policy gradient methods in markov decision processes*. Conference on Learning Theory (COLT 2020) [5]

- Max Hopkins, Daniel M. Kane, Shachar Lovett and Gaurav Mahajan. *Point location and active learning: Learning halfspaces almost optimally*. 61st Annual Symposium on Foundations of Computer Science (FOCS 2020) [44]

- Simon S. Du, Jason D. Lee, Gaurav Mahajan and Ruosong Wang: *Agnostic -learning with Function Approximation in Deterministic Systems*. Advances in Neural Information Processing Systems (NeurIPS 2020) [32]

ABSTRACT OF THE DISSERTATION

Computational and Statistical Complexity of Learning in Sequential Models

by

Gaurav Mahajan

Doctor of Philosophy in Computer Science

University of California San Diego, 2023

Professor Sanjoy Dasgupta, Co-Chair
Professor Shachar Lovett, Co-Chair

Recent success of machine learning is driven by scaling laws: larger architectures trained using more data and compute lead to more "intelligent" agents. Therefore, even minor enhancements to the sample and compute complexity of these algorithms can have significant scientific and financial implications. In this dissertation, we study these question in the context of sequential models. In particular, we study the following questions.

- **Computational-statistical gaps in reinforcement learning.** In this part, we study the computational and statistical complexity of sequential decision-making under the framework of reinforcement learning. A fundamental assumption in theory of reinforcement

learning is ”RL with linear function approximation”. Under this assumption, the optimal value function (either Q*, or V*, or both) can be obtained as the linear combination of finitely many known basis functions. Even though it was observed as early as 1963 that there are empirical benefits of using linear function approximation, only recently a series of work designed sample efficient algorithms for this setting. These works posed an important open problem: *Can we design polynomial time algorithms for this setting?* Here, we show progress on this open problem by proving: unless NP=RP, no polynomial time algorithm exists for this settings.

- **Computationally efficient algorithms for learning HMMs.** In this part, we study the computational complexity of learning structured distributions over sequences of observations (e.g. DNA sequences, proteins, spoken words and so on). In particular, we are concerned with the computational complexity of learning Hidden Markov Model (HMM). Although HMMs are some of the most widely used tools in sequential and time series modeling, they are cryptographically hard to learn in the standard setting where one has access to i.i.d. samples of observation sequences. Here, we show a positive result: computationally efficient algorithm for learning HMMs when the learner has access to conditional samples from the target distribution. We also show that these results extend to “low rank” distributions.

- **Understanding algorithms in practice.** In this part, we study the most commonly used algorithms for sequence decision-making in practice: policy gradient methods. Even though these algorithms are simple to implement, their convergence properties are only established at a relatively coarse level; in particular, the folklore guarantee is that these methods converge to a stationary point of the objective. Here, we present the first global convergence results for policy gradient methods like vanilla policy gradient (w/wo regularization) and natural policy gradient.

Chapter 1

Introduction

Recent years have seen empirical success of simple gradient based algorithms in complex sequential tasks, ranging from playing games like Chess, Go to more serious endeavors like robotics, stratospheric flight, conversational AI, etc. Theoretically, this was surprising since we expect the worst case data and compute requirement for such algorithms to be unnaturally large (for example, according to classical theory, Chess should require $> 2^{100}$ samples and compute.). Then, a natural question arises: What properties of these environments allows these simple algorithms to escape worst case scenarios?

This question is theoretically enticing as it requires understanding how the complexity of this problem depends on its structure (similar attempts like semi-random models [22] in graph theory have been very fruitful). But this question also has important practical implications. As we have become more ambitious in our goals, the data requirements for existing algorithms has become exceedingly high, preventing us from automating higher cognition tasks. For example, OpenAI Five, a bot for a collaborative game DOTA2, was trained for almost 10 months in real time. Moreover, many real world tasks require data generation via complex interaction with humans or interaction with expensive hardware. As a result, we have not seen much success in these domains (e.g. healthcare, self-driving cars). *My research goal is to investigate if such data requirements are fundamental or can we design efficient algorithms for these applications?*

1.1 Our results

1.1.1 Computational-statistical gaps in reinforcement learning

We first study this question from the perspective of sequential decision-making under the framework of reinforcement learning. There is a growing interest in reinforcement learning theory community, to design and analyze efficient algorithms for the large state space regime. In this regime, the goal is to design algorithms whose complexity does not polynomially depend on the size of the state space. Since, this is impossible when we do not make any assumptions about the environment, much effort has been spent on finding minimal assumptions under which an optimal policy can be found efficiently: State Aggregation [65, 30], Linear q^{π} [33, 63, 102, 96], Linear MDPs [101, 52], Linear Mixture MDPs [69, 12, 104], Reactive POMDPs [62], Block MDPs [34], FLAMBE [4], Reactive PSRs [66], Linear Bellman Complete [72, 103], Bellman rank [48], Witness rank [90], Bilinear Classes [31], Bellman Eluder [51] and Decision-Estimation Coefficient [38].

One such minimal assumption that came out of this line of work is RL with linear function approximation: when the optimal value function (either Q^*, or V^*, or both) can be obtained as the linear combination of finitely many, known basis functions. Under this assumption, a series of works [31, 98, 97, 94, 38] showed sample efficient algorithms for constant number of actions. These works leave finding a computationally efficient algorithm for this setting as an important open question.

In a joint work with Daniel Kane, Sihan Liu and Shachar Lovett [56], we make progress on this open problem by showing that under well believed complexity assumptions (NP doesn't have efficient randomized algorithms), no polynomial time algorithm exists for RL with linear function approximation. In a follow-up work [57], we show an almost tight computational lower bound, which is exponential in the number of basis functions and horizon under the Randomized Exponential Time Hypothesis.

There are a couple of implications of these results. First, this shows for the first time a computational-statistical gap in RL, that is a regime where the underlying statistical problem is information theoretically possible, but no computationally efficient algorithm exists.[1] Second, this shows the effect of noise in RL, adding little noise to reward signal turns this problem from linear-time to computationally hard.[2] On a high level, this is very similar to how solving LWE is computationally hard, whereas exact linear equations can be efficiently solved by Gaussian Elimination.

1.1.2 Computationally efficient algorithms for learning HMMs

We next study this question from the perspective of learning distributions over observation sequences. Hidden Markov Models (HMMs) are among the most fundamental tools for modeling temporal and sequential phenomena. These probabilistic models specify a joint distribution over a sequence of observations generated via a Markov chain of latent states. This structure enjoys the simultaneous benefits of low description complexity, sufficient expressivity to capture long-range dependencies, and efficient inference algorithms. For these reasons, HMMs have become ubiquitous building blocks for sequence modeling in varied fields, ranging from bioinformatics to natural language processing to finance. A long-standing challenge, in both theory and practice, is the computational difficulty of learning an unknown HMM in TV distance.

In the standard realizable formulation, we are given observation sequences randomly sampled from an underlying HMM and are asked to efficiently compute a distribution that is close to the HMM in TV distance. Under this formulation, maximum likelihood estimation is known to be statistically efficient, but no computationally efficient implementations of this approach are known. More generally, HMMs can encode the parity with noise problem [70], which is widely believed to be computationally hard [21, 59, 8], and so we do not expect to find efficient algorithms for general HMMs. Recent works have therefore focused on obtain-

[1] This phenomenon is also observed in other problems in cs theory like community detection, planted clique and sparse principal component analysis.

[2] In another work [32], we showed that noiseless version of this problem has a simple (computationally efficient) linear-time algorithm.

ing computationally efficient algorithms under structural assumptions which evade these hard instances [29, 46, 88].

In a joint work with Sham Kakade, Akshay Krishnamurthy and Cyril Zhang [55], we develop new algorithms and techniques for learning Hidden Markov models when provided with conditional samples from HMM. We show how a generalization of Angluin's L^* algorithm can efficiently learn any HMM when the learner can query for *exact* conditional probabilities. We then extend this result to more natural setting, where the learner only has access to samples from the conditional distributions. Here, we obtain an algorithm that is computationally efficient for all HMMs with "high fidelity," a new property we introduce. Our results require a number of new algorithmic ideas and analysis techniques, most notably: an efficient representation for distributions over exponentially large domains and a new perturbation argument for mitigating error amplification over long sequences.

1.1.3 Understanding algorithms in practice

Lastly, we study the most commonly used algorithms for sequence decision-making in practice: policy gradient methods. Policy gradient methods have a long history in the reinforcement learning (RL) literature [99, 91, 61, 53] and are an attractive class of algorithms as they are applicable to any differentiable policy parameterization; admit easy extensions to function approximation; easily incorporate structured state and action spaces; are easy to implement in a simulation based, model-free manner. Owing to their flexibility and generality, there has also been a flurry of improvements and refinements to make these ideas work robustly with deep neural network based approaches (see e.g. [84, 85]).

Despite the large body of empirical work around these methods, their convergence properties are only established at a relatively coarse level; in particular, the folklore guarantee is that these methods converge to a stationary point of the objective, assuming adequate smoothness properties hold and assuming either exact or unbiased estimates of a gradient can be obtained (with appropriate regularity conditions on the variance). However, this local convergence viewpoint does not address some of the most basic theoretical convergence questions, including: 1) if and how fast they converge to a globally optimal solution (say with a sufficiently rich policy class); 2) how they cope with approximation error due to using a restricted class of parametric policies; or 3) their finite sample behavior. These questions are the focus of this work.

In a joint work with Alekh Agarwal, Sham Kakade and Jason Lee, we analyze typical variants of policy gradient methods and show that just like supervised learning, the non-convexity of the policy optimization problem is not the fundamental challenge for policy gradient approach. We answer: 1) if and how fast they converge to a globally optimal solution; and 2) how they cope with approximation error due to using a restricted class of parametric policies. Overall, the results of this work place policy gradient methods under a solid theoretical footing, analogous to the global convergence guarantees of iterative value function based algorithms.

Chapter 2

Computational-statistical gaps in reinforcement learning

2.1 Preliminaries

2.1.1 Markov Decision Process (MDP)

We first define the framework for reinforcement learning, a Markov Decision Process (MDP). We define a deterministic MDP as a tuple $M = (\mathscr{S}, \mathscr{A}, R, P)$, where $\mathscr{S}$ is the state space, $\mathscr{A}$ is the action space, $R : \mathscr{S} \times \mathscr{A} \mapsto \Delta([0,1])$ is the stochastic reward function [1], and $P : \mathscr{S} \times \mathscr{A} \mapsto \mathscr{S}$ is the deterministic transition function. An MDP M defines a discrete time sequential decision process where the agent starts from a starting state $s_0 \in \mathscr{S}$. Then, at each time t, the agent at some current state S_t, takes action A_t, receiving reward $R_t \sim R(S_t, A_t)$ and transitions to next state S_{t+1}. This goes on till the agent reaches the end state $\perp$. Each such trajectory/path from starting state s_0 to end state $\perp$ is of length at most horizon H. A deterministic, stationary policy $\pi : \mathscr{S} \mapsto \mathscr{A}$ specifies a decision-making strategy in which the agent chooses actions adaptively based on the current state, i.e. $A_t = \pi(S_t)$. Given a policy π and a state-action pair $(s,a) \in \mathscr{S} \times \mathscr{A}$, the Q-function and V-function under a policy π are defined as

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\tau-1} R(S_t, A_t) \mid S_0 = s, \pi\right], \quad Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\tau-1} R(S_t, A_t) \mid S_0 = s, A_0 = a, \pi\right], \tag{2.1}$$

[1] $\Delta([0,1])$ denotes the set of all distributions over interval $[0,1]$.

where $S_1, A_1, \dots S_{\tau-1}, A_{\tau-1}$ are obtained by executing policy π in the MDP M and τ is the first time when policy π reaches the end state $\perp$, that is $S_\tau = \perp$ where it always holds that $\tau \leq H$. We use Q^* and V^* to denote the optimal value functions

$$V^*(s) = \sup_\pi V^\pi(s)\,, \quad Q^*(s,a) = \sup_\pi Q^\pi(s,a)\,, \quad s \in \mathscr{S}, a \in \mathscr{A}$$

We say that the optimal value functions V^* and Q^* can be written as a linear function of d-dimensional features ψ: $\mathscr{S} \cup (\mathscr{S} \times \mathscr{A}) \to \mathbb{R}^d$ if for all state s and action a, $V^*(s) = \langle \theta, \psi(s) \rangle$ and $Q^*(s,a) = \langle \theta, \psi(s,a) \rangle$ for some fixed $\theta \in \mathbb{R}^d$ independent of s and a.

2.1.2 Computational problems

We next introduce 3-SAT, a satisfiability problem for 3-CNF formulas. In a 3-SAT problem, we are given as input, a 3-CNF formula φ with v variables and $O(v)$ clauses and our goal is to decide if φ is satisfiable. Our computational lower bound is based on a reduction from UNIQUE-3-SAT, a variant of 3-SAT. UNIQUE-3-SAT is the promise version of 3-SAT where the given formula is promised to have either 0 or 1 satisfying assignments.

The focus of this work is the computational RL problem, LINEAR-k-RL. In a LINEAR-k-RL problem with feature dimension d, we are given access to a deterministic MDP M with k actions and horizon $H = O(d)$ such that the optimal value functions Q^* and V^* can be written as a linear function of d-dimensional features ψ. Our goal is to output a good policy, which we define as any policy π that satisfies $V^\pi > V^* - 1/4$. Note that here V^π and V^* refers to the value of the policy π and optimal policy respectively at the starting state and is always in $[0,H]$ [2]. Moreover, the constant $1/4$ can be replaced by any arbitrary constant < 1. From now on, we always assume number of actions k is 2 or 3.

[2] in our constructions, we satisfy the more stringent condition that $V^* \in [0,1]$.

Complexity problem LINEAR-k-RL

Oracle: a deterministic MDP M with k actions, optimal value functions V^* and Q^* linear in d dimensional features ψ and horizon $H = O(d)$.

Goal: find policy π such that $V^\pi > V^* - 1/4$.

We now describe how the algorithm interacts with the MDP. We assume that the algorithm has access to the associated (i) reward function R, (ii) transition function P and (iii) features ψ. For all these functions, the algorithm provides a state s and action a (if needed) and receives a random sample from the distribution $R(s,a)$ (for the reward function), the state $P(s,a)$ (for the transition function) or feature $\psi(s)$ or $\psi(s,a)$ (for the features). We assume that each call accrues constant runtime and input/output for these functions are of size polynomial in feature dimension d.

We will often talk about randomized algorithm A solving a problem in time t with error probability p. By this we mean (i) A runs in time $O(t)$; (ii) for satisfiability problems, it returns YES on positive input instances with probability at least $1-p$ and returns NO on negative input instances with probability 1; and (iii) for RL problem, it returns a good policy with probability at least $1-p$.

2.2 Our results

With these considerations in mind, we present our main result that asserts that unless NP=RP, no randomized polynomial time algorithm can find a good policy in deterministic MDPs with a constant number of actions and linear optimal value functions.

Theorem 1 (LINEAR-3-RL $\in$ RP $\implies$ NP=RP)**.** *Unless* NP=RP*, no randomized algorithm can solve* LINEAR-3-RL *with feature dimension d in time polynomial in d with error probability* $1/10$.

This resolves the open problem from [97] and [31] by showing that unless RP=NP, no polynomial time randomized algorithm exists for deterministic transition MDPs with a constant number of actions and linear optimal value functions.

Our main technical contribution is a reduction from UNIQUE-3-SAT to LINEAR-3-RL such that a polynomial time algorithm for LINEAR-3-RL implies a polynomial time algorithm for UNIQUE-3-SAT. To achieve this, we use the input for UNIQUE-3-SAT: a 3-CNF formula φ with v variables, to design an input for LINEAR-3-RL: an MDP M_φ with 3 actions and optimal value functions V^* and Q^* linear in d-dimensional features. On a high level, the MDP is constructed such that each state represents an assignment to the UNIQUE-3-SAT variables and the goal is to "search" for the solution to the UNIQUE-3-SAT instance. In particular, at each state, the 3 actions available to the agent correspond to an unsatisfied clause which ensures at least one action available to the agent decreases the distance to the solution. To incentivize finding the solution, a large reward is awarded on reaching the solution and a very small expected reward on reaching the horizon (this reward is small enough that any polynomial time RL algorithm only receives 0 reward with high probability on reaching the horizon). This ensures that (i) finding a good policy also finds the satisfying assignment of φ and (ii) the optimal value functions V^* and Q^* are linear in some low dimensional features. We present this construction in Section 2.3.

These reductions allow us to simulate a polynomial time algorithm for UNIQUE-3-SAT on input φ by running the polynomial time algorithm for LINEAR-3-RL on MDP M_φ. More formally, our reduction gives a polynomial relationship between the complexity of UNIQUE-3-SAT and LINEAR-3-RL: a polynomial d^q time algorithm for LINEAR-3-RL implies a polynomial $v^{O(q^2)}$ time algorithm for UNIQUE-3-SAT.

Proposition 1. *Suppose* $q \geq 1$. *If* LINEAR-3-RL *with feature dimension d can be solved in time d^q with error probability* $1/10$*, then* UNIQUE-3-SAT *with v variables can be solved in time $v^{O(q^2)}$ with error probability* $1/8$.

This relates the complexity of UNIQUE-3-SAT to LINEAR-3-RL. To relate these problems to complexity class NP, we use a seminal result from [93] which showed that uniqueness of solution can not be used to solve search problems quickly. In particular, they showed a randomized polynomial time reduction from 3-SAT to UNIQUE-3-SAT.

Theorem 2 (Valiant-Vazirani Theorem)**.** *Unless* NP=RP*, no polynomial time randomized algorithm can solve* UNIQUE-3-SAT *with error probability* $1/8$.

Combining our reduction with Valiant-Vazirani Theorem proves our main result, Theorem 4.

2.3 Proof of the main result

In this section, we will prove Proposition 1. The overall idea is to first build a randomized algorithm $\mathscr{A}_{SAT}$ which can decide UNIQUE-3-SAT using a randomized algorithm $\mathscr{A}_{RL}$ which solves LINEAR-3-RL. The two reductions only differ in their settings of parameters.

In the first setting, which we use to prove that no polynomial time algorithm exists for LINEAR-3-RL, we set the feature dimension d to be polynomial in the number of variables v. Under this setting, we can build a polynomial time randomized algorithm for UNIQUE-3-SAT using a polynomial time randomized algorithm for LINEAR-3-RL.

Proposition 2 (Restatement of Proposition 1)**.** *Suppose* $q \geq 1$*. If* LINEAR-3-RL *with feature dimension* d *can be solved in time* d^q *with error probability* $1/10$*, then* UNIQUE-3-SAT *with* v *variables can be solved in time* $O(v^{8q+16q^2})$ *with error probability* $1/8$.

Before we prove this results, we give a brief outline of our reduction from UNIQUE-3-SAT to LINEAR-3-RL. On a high level, we construct an MDP where the goal is to ”search” for the solution w^* to a UNIQUE-3-SAT instance with v variables. In particular, at each time, the agent is given an unsatisfied clause and asked to flip assignment for a variable present in the clause. Notice that since the clause is unsatisfied, there must be at least one variable whose assignment differs from the solution and therefore, the agent can “reach” the solution in at most $d(w,w^*)$ steps. To incentivize the agent, if the agents at time l finds the solution i.e. $w = w^*$ or

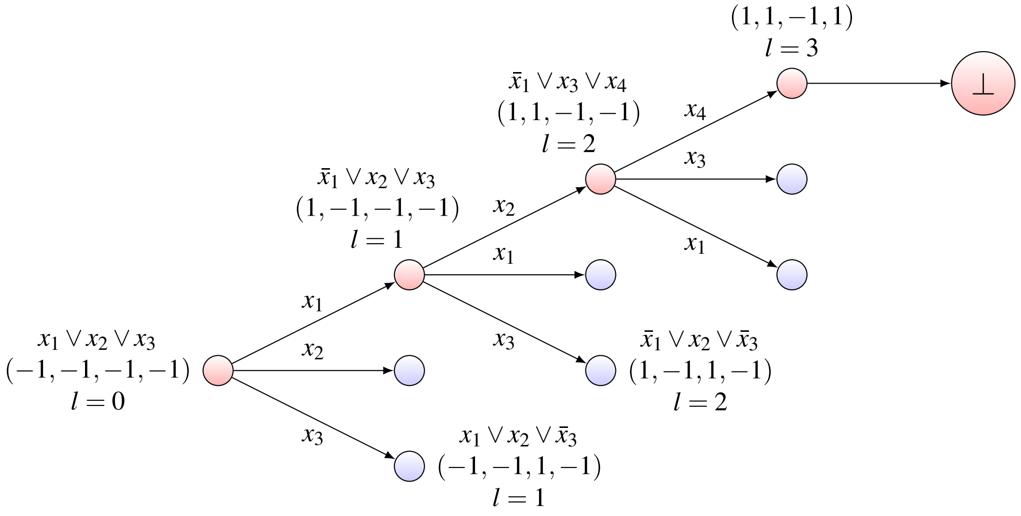

Figure 2.1. Example construction of 3-action MDP M_φ from a 3-CNF formula $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_3 \vee \bar{x}_3 \vee \bar{x}_3) \wedge (x_1 \vee x_1 \vee x_1)$. The only satisfying assignment for this formula is $(1,1,-1,1)$.

reaches the end of the MDP i.e. $l = H$, it receives reward according to the following degree-r polynomial

$$g(l,w) = \left(1 - \frac{l + \mathrm{dist}(w,w^*)}{H+v}\right)^r .$$

We show how to build an MDPs from a UNIQUE-3-SAT instance in Section 2.3.1. Furthermore, we show that the optimal value functions V^* and Q^* for the constructed MDP are linear in $d = O(v^r)$-dimensional features. Since the expected reward at last layer of the MDP is $O(v^{-r^2})$ (which can be replaced with 0 for any poly(d) time RL algorithm), the only non-zero reward is achieved by solving the underlying UNIQUE-3-SAT instance, proving our reduction. We give a formal argument in Section 2.3.2, where we show how to build a randomized algorithm for UNIQUE-3-SAT using a randomized algorithm for LINEAR-3-RL. In Section 2.3.3, we discuss the setting of parameters which will prove Proposition 2.

2.3.1 From 3-CNF formulas to 3-action MDPs

We will start by defining a mapping from an input of UNIQUE-3-SAT problem: 3-CNF formula φ with v variables and $O(v)$ clauses to an MDP M_φ with 3 actions and $H = O(d)$ horizon with optimal value functions linear in d dimensions. Our informal goal is to design an MDP M_φ such that finding a good policy also implies finding the satisfying assignment for the formula φ. We now formally describe the MDP M_φ when the formula φ has a unique satisfying assignment $w^* \in \{-1,1\}^v$ and later show how the MDP M_φ differs when the formula φ has no solution. See Figure 2.1 for an example.

Transitions. In our setting, it will be useful to visualize an MDP as a tree, where nodes represent states and edges represent actions. A policy is then a sequence of actions or equivalently a path in the aforementioned tree. The MDP M_φ is a ternary tree i.e. each state/node in the tree has 3 children. The transitions/dynamics are deterministic i.e. the first action goes to first child, the second action goes to second child and so on.

Assignments. Each state is associated with an assignment to the v variables i.e. a binary vector in $\{-1,1\}^v$ and a natural number l denoting the depth of the state. Our goal here is to choose assignments such that it is always possible to choose an action which decreases the hamming distance to the satisfying assignment. The root in the tree is associated with the all zeroes assignment $(-1,-1,\ldots,-1)$. For any state s with a non-satisfying assignment $w = (w_1, w_2, \ldots, w_v) \neq w^*$, the assignment associated to the three children are as follows. Since w is not a satisfying assignment, consider the first unsatisfied clause with variables $x_{i_1}, x_{i_2}, x_{i_3}$. The first child is associated with the assignment where the i_1-th bit of w is flipped, the second child is associated with vector where i_2-th bit is flipped and so on. More formally, the assignment associated to j-th child is $(w'_1, w'_2, \ldots, w'_v)$ where $w'_k = \neg w_k$ if $k = i_j$ and $w'_k = w_k$ otherwise. The two exceptions to this are (i) states with the satisfying assignment w^* and (ii) states at the last level H. For such states, all actions go to the end state $\perp$.

Rewards. To ensure that finding good policies implies finding the satisfying assignment in our MDP, we will only give rewards when a satisfying assignment is found or at the last layer. More formally, the rewards everywhere are zero except on (i) states with the satisfying assignment w^* and (ii) states on the last level H. In both the cases above, say the state is at level l with assignment w, then the associated reward distribution for any action is a Bernoulli distribution $Ber(g(l,w))$ where

$$g(l,w) = \left(1 - \frac{l + \text{dist}(w,w^*)}{H+v}\right)^r$$

and the Bernoulli distribution $Ber(p)$ is 1 with probability p and 0 with probability $1-p$. Here r is a parameter which we will specify in Section 2.3.3. When the formula φ has no satisfying assignment, all rewards are 0. Note that in our simulation (Section 2.3.2), we don't know/use w^* and instead use an approximate reward function that is easy to compute.

Linear Optimal Value Functions. We next show that in the MDP M_φ, the optimal value functions V^* and Q^* can be written as a linear function of $d = O(v^r)$ dimensional features ψ, where $\psi(s)$ or $\psi(s,a)$ depends only on w, the corresponding assignment, and l, the depth of the state.

Proposition 3. *For any state s in level l with assignment w and action a,*

(i) the optimal value function is $V^*(s) = g(l,w)$.

(ii) for large enough v, there exists features $\psi(s), \psi(s,a) \in \mathbb{R}^d$ with feature dimension $d \leq 2v^r$ depending only on state s and action a; and $\theta \in \mathbb{R}^d$ depending only on w^* such that V^* and Q^* can be written as a linear function of features ψ i.e. $V^*(s) = \langle \theta, \psi(s) \rangle$ and $Q^*(s,a) = \langle \theta, \psi(s,a) \rangle$.

Proof. To prove our first claim, we start by showing that there exists a policy π that achieves this value for each state. Let π be the policy which for any state s with assignment $w \neq w^*$ chooses the action which decreases the hamming distance $\text{dist}(w, w^*)$ by 1. Note that one such action always exists in our construction, since a satisfying assignment satisfies all clauses. Therefore, from a state s at level l with assignment w, we can reach a state with assignment w_1 such that either (i) w_1 is a satisfying assignment or (ii) w_1 is at the last level and on the optimal path from w to w^* i.e. $\text{dist}(w, w^*) = \text{dist}(w, w_1) + \text{dist}(w_1, w^*)$. In both cases,

$$V^{\pi}(s) = \left(1 - \frac{l + \text{dist}(w, w_1) + \text{dist}(w_1, w^*)}{H + v}\right)^r = g(l, w)$$

Next, for any other policy π' that ends on state s' at level l' with assignment w' (i.e. either $l' = H$ or $w' = w^*$), we have

$$V^{\pi'}(s) = \left(1 - \frac{l' + \text{dist}(w', w^*)}{H + v}\right)^r \leq \left(1 - \frac{l + \text{dist}(w, w') + \text{dist}(w', w^*)}{H + v}\right)^r \leq g(l, w)$$

where the first inequality follows from $l' - l \geq \text{dist}(w, w')$. This proves our first claim about V^* i.e. $V^*(s) = g(l, w)$.

To prove our second claim, that V^* and Q^* can be written as a linear function of features ψ, we will show that $V^*(s)$ can be written as a polynomial of degree at most r in w^*. To see why this is enough, we set θ to be all monomials in w^* of degree at most r. That is, each coordinate of θ corresponds to a multiset $S \subset [v]$ of size $|S| \leq r$, and its value is $\theta_S = \prod_{i \in S} w_i^*$. We set $\psi(s)$ to be the corresponding coefficients in the polynomial V^*. Then, we can write $V^*(s) = \langle \theta, \psi(s) \rangle$. Since, there are at most $\sum_{i=0}^{r} v^i \leq 2v^r$ many coefficients we can set the feature dimension as $d = 2v^r$.

Finally, we prove that $V^*(s)$ can be written as a polynomial of degree at most r in w and

w^*. Firstly hamming distance $\mathrm{dist}(w,w^*)$ is linear in both w and w^* i.e.

$$\mathrm{dist}(w,w^*) = \frac{v - \langle w, w^* \rangle}{2}$$

Our claim follows from noting that $g(l,w)$ is a polynomial of degree r in $\mathrm{dist}(w,w^*)$. Note that linear V^* implies linear Q^* in deterministic MDPs for $\psi(s,a) = \psi(P(s,a))$, since by definition, in MDPs with deterministic transition, $Q^*(s,a) = V^*(P(s,a))$. □

Even though $\psi(s)$ does not depend on w^*, unlike the constructions of [95, 94], $\psi(s)$ does depend on the MDP M_φ making this construction statistically easy but computationally hard to solve.

2.3.2 From RL algorithms to 3-SAT algorithms

We now build a randomized algorithm $\mathscr{A}_{SAT}$ for UNIQUE-3-SAT using a randomized algorithm $\mathscr{A}_{RL}$ for the RL problem. However, as mentioned before, since the runtime for $\mathscr{A}_{RL}$ accrues only constant runtime for each call to the MDP oracle, to efficiently build $\mathscr{A}_{SAT}$ using $\mathscr{A}_{RL}$, we need to be able to efficiently simulate the calls to MDP oracle, namely: calls to the reward function, the transition function and the features. To do so, we build an "approximate" simulator $\bar{M}_\varphi$ for the MDP oracle M_φ. The simulator $\bar{M}_\varphi$ is exactly MDP M_φ in terms of transition function and features associated with the MDP M_φ, but differs in the reward function at the last layer which is always 0 for the simulator $\bar{M}_\varphi$. This modification is crucial for an efficient reduction because unlike transitions and features for any state which can be computed in time $\mathrm{poly}(d)$ on the MDP M_φ, the rewards at the last layer when $\mathrm{dist}(w,w^*) \neq 0$ require access to w^* which can not be done efficiently. With the purposed modification, we can execute each call to simulator $\bar{M}_\varphi$ in time $\mathrm{poly}(d)$.

Algorithm. On input 3-CNF formula φ, $\mathscr{A}_{SAT}$ runs the algorithm $\mathscr{A}_{RL}$ replacing each call to MDP oracle M_φ with the corresponding call to simulator $\bar{M}_\varphi$. Recall that the output for the RL algorithm in our setting is a sequence of actions. If the sequence of actions returned by $\mathscr{A}_{RL}$ ends on a state with assignment w, $\mathscr{A}_{SAT}$ outputs YES if w is the satisfying assignment and returns NO otherwise.

Correctness. We set the horizon $H = v^r$. We will assume throughout that $r \geq 2$ and that the runtime of $\mathscr{A}_{RL}$ is $\leq v^{r^2/4}$. The setting of r satisfying these assumptions will prove Proposition 2 for 3-action MDPs, which we will discuss in Section 2.3.3. To complete our reduction, we will show the following:

(i) If algorithm $\mathscr{A}_{RL}$ outputs a policy π such that $V^\pi > V^* - 1/4$, then $\mathscr{A}_{SAT}$ on 3-CNF formula φ outputs YES if φ is satisfiable and NO otherwise.

(ii) If $\mathscr{A}_{RL}$ with access to MDP oracle M_φ outputs a policy π such that $V^\pi > V^* - 1/4$ with error probability $1/10$, then $\mathscr{A}_{RL}$ with access to simulator $\bar{M}_\varphi$ outputs a policy π such that $V^\pi > V^* - 1/4$ with error probability $1/8$.

These together will show that $\mathscr{A}_{SAT}$ solves UNIQUE-3-SAT with error probability $\leq 1/8$. We start by proving that if $\mathscr{A}_{RL}$ succeeds on MDP $\bar{M}_\varphi$, then $\mathscr{A}_{SAT}$ succeeds on 3-CNF formula φ. This follows from the fact that any good policy in the MDP M_φ must reach a state with satisfying assignment w^*.

Proposition 4. *Suppose $r > 1$ and horizon $H = v^r$. If $\mathscr{A}_{RL}$ outputs a policy π such that $V^\pi > V^* - 1/4$, then $\mathscr{A}_{SAT}$ on* 3-CNF *formula φ outputs YES if φ is satisfiable and NO otherwise.*

Proof. Since algorithm $\mathscr{A}_{SAT}$ always returns NO on an unsatisfiable formula, we restrict our attention to a satisfiable formula φ. In the MDP M_φ, (i) rewards are "very small" everywhere except on reaching the satisfying assignment i.e. the expected reward at the last layer in the MDP M_φ is upper bounded by (for large enough v and $r > 1$)

$$\left(1 - \frac{H}{H+v}\right)^r = \left(\frac{v}{H+v}\right)^r \leq v^{-r^2+r} < 1/4$$

and (ii) the optimal value V^* is large

$$V^* \geq \left(1 - \frac{v}{H+v}\right)^r = \left(1 + \frac{v}{v^r}\right)^{-r} \geq 1 - \frac{rv}{v^r} \geq \frac{1}{2}$$

where the second last inequality follows from Bernoulli's inequality and the last inequality holds for large enough v and $r > 1$. Therefore, if the value of policy is large i.e. $V^\pi > V^* - 1/4$, then the policy π (and therefore the corresponding sequence of actions) has to end on a state with the satisfying assignment w^*. By construction of $\mathscr{A}_{SAT}$, this implies $\mathscr{A}_{SAT}$ will succeed on the formula φ. □

Since we can not simulate the rewards on MDP oracle M_φ efficiently, our reduction runs the algorithm $\mathscr{A}_{RL}$ on an approximate simulator $\bar{M}_\varphi$. However, it's not clear why $\mathscr{A}_{RL}$ would still succeed when each call to MDP oracle is replaced by a call to the simulator $\bar{M}_\varphi$. The following proposition shows that in fact $\mathscr{A}_{RL}$ would succeed on the outputs of simulator $\bar{M}_\varphi$ albeit with a smaller constant probability.

Proposition 5. *Suppose $r \geq 2$ and horizon $H = v^r$. Suppose $\mathscr{A}_{RL}$ with access to MDP oracle M_φ runs in time $v^{r^2/4}$ and outputs a policy π such that $V^\pi > V^* - 1/4$ with error probability $1/10$. Then $\mathscr{A}_{RL}$ with access to simulator $\bar{M}_\varphi$, still running in time $v^{r^2/4}$, outputs a policy π such that $V^\pi > V^* - 1/4$ with error probability $1/8$.*

Proof. Let $\Pr_{M_\varphi}$ and $\Pr_{\bar{M}_\varphi}$ denote the distribution on the observed rewards and output policies induced by the algorithm $\mathscr{A}_{RL}$ when running on access to MDP oracle M_φ and simulator $\bar{M}_\varphi$ respectively. Let R_i denote the reward received on the last layer at the end of i-th trajectory. Let T be the total number of trajectories sampled by algorithm $\mathscr{A}_{RL}$ when running on access to MDP oracle M_φ. By our assumption, $\mathscr{A}_{RL}$ runs in time $v^{r^2/4}$ and therefore $T \leq v^{r^2/4}$. Since the expected reward at the last layer in the MDP M_φ is upper bounded by (for large enough v and

$r \geq 2$)

$$\left(1 - \frac{H}{H+v}\right)^r = \left(\frac{v}{H+v}\right)^r \leq v^{-r^2+r} \leq v^{-\frac{r^2}{2}}$$

and and the algorithm only visits at most $v^{r^2/4}$ states on last layer, we get by the union bound that with high probability all the rewards at the last level are zero. More precisely (and assuming v is large enough),

$$\Pr_{M_\varphi}[R_i = 0 \; \forall i \in [T]] \geq 1 - v^{-r^2/4} \geq \frac{4}{5}$$

We say $\mathscr{A}_{RL}$ succeeds with access to M_φ (or $\bar{M}_\varphi$) if the output policy π after running for time at most $v^{r^2/4}$ satisfies $V^\pi > V^* - 1/4$. Using the above reasoning and the assumption that $\mathscr{A}_{RL}$ succeeds with access to MDP oracle M_φ with probability $9/10$ implies

$$\Pr_{M_\varphi}\left[\mathscr{A}_{RL} \text{ succeeds with access to } M_\varphi \mid R_i = 0 \; \forall i \in [T]\right] \geq \frac{\frac{9}{10} - \frac{1}{5}}{\frac{4}{5}} = \frac{7}{8}$$

Note that the marginal distributions $\Pr_{M_\varphi}$ and $\Pr_{\bar{M}_\varphi}$ on output policy π given $R_i = 0 \; \forall i \in [T]$ are exactly the same because MDP oracle $\bar{M}_\varphi$ and simulator M_φ only differ on last layer rewards. This implies

$$\begin{aligned} &\Pr_{\bar{M}_\varphi}\left[\mathscr{A}_{RL} \text{ succeeds with access to } \bar{M}_\varphi \mid R_i = 0 \; \forall i \in [T]\right] \\ &= \Pr_{M_\varphi}\left[\mathscr{A}_{RL} \text{ succeeds with access to } M_\varphi \mid R_i = 0 \; \forall i \in [T]\right] \end{aligned}$$

Since, $\Pr_{\bar{M}_\varphi}[R_i = 0 \; \forall i \in [T]] = 1$, we conclude that

$$\Pr_{\bar{M}_\varphi}\left[\mathscr{A}_{RL} \text{ succeeds with access to } \bar{M}_\varphi\right] \geq \frac{7}{8}$$

□

2.3.3 Setting of Parameters

It follows from Propositions 3 to 5 that if LINEAR-3-RL with feature dimension $d = 2v^r$ can be solved in time $v^{r^2/4}$ with error probability $1/10$, then UNIQUE-3-SAT with v variables can be solved in time $d \cdot v^{r^2/4}$ with error probability $1/8$ (here the extra d factor is because each call to the simulator $\bar{M}_\varphi$ takes d time). In this section, we discuss the two different settings of r we use to prove our lower bounds. As we increase r, we decrease the expected reward available to the algorithm at the last layer on the order of $v^{-O(r^2)}$, making the problem harder. However, increasing r also increases the feature dimension on the order of v^r. This non-polynomial gap in the feature dimension and expected reward at the last layer will give our main reduction.

In the first setting, we will set r to be a constant wrt number of variables v and prove that a polynomial algorithm for LINEAR-3-RL implies a polynomial algorithm for UNIQUE-3-SAT.

Proof of Proposition 2. For any $q \geq 1$, we set

$$r = 8q\,. \tag{2.2}$$

Note that $q \geq 1$ implies $r \geq 2$. Therefore, to prove our proposition, we just need to show

$$d^q \leq v^{r^2/4} \tag{2.3}$$

$$d \cdot v^{r^2/4} \leq v^{8q+16q^2+1} \tag{2.4}$$

under this setting of d and r. Here the first equation bounds the time complexity of LINEAR-3-RL in terms of feature dimension d and the second equation bounds the time complexity of UNIQUE-3-SAT in terms of the number of variables v. Equation (2.3) is true as

$$v^{\frac{r^2}{4}} = (v^r)^{\frac{r}{4}} \geq d^{\frac{r}{8}} = d^q$$

where the first inequality follows from $d \leq v^{2r}$ for large enough v and the last equality follows from Equation (2.2) above. Equation (2.4) holds since

$$d \cdot v^{r^2/4} = 2v^{r+r^2/4} = O(v^{8q+16q^2}),$$

where the first equality follows from $d = 2v^r$ and the last equality follows from Equation (2.2) for large enough v. □

Acknowledgements. Chapter 2 contains a reprint of the material as it appears in Conference on Learning Theory (COLT 2022). Daniel Kane, Sihan Liu, Shachar Lovett, Gaurav Mahajan. *Computational-statistical gaps in reinforcement learning*. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Computationally efficient algorithms for learning HMMs

3.1 Preliminaries

Notation. Let $\mathscr{O} := \{1, \ldots, O\}$ denote a finite observation space and let $\mathscr{O}^*$ denote observation sequences of arbitrary length. We consider a distribution $\Pr[\cdot]$ over T random variables $\mathbf{x}_1, \ldots, \mathbf{x}_T$ with a sequential ordering, and we use $x_t \in \mathscr{O}$ to denote the value taken by the t^{th} random variable. For convenience, we often simply write $\Pr[x_1, x_2, \ldots, x_T]$ in lieu of $\Pr[\mathbf{x}_1{=}x_1, \ldots, \mathbf{x}_T{=}x_T]$, omitting explicit reference to the random variables themselves.

When considering conditionals of this distribution, we *always* condition on assignment to a prefix of the random variables and marginalize out a suffix. For example, we consider conditionals of the form $\Pr[\mathbf{x}_{t+1}{=}x_{t+1}, \ldots, \mathbf{x}_{t+k}{=}x_{t+k} | \mathbf{x}_1{=}x_1, \ldots, \mathbf{x}_t{=}x_t]$, and we write this as $\Pr[x_{t+1}, \ldots, x_{t+k} | x_1, \ldots, x_t]$. Similarly, when considering tuples $f := (x'_1, \ldots, x'_k) \in \mathscr{O}^k$ and $h := (x_1, \ldots, x_t) \in \mathscr{O}^t$, we write $\Pr[\mathbf{x}_{t+1}{=}x'_1, \ldots, \mathbf{x}_{t+k}{=}x'_k | \mathbf{x}_1{=}x_1, \ldots, \mathbf{x}_t{=}x_t]$ as $\Pr[f|h]$, noting that the random variables assigned to f are determined by the length of h.

We lift this conditioning notation to sets of observation sequences in the following manner. If $F := \{f_1, f_2, \ldots\}$ and $H := \{h_1, h_2, \ldots\}$ where each $f_i, h_j \in \mathscr{O}^*$, we write $\Pr[F|H]$ to denote the $|F| \times |H|$ matrix whose $(i,j)^{\text{th}}$ entry is $\Pr[f_i | h_j]$. We allow the sequences in F and H to have different lengths, but always ensure that $\operatorname{len}(f_i) + \operatorname{len}(h_j) \leq T$ so that this matrix is well-defined.

We refer to rows and columns of this matrix as $\Pr[f|H]$ and $\Pr[F|h]$ respectively.[1]

Lastly, for $h = (x_1, \ldots, x_t)$ we use $ho = (x_1, \ldots, x_t, o)$ to denote concatenation, and we lift this notation to sequences and sets. For instance, if $H = \{h_1, h_2, \ldots\}$ then $Ho = \{h_1 o, h_2 o, \ldots\}$.

3.1.1 Hidden Markov Models and low rank distributions

Hidden Markov Models provide a low-complexity parametrization for distributions over observation sequences. These models are defined formally as follows.

Definition 6 (Hidden Markov Models)**.** Let $\mathscr{S} := \{1, \ldots, S\}$. An HMM with $S \in \mathbb{N}$ hidden states is specified by (1) an initial distribution $\mu \in \Delta(\mathscr{S})$, (2) an emission matrix $\mathbb{O} \in \mathbb{R}^{O \times S}$, and (3) a state transition matrix $\mathbb{T} \in \mathbb{R}^{S \times S}$, and defines a distribution over sequences of length T via:

$$\Pr[x_1, \ldots, x_T] := \sum_{s_1, \ldots, s_{T+1} \in \mathscr{S}^{T+1}} \mu(s_1) \prod_{t=1}^{T} \mathbb{O}[x_t, s_t] \mathbb{T}[s_{t+1}, s_t]. \tag{3.1}$$

Here $M[i, j]$ represents the $(i, j)^{\text{th}}$ entry of a matrix M.

As the name suggests, HMMs parameterize the distribution with a Markov chain over a hidden state sequence along with an emission function that generates observations. While this specific model is particularly natural, our analysis only leverages a certain low rank structure present in HMMs. To highlight the importance of this structure, we define the *rank* of a distribution.

Definition 7 (Rank of a distribution)**.** We say distribution $\Pr[\cdot]$ over observation sequences of length T has rank r if, for each $t \in [T]$, the conditional probability matrix $\Pr[\mathscr{O}^{\leq T-t} | \mathscr{O}^t]$ has rank at most r.[2]

[1] We always refer to rows, columns, and entries of these matrices in this manner, so no confusion arises when constructing these matrices from (unordered) sets of sequences.

[2] When some histories occur with zero probability, there might be multiple consistent conditional probability functions associated to a distribution, in which case the rank is not uniquely defined. We address this by *defining* the distribution via its conditionals (which determine the rank); see [55].

An HMM with S hidden states has rank at most S, which can be verified using the fact that the hidden states form a Markov chain (we give a proof in [55]).[3] More generally, the rank identifies a low dimensional structure in the distribution: we have exponentially many vectors $\Pr[\mathscr{O}^{\leq T-t}|h]$, one for each history h, in an r-dimensional subspace of an exponentially larger ambient space. Thus, we are interested in algorithms that exploit the low dimensional structure and admit statistical and computational guarantees scaling polynomially with the rank.

3.1.2 Learning models

To circumvent computational hardness, we allow the learner to access conditional distributions of the underlying distribution $\Pr[\cdot]$. We specifically consider two access models formalized with the following oracles: [4]

Definition 8 (Exact conditional probability oracle)**.** The exact conditional probability oracle is given as input: observation sequences h and f of length $t \leq T$ and $T-t$ respectively, chosen by the algorithm, and returns the scalar $\Pr[f|h]$.

Definition 9 (Conditional sampling oracle)**.** The conditional sampling oracle is given as input: an observation sequence h of length $t \leq T$, chosen by the algorithm, and returns an observation sequence f of length $T-t$ such that the probability that f is returned is $\Pr[f|h]$, independently of all other randomness.

When considering the exact probability oracle, we also allow the learner to obtain independent samples from the joint distribution $\Pr[\cdot]$. Note that this oracle equivalently provides access to exact (unconditional) probabilities of length T sequences. We view this as a noiseless analog of the conditional sampling oracle, which is the main model of interest.

As a learning goal, we consider distribution learning in total variation distance as studied in prior works [59, 70, 46, 9]. Given access to a target distribution $\Pr[\cdot]$ we want to efficiently

[3] In fact the rank of the HMM can be much smaller, since the decomposition alluded to above realizes the non-negative rank of the matrix, which can be exponentially larger than the rank.

[4] Both oracles require committing to a consistent choice of conditional probability distribution when conditioning on zero probability events. See [55].

compute an estimate $\widehat{\Pr}[\cdot]$ that is close in total variation distance to $\Pr[\cdot]$. Formally, we want an algorithm that, when given parameters $\varepsilon, \delta > 0$, computes an estimate $\widehat{\Pr}[\cdot]$ such that with probability at least $1-\delta$:

$$\mathrm{TV}(\Pr, \widehat{\Pr}) := \frac{1}{2} \sum_{x_1,\ldots,x_T \in \mathscr{O}^T} \left| \Pr[x_1, \ldots, x_T] - \widehat{\Pr}[x_1, \ldots, x_T] \right| \leq \varepsilon.$$

The algorithm is efficient if its computational complexity (and hence number of oracle calls) scale polynomially in $r, T, O, 1/\varepsilon$ and $\log(1/\delta)$.

Remark 10. Note that, as the support of $\Pr[\cdot]$ is exponentially large in T, it is not possible to write down all $\mathscr{O}^T$ values of $\widehat{\Pr}$ efficiently. Instead, the goal is to return an efficient representation from which we can evaluate $\widehat{\Pr}[x_1, \ldots, x_T]$ for any sequence $x_1, \ldots, x_T$ efficiently. It will become clear what constitutes an efficient representation for low rank distributions in the sequel; indeed the fact that one even exists is one of our central structural results. for HMMs, for example, the tuple of initial distribution μ, observation operator $\mathbb{O}$, and transition operator $\mathbb{T}$ form an efficient representation.

3.2 Our results

Our first result studies the computational power provided by the exact probability oracle (Definition 8). We show how a generalization of Angluin's L^* algorithm can efficiently learn any HMM given access to this oracle. The result is summarized in the following theorem:[5]

Theorem 3 (Learning with exact conditional probabilities)**.** *Assume* $\mathscr{O} = \{0, 1\}$. *Let* $\Pr[\cdot]$ *be any rank r distribution over observation sequences of length T. Pick any* $0 < \varepsilon, \delta < 1$. *Then Algorithm 1 with access to an exact probability oracle and samples from* $\Pr[\cdot]$*, runs in poly*$(r, T, 1/\varepsilon, \log(1/\delta))$ *time and returns an efficiently represented approximation* $\widehat{\Pr}[\cdot]$ *satisfying* $\mathrm{TV}(\Pr, \widehat{\Pr}) \leq \varepsilon$ *with probability at least* $1-\delta$.

[5] As this result is a warmup for our main result, we focus on the setting where $\mathscr{O} = \{0, 1\}$ for simplicity.

The main technical challenge is finding a succinct and observable representation of the distribution, so that we can infer all conditional distributions using polynomially many queries. This observable parameterization plays a central role in our main result, and in this sense Theorem 3 can be seen as an insightful warmup.

Our main contribution is in extending this result to the more natural interactive setting where the learner only accesses conditional samples via the oracle in Definition 9. Our algorithm here can be viewed as a robust version of L^*, and we obtain the following guarantee:

Theorem 4 (Learning with conditional samples)**.** *Let* $\Pr[\cdot]$ *be any rank r distribution over observation sequences of length T. Assume distribution* $\Pr[\cdot]$ *has fidelity* Δ^**. Pick any* $0 < \varepsilon, \delta < 1$*. Then Algorithm 2 with access to a conditional sampling oracle runs in poly*$($*r, T, O,* $1/\Delta^*$*,* $1/\varepsilon$*,* $\log(1/\delta))$ *time and returns an efficiently represented approximation* $\widehat{\Pr}[\cdot]$ *satisfying* $\mathrm{TV}(\Pr, \widehat{\Pr}) \leq \varepsilon$ *with probability at least* $1 - \delta$*.*

The theorem provides a robust analog to Theorem 3 in the much weaker conditional sampling access model. The caveat is that the guarantee depends on a spectral property of a distribution, which we call the fidelity. The definition of fidelity (Definition 13) requires further development of the algebraic structure in $\Pr[\cdot]$ and is deferred to Section 3.3. Nevertheless, we can show that the cryptographically hard examples of HMMs and positive results from prior work on learning HMMs have fidelity that is lower bounded by a (small) polynomial of the other parameters and thus are efficiently learnable by our algorithm (see [55]). On the other hand, there are HMMs with exponentially small fidelity, and we have no evidence that these instances are computationally intractable when provided with conditional samples. This leads to the main open question stemming from our work.

Open Problem 11. *Is there a computationally efficient algorithm for learning* any *low rank distribution given access to a conditional sampling oracle?*

Chapter organization. In Section 3.3, we present an overview of our techniques, explaining the challenges and how we address them. Then we turn to the more formal presentation of the proofs, with Section 3.4 devoted to Theorem 3 and Section 3.5 devoted to Theorem 4. These sections present our algorithms and the main ingredients for their analysis, with some details deferred to the appendices. We close the main body of the chapter in Section 3.6, with some further discussion regarding Open Problem 11.

3.3 Technical overview

To explain the central challenges with learning low rank distributions and how we overcome them, let us introduce the following notation: let $H_t := \mathscr{O}^t$ and $F_t := \mathscr{O}^{T-t}$ denote the observation sequences of length t and $T-t$ respectively. Then the matrix $\Pr[F_t|H_t]$ is a submatrix of $\Pr[\mathscr{O}^{\leq T-t}|\mathscr{O}^t]$ and hence is rank at most r by assumption. If we define these matrices for each length $t \in [T]$, then clearly we have encoded the entire distribution. Hence, estimating these matrices in an appropriate sense would suffice for distribution learning. Although the matrices all have rank at most r, they are exponentially large, so the low rank property does not immediately yield an efficient representation of the distribution. Indeed, we must leverage further structure to obtain efficient algorithms.

3.3.1 Background: Observable operators and hard instances

For HMMs, we can hope to leverage the explicit formula for the probability of a sequence (Equation (3.1)) to obtain an efficient algorithm. Indeed, this is the approach adopted by Hsu, Kakade, and Zhang [46]. Specifically, they use the *observable operator* representation [47]: if we define $S \times S$ matrices $\{\mathbb{A}_o\}_{o\in\mathscr{O}}$ as $\mathbb{A}_o := \mathbb{T}\mathrm{diag}(\mathbb{O}[o,\cdot])$ then we can write the probability of any observation sequence as

$$\Pr[x_1,\ldots,x_T] = \mathbf{1}^\top \mathbb{A}_{x_T} \ldots \mathbb{A}_{x_1} \mu,$$

where $\mathbf{1}$ is the all-ones vector and recall that μ is the initial state distribution. Hsu, Kakade and Zhang show that these operators can be estimated, up to a linear transformation, whenever $\mathbb{T}$ and

$\mathbb{O}$ have full column rank. In fact, under their assumptions, these operators can be recovered from $\Pr[\mathbf{x}_1{=}\cdot, \mathbf{x}_2{=}\cdot, \mathbf{x}_3{=}\cdot]$ alone; no higher order moments of the distribution are required.

Unfortunately, this approach fails if either $\mathbb{T}$ or $\mathbb{O}$ are (column) rank deficient, and it is conjectured that the rank deficient HMMs are precisely the hard instances [70]. On the other hand, many interesting HMMs *are* rank deficient. For example, any *overcomplete* HMM—one with fewer observations than states—cannot have a full column rank $\mathbb{O}$ matrix. This captures all deterministic finite automata where the alphabet size is smaller than the number of states as well as the parity with noise problem.

Learning parity with noise is a particularly interesting case. The standard formulation is that we obtain samples of the form $(\mathbf{z}, \mathbf{y}) \in \{0,1\}^{T-1} \times \{0,1\}$ where $\mathbf{z}$ is uniformly distributed on the hypercube and $\mathbf{y} = \bigoplus_{i \in I} \mathbf{z}_i$ with probability $1 - \alpha$ and $\mathbf{y} = 1 - \bigoplus_{i \in I} \mathbf{z}_i$ with the remaining probability. Here $\bigoplus$ denotes the parity operation, I is a secret subset of indices $I \subseteq [T-1]$, and $\alpha \in (0, 1/2)$ is a noise parameter. We want to learn the subset I, given samples from this process. This problem is widely believed to be computationally hard and can be encoded as an HMM with $\mathscr{O} = \{0,1\}$ and $4T$ states (see [55]). Considering this problem, it is quite apparent that low degree moments, like those used by Hsu, Kakade, and Zhang, reveal no information about the subset I. In particular, the observable operators $\mathbb{A}_o$ are not identifiable from low degree moments. One must use higher order information, i.e., statistics about long sequences, to solve this problem.

3.3.2 Efficient representation

For rank deficient HMMs, it is not clear how to identify the observable operators and it is not even clear that such operators exist for the more general case of low rank distributions. So, we must return to the question of how to efficiently represent the distribution. Here, our first observation is that any submatrix of $\Pr[F_t | H_t]$ that has the same rank as the entire matrix can be used to build an efficient representation. To see why, suppose we have such a submatrix, and let us index the columns/histories of the submatrix by B_t, which we refer to as the *basis*. It follows

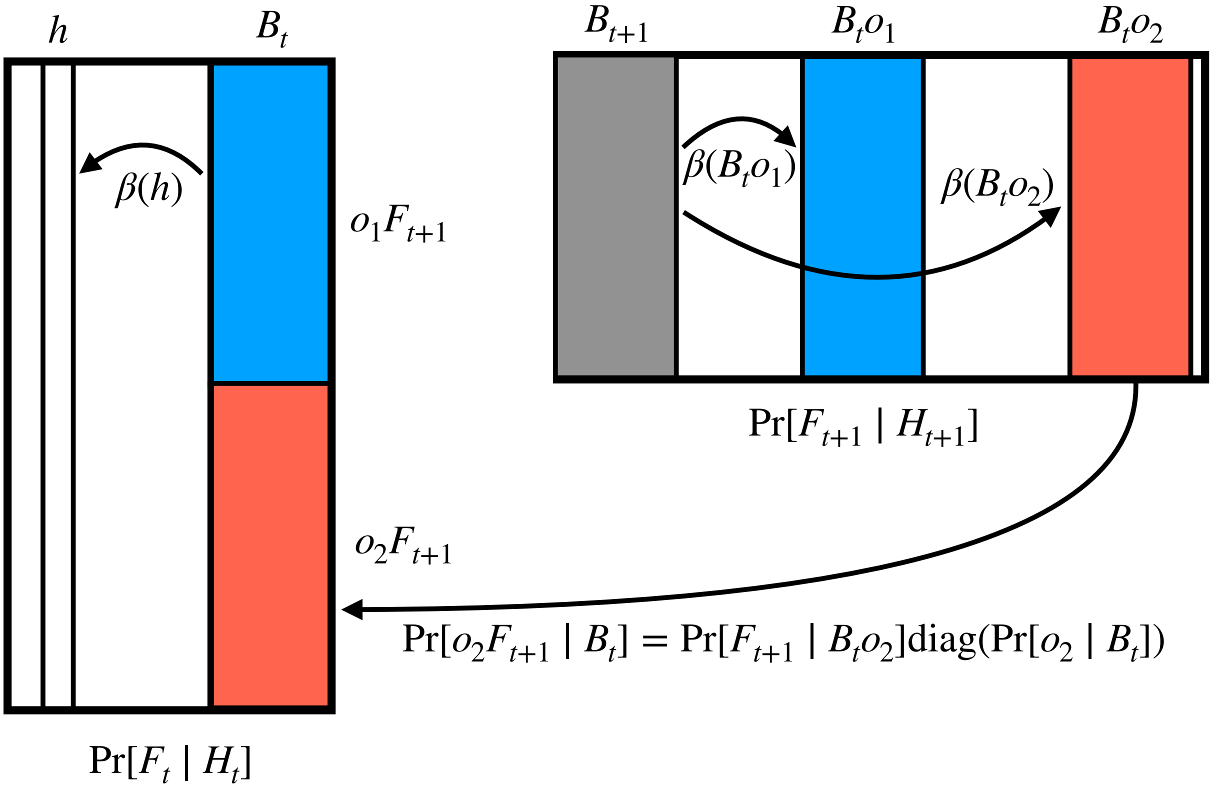

Figure 3.1. Schematic of the circulant structure relating the $\Pr[F_t|H_t]$ and $\Pr[F_{t+1}|H_{t+1}]$ matrices. Columns of $\Pr[F_t \mid H_t]$ can be represented linearly in basis B_t using coefficients $\beta(\cdot)$. The blocks $\Pr[oF_{t+1} \mid B_t]$ appear in the next matrix $\Pr[F_{t+1} \mid H_{t+1}]$ (up to scaling), so they can be represented in basis B_{t+1}.

that $\Pr[F_t|B_t]$ spans the column space of $\Pr[F_t|H_t]$, which implies that for any history $h \in H_t$ there exists coefficients $\beta(h) \in \mathbb{R}^{|B_t|}$ such that

$$\Pr[F_t|h] = \Pr[F_t|B_t]\beta(h).$$

The main observation toward obtaining an efficient representation is to exploit a certain circulant structure in the matrices $\{\Pr[F_t|H_t]\}_{t\leq T}$ to model the evolution of the coefficients (visualized in Figure 3.1). The circulant structure is simply that for basis B_t, observation o, and future $f \in F_{t+1}$ (i.e., of length $T-t-1$) the vector $\Pr[B_tof]$ appears in two of the matrices (albeit with different scaling). It appears in the matrix $\Pr[F_t|H_t]$ in row of and columns B_t, and it appears in the matrix $\Pr[F_{t+1}|H_{t+1}]$ in row f and columns B_to. Thus, if we learn how to represent the columns $\Pr[F_{t+1}|B_t o]$ in terms of the columns $\Pr[F_{t+1}|B_{t+1}]$—which we can do via the coefficients—the circulant property provides a connection between the matrices $\Pr[F_{t+1}|H_{t+1}]$ and $\Pr[F_t|H_t]$.

Formally, we can define operators $\{A_{o,t}\}$ for each observation $o \in \mathscr{O}$ and sequence length $t \in [T]$ satisfying

$$\Pr[F_{t+1}|B_{t+1}]A_{o,t} = \Pr[oF_{t+1}|B_t], \tag{3.2}$$

which can then be used to express sequence probabilities by iterated application. Indeed, we have

$$\begin{aligned} \Pr[x_1,\ldots,x_T] = \Pr[x_1,\ldots,x_T|B_0] = \Pr[x_2,\ldots,x_T|B_1]A_{x_1,0} = \ldots \\ \ldots = \Pr[x_T|B_{T-1}]A_{x_{T-1},T-2}\ldots A_{x_1,0} = A_{x_T,T-1}\ldots A_{x_1,0}\,, \end{aligned} \tag{3.3}$$

where by an explicit choice of B_0, B_T and F_T, the matrices $A_{x_1,0}$ and $A_{x_T,T-1}$ are column and row vectors respectively, and so the right-hand side is a scalar (see Proposition 16 for details)[6]. More importantly, these operators can also be viewed as evolving the coefficients via the identity:

$$\forall h \in H_t, o \in \mathscr{O} : \beta(ho) = \frac{A_{o,t}\beta(h)}{\Pr[o|h]}. \tag{3.4}$$

This identity is proved in Proposition 16. We highlight the scaling, which results in a nonlinear update equation and appears because the coefficients express conditional rather than joint probabilities. This viewpoint of operators evolving coefficients will play a central role in our error analysis.

Thus, it remains to find the bases $\{B_t\}_{t\leq T}$, estimate the operators $\{A_{o,t}\}_{o\in\mathscr{O},t\leq T}$, and

[6] We define B_0, B_T and F_T to be singleton sets. B_0 and F_T contain the empty string φ and B_T contains any length T observation sequence. These new definitions, in conjunction with Proposition 16 imply: $A_{x_T,T-1} = \Pr[x_T|B_{T-1}]$ and therefore will be a row vector. Similarly, $A_{x_1,0}$ is a solution of $\Pr[F_1|B_1]A_{x_1,0} = \Pr[x_1F_1|\varphi]$ and is therefore a column vector.

control the error amplification from iteratively multiplying these estimates. We turn to these issues next.

Remark 12. The approach of Hsu, Kakade, and Zhang can also be viewed as estimating operators via Equation (3.2) with the particular choice of basis. They show that conditional distribution of futures given any history can be written in the span of the conditional distributions of the single observation histories, so that $\mathscr{O}$ itself forms a basis. This is implied by their assumptions and it permits using only second and third degree moments to estimate the operators. However, in general we will need to use long sequences in our bases and interactive access will be crucial for estimation. Additionally, under their choice of bases and their assumptions they show that the solution of Equation (3.2) is related to the observable operators [47], explicitly given by $\mathbb{T}$ and $\mathbb{O}$, by an invertible and bounded transformation, which is instrumental in their error analysis. When considering general bases B, we do not have such a connection and will require a novel error propagation argument.

3.3.3 Error propagation

Although finding the bases B_t and estimating corresponding operators $A_{o,t}$ is nontrivial, even if we have estimated these operators accurately, we must address the error amplification that can arise from repeated application of the learned operators. This challenge makes up the majority of our technical analysis. We discuss estimating operators $A_{o,t}$ in Section 3.3.4 and how to find the basis in Section 3.3.5.

To explain the error amplification challenge, suppose for now that we are given bases $\{B_t\}_{t\leq T}$ and subsequently estimate the operators $A_{o,t}$ in ℓ_2 norm, i.e., we have estimate $\widehat{A}_{o,t}$ satisfying $\|\widehat{A}_{o,t} - A_{o,t}\|_2 \leq \varepsilon$. We first define our estimated model $\widehat{\Pr}$ in terms of the estimated operators $\widehat{A}_{o,t}$. Considering Equation (3.3), the natural estimator is

$$\widehat{\Pr}[x_1,\dots,x_T] = \widehat{A}_{x_T,T-1}\dots\widehat{A}_{x_2,1}\widehat{A}_{x_1,0}, \tag{3.5}$$

where, as before, the matrices $\widehat{A}_{x_1,0}$ and $\widehat{A}_{x_T,T-1}$ are column and row vectors respectively, so the right hand side is a scalar. To simplify notation for this section, we omit the time indexing on the operators.

Given this estimate, the total variation distance is

$$\frac{1}{2}\sum_{x_1,\ldots,x_T\in\mathscr{O}^T}\left|\widehat{A}_{x_T}\ldots\widehat{A}_{x_1}-A_{x_T}\ldots A_{x_1}\right|.$$

Let us first discuss two strategies for bounding this expression that can work in some cases, but do not seem to work in our setting. One idea is to pass to the ℓ_2 norm and use a telescoping argument to obtain several terms of the form

$$\sum_{x_1,\ldots,x_T\in\mathscr{O}^T}\|\widehat{A}_{x_T}\ldots\widehat{A}_{x_{t+2}}\|_2\cdot\|\left(\widehat{A}_{x_{t+1}}-A_{x_{t+1}}\right)A_{x_t}\ldots A_{x_1}\|_2$$

These terms are convenient because the matrix products only disagree in the t^{th} operator. However, both the "incoming" product $A_{x_t}\ldots A_{x_1}$ that pre-multiplies this difference and the "outgoing" product $\widehat{A}_{x_T}\ldots\widehat{A}_{x_{t+2}}$ whose norm we must bound can be rather poorly behaved. For example, the product $A_{x_t}\ldots A_{x_1}$ can have ℓ_2 norm that grows exponentially with t, since the ℓ_2 norm of the individual matrices can be much larger than 1. An even worse problem is that we have exponentially many terms in the sum, so that even bounding each term by ε (which would be possible if the incoming and outgoing products were well behaved) is grossly insufficient.

The other approach is the strategy adopted by Hsu, Kakade, and Zhang [46], which uses the definition of the observable operators [47], $\mathbb{A}_x=\mathbb{T}\mathrm{diag}(\mathbb{O}[x,\cdot])$, explicitly. This allows them to control the incoming and outgoing products in a decomposition analogous to the one above, but in the ℓ_1 norm. Their decomposition involves several terms, but to convey the main idea, observe that we can bound

$$\sum_{x_1,\ldots,x_{t+1}}\|\left(\widehat{\mathbb{A}}_{x_{t+1}}-\mathbb{A}_{x_{t+1}}\right)\mathbb{A}_{x_t}\ldots,\mathbb{A}_{x_1}\|_1\lesssim O\varepsilon\cdot\sum_{x_1,\ldots,x_t}\|\mathbb{A}_{x_t}\ldots,\mathbb{A}_{x_1}\|_1\leq O\varepsilon.$$

The idea is that each term in the final sum can be seen as a joint probability of the history $x_1,\ldots,x_t$ and the hidden state s_{t+1}, so we can sum over all histories with no error amplification. Unfortunately, there is no hidden state in the more general setting (and for the rank deficient case, the observable operators can not be learned accurately as discussed in Section 3.3.1), so we cannot appeal to an argument of this form. Indeed, our main technical contribution is a new perturbation analysis that relies on no structural assumptions.

At a more technical level, the issue with both of these arguments is that passing to any norm, seems to be too coarse to adequately control the error amplification. Instead, our argument carefully tracks the error in the space of the coefficients. Precisely, given estimates $\widehat{A}_{o,t}$ that satisfy $\|\widehat{A}_{o,t} - A_{o,t}\|_2 \leq \varepsilon$, we can show, via an inductive argument, that for any $x_1,\ldots,x_t$

$$(\widehat{A}_{x_t}\ldots\widehat{A}_{x_1} - A_{x_t}\ldots A_{x_1}) = \sum_{h\in H_t} \beta(h)\alpha_h + \sum_{v\in V_t^{\perp}} v\gamma_v,$$

where $V_t^{\perp}$ is an orthonormal basis for the kernel of $\Pr[F_t \mid B_t]$ and α_h, γ_v are scalars. Moreover, the TV distance between $\Pr[\cdot]$ and $\widehat{\Pr}[\cdot]$ is exactly equal to the sum of these scalars over all sequences $x_1,\ldots,x_T$. Even though there could be exponentially many terms in this sum, we show that this sum is small via an inductive argument. This makes up the most technical component of our proof, and we give a more detailed overview in Section 3.5 with the formal proofs in [55].

3.3.4 Estimating operators

We next discuss estimating the operators $\{A_{o,t}\}_{o\in\mathscr{O},t\leq T}$ using the conditional sampling oracle. A natural idea is to use samples to estimate both sides of the system in Equation (3.2) and solve the noisy version via linear regression. Unfortunately, this system may have exponentially small (in $T-t$) singular values, making it highly sensitive to perturbation. There is also a cosmetic issue when working with $\Pr[F_{t+1}|B_{t+1}]$, namely this matrix is exponentially large.

To address these challenges, we introduce a particular preconditioner that stabilizes the system. Specifically, we instead estimate and solve

$$\Pr[F_{t+1}|B_{t+1}]^\top D_{t+1}^{-1}\Pr[F_{t+1}|B_{t+1}]A_{o,t} = \Pr[F_{t+1}|B_{t+1}]^\top D_{t+1}^{-1}\Pr[oF_{t+1}|B_t]\,, \tag{3.6}$$

where D_{t+1} is a diagonal matrix with entries $d_{t+1}(f) := \frac{1}{|B_{t+1}|}\sum_{b\in B_{t+1}}\Pr[f|b]$ on the diagonal.[7] The benefit of this preconditioner is that the new matrices are of size $|B_{t+1}| \times |B_{t+1}|$ rather than exponentially large, and yet they can still be estimated efficiently using the conditional sampling oracle. To see why the latter holds, observe that the $(i,j)^{\text{th}}$ entry of the matrix on the LHS is

$$\left[\Pr[F_{t+1}|B_{t+1}]^\top D_{t+1}^{-1}\Pr[F_{t+1}|B_{t+1}]\right]_{i,j} = \sum_{f\in F_{t+1}} d_{t+1}(f)\left[\frac{\Pr[f|b_i]\Pr[f|b_j]}{d_{t+1}(f)^2}\right],$$

where $B_{t+1} = \{b_1, b_2, \ldots, \}$. Intuitively, we can estimate this entry by sampling futures f from $\Pr[\cdot|b]$ to approximate any term in the sum and sampling futures from $d_{t+1}(\cdot)$ to approximate the sum itself. While this is true, there is one technical issue to overcome: to estimate the ratio to additive accuracy, we must estimate the individual probabilities $\Pr[f \mid b_i]$, $\Pr[f \mid b_j]$ and $d_{t+1}(f)$ to relative accuracy. We can obtain $(1\pm\zeta)$ relative error estimates using conditional samples as long as the one-step probabilities are at least $\Omega(\zeta/T)$, but this is challenging when even a single one-step probability is small. To address this issue, we show that such futures actually contribute very little to the overall sum, and we design a test to safely ignore them. See [55] for details.

While the ability to estimate the entries is clearly important, the hope with preconditioning is that it dramatically amplifies the singular values of the matrix on the left hand side. In particular, we want that the matrix $\Pr[F_{t+1}|B_{t+1}]^\top D_{t+1}^{-1}\Pr[F_{t+1}|B_{t+1}]$ has large (non-zero) singular values, as this will allow us to estimate the operators $A_{o,t}$ in the ℓ_2 norm. Our choice of preconditioner does achieve this in the important example of parity with noise: we can show that $\Pr[F_{t+1}|B_{t+1}]$ has exponentially small (in $T-t$) singular values for every choice of B_{t+1}, while there exists a basis B_{t+1} for which the non-zero singular values of the preconditioned matrix are $\Omega(1)$ (see [55]).

[7] This choice of D_{t+1} ensures there is no division-by-zero issue, see [55].

Unfortunately, in general, a basis which ensures the preconditioner has large singular values might not exist, and we address this by introducing the notion of fidelity.

Definition 13 (Fidelity)**.** We say that distribution $\Pr[\cdot]$ has fidelity Δ^* if there exists some bases $\{B_t\}_{t\in[T]}$, such that $\max_t |B_t| \leq 1/\Delta^*$ and

$$\forall t \in [T]:\ \sigma_+ \left(S_t^{\frac{1}{2}} \Pr[F_t|H_t]^\top D_t^{-1} \Pr[F_t|H_t] S_t^{\frac{1}{2}} \right) \geq \Delta^*$$

where $\sigma_+(M)$ denotes the magnitude of the smallest non-zero eigenvalue of M, D_t is a diagonal matrix of size $|F_t| \times |F_t|$ with entries $d_t(f) := \frac{1}{|B_t|}\sum_{b\in B_t} \Pr[f|b]$, and S_t is a diagonal matrix of size $|H_t| \times |H_t|$ with entries $s_t(h) := \Pr[h]$.

Importantly, we only assume the existence of bases with this property, not that it is given to us or otherwise known in advance. Note that, although the matrix with large eigenvalues according to the fidelity definition is not the same as the preconditioned matrix we care about for learning operators, nevertheless when the distribution has high fidelity (i.e., Δ^* is large), we can find a basis for which $\Pr[F_{t+1}|B_{t+1}]^\top D_{t+1}^{-1} \Pr[F_{t+1}|B_{t+1}]$ has large eigenvalues. This, combined with our approach for estimating entries of the preconditioned matrix, allow us to learn operators $A_{o,t}$ in the ℓ_2 norm. We provide details in [55].

Remark 14. Although our approach seems to require large fidelity, the parity with noise example suggests that this definition of fidelity, which can lead to a favorable preconditioned system, is more appropriate than directly assuming $\Pr[F_{t+1}|B_{t+1}]$ has large singular values. Indeed, we can also show that fidelity captures all previously studied positive results for learning HMMs. We also believe our approach can be extended to learn HMMs with small fidelity as described in Section 3.6.

3.3.5 Finding the basis

The only remaining challenge is to find the bases $\{B_t\}_{t\in[T]}$. Recall that, when considering the conditional sampling oracle, we want bases for which the preconditioned matrices have large eigenvalues. It turns out that when the distribution has high fidelity a random sample of polynomially many histories will form a basis with this property with high probability. Given that the other aspects of our analysis seem to require high fidelity, this random sampling approach thus suffices to prove Theorem 4.

On the other hand, for low fidelity distributions, random sampling will fail to cover the directions with small singular value, and so basis finding becomes an intriguing aspect of learning with the conditional sampling oracle. Basis finding is also the final issue to address for Theorem 3, using the exact oracle. In both cases, we provide adaptations of Angluin's L^* algorithm that finds bases for any low rank distribution. We defer discussion of the conditional sampling version to [55] and hope that it serves as a starting point toward resolving Open Problem 11.

Adapting L^* for basis finding with the exact oracle. We close this section by explaining how to find a basis when provided with the exact probability oracle. As a first observation, note that we need not construct the entire system in Equation (3.2) to identify operators $A_{o,t}$. It suffices to find a set of futures $\Lambda_t \subset F_t$ such that $\Pr[\Lambda_t \mid H_t]$ spans the row space of $\Pr[F_t \mid H_t]$. In other words, we just need B_t and Λ_t for which $\Pr[\Lambda_t \mid B_t]$ has the same rank as $\Pr[F_t \mid H_t]$.

The difficulty is that there is no universal choice of B_t, Λ_t for general low rank distributions, and finding these sets poses a challenge search problem in an exponentially large space. We address this challenge using the exact probability oracle and an adaptation of Angluin's L^* algorithm for learning DFAs. The basic idea is as follows: given sets B_t, Λ_t whose submatrix is not of the required rank, we can still solve the underdetermined system

$$\Pr[\Lambda_t | B_t] A_{o,t} = \Pr[o\Lambda_t | B_t]$$

and obtain an estimate $\widehat{\Pr}[\cdot]$ via Equation (3.5). Then, we can sample sequences $x_1, \ldots, x_t \sim \Pr[\cdot]$ and check if our estimate makes the correct predictions on these sequences. In particular, we check

$$\widehat{\Pr}[x_1,\ldots,x_t,\Lambda_t] \overset{?}{=} \Pr[x_1,\ldots,x_t,\Lambda_t].$$

If the predictions are accurate (i.e., these equalities hold) for each t and for polynomially many random sequences, then we can show that $\widehat{\Pr}[\cdot]$ is close $\Pr[\cdot]$ in total variation distance.

On the other hand, if these equalities do not hold for some sample $x_1,\ldots,x_t$, then we can use it as a counterexample to improve our basis. We provide all the details in Section 3.4.

3.4 Learning with conditional probabilities

In this section we prove Theorem 3.

Theorem 3 (Learning with exact conditional probabilities)**.** *Assume* $\mathscr{O} = \{0,1\}$*. Let* $\Pr[\cdot]$ *be any rank r distribution over observation sequences of length T. Pick any* $0 < \varepsilon, \delta < 1$*. Then Algorithm 1 with access to an exact probability oracle and samples from* $\Pr[\cdot]$*, runs in* $poly(r,T,1/\varepsilon,\log(1/\delta))$ *time and returns an efficiently represented approximation* $\widehat{\Pr}[\cdot]$ *satisfying* $\mathrm{TV}(\Pr,\widehat{\Pr}) \leq \varepsilon$ *with probability at least* $1-\delta$*.*

We first introduce some notation, which differs from Section 3.3 slightly. We define $H_t := \mathscr{O}^t$ to be the set of histories of length t. Similarly, we define $F_t := \mathscr{O}^{\leq T-t}$ to be the set of futures of length $\leq T-t$, coinciding with our rank definition. Notice that unlike in Section 3.3, we take F_t to be all futures of length *up to* $T-t$, so that one may append elements from the futures F_t to elements from the histories H_t to obtain a valid observation sequence of length at most T. To simplify the technical notation, let φ be the empty string and define probabilities associated to empty string as: $\Pr[x_1\ldots x_T|\varphi] = \Pr[x_1\ldots x_T]$ and $\Pr[\varphi|x_1\ldots x_T] = 1$ for any T-length sequence $x_1,\ldots,x_T$.

We now formally define the notion of bases for distribution $\Pr[\cdot]$.

Definition 15 (Basis)**.** Let $\Pr[\cdot]$ be any distribution over observation sequences of length T. A set $\{B_t\}_{t\in[T]}$, where each $B_t \subset H_t$, forms *bases* for $\Pr[\cdot]$, if for each $t \in [T]$ and all $x \in \mathscr{O}^t$, there exists coefficients $\beta(x)$ such that:

$$\Pr[F_t|x] = \Pr[F_t|B_t]\beta(x)\,.$$

We call each B_t a *basis* for $\Pr[\cdot]$ at sequence length t.

In other words, a set $B_t \subset H_t$ forms a basis for distribution $\Pr[\cdot]$ if the column vectors $\Pr[F_t|B_t]$ span the column space of $\Pr[F_t|H_t]$. For now, when choosing B_t, we impose no constraint on the size of these coefficients, and we also do not require the columns $\Pr[F_t \mid B_t]$ to be linearly independent. The low rank property of $\Pr[\cdot]$ directly implies that for each t, there exists a basis B_t with $|B_t| \leq r$. However, as discussed in Section 3.3.2, there are exponentially many histories in H_t, so even if we had such a small basis B_t, simply learning the coefficients for each history will not suffice for an efficient algorithm. We address this issue with the following structural result: because of the circulant structure of the conditional probability matrix, we can generate all the coefficients using OT matrices each of size at most $r \times r$.

Proposition 16 (Existence of efficient representation)**.** *Let $B_0 = F_T = \{\varphi\}$ and $B_T = \{h\}$ for any observation sequence $h \in H_T$.*[8] *For $t \in \{1,\ldots,T-1\}$, let $B_t \subset H_t$ be any basis for distribution* $\Pr[\cdot]$ *at sequence length t. Then, the probability distribution* $\Pr[\cdot]$ *can be written as*[9]*:*

$$\Pr[x_1\ldots x_T] = A_{x_T,T-1}A_{x_{T-1},T-2}\ldots A_{x_1,0}$$

[8] We set B_T to be a singleton set for notational clarity, as otherwise we would have to pre-multiply our probability estimate with the all ones row vector. Note that any singleton set forms a basis because $\Pr[F_T|H_T]$ is the all ones matrix.

[9] Here by choice of basis B_0 and B_T, $A_{x_T,T-1} = \Pr[x_T|B_{T-1}]$ by definition and is therefore a row vector. Similarly, $A_{x_1,0}$ is a solution of $\Pr[F_1|B_1]A_{x_1,0} = \Pr[x_1F_1|\varphi]$ and is therefore a column vector.

where matrices $A_{o,t}$ for every $o \in \mathcal{O}$ and $t \in \{0,\ldots,T-1\}$ satisfy

$$\Pr[F_{t+1}|B_{t+1}]A_{o,t} = \Pr[oF_{t+1}|B_t]. \tag{3.7}$$

Moreover, this equation always has a solution.

Proof. We first show there exists a solution $A_{o,t}$ for Equation (3.7). For basis $B_t = \{b_1,\ldots,b_n\}$ and B_{t+1}, we claim the following $A_{o,t}$ is a solution:

$$A_{o,t} = \begin{bmatrix} \beta(b_1o) & \beta(b_2o) & \cdots & \beta(b_no) \end{bmatrix} \begin{bmatrix} \Pr[o|b_1] & 0 & \cdots & 0 \\ 0 & \Pr[o|b_2] & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \Pr[o|b_n] \end{bmatrix} \tag{3.8}$$

Here $\beta(x)$ and $\beta(xo)$ are the coefficients associated to history x of length t under B_t and history xo of length $t+1$ under B_{t+1} respectively. Recall that, in particular, these coefficients are such that $\Pr[F_{t+1} \mid B_{t+1}]\beta(xo) = \Pr[F_{t+1} \mid xo]$. By definition of $A_{o,t}$,

$$\begin{aligned}
&\Pr[F_{t+1}|B_{t+1}]A_{o,t} \\
=&\Pr[F_{t+1}|B_{t+1}] \begin{bmatrix} \beta(b_1o) & \beta(b_2o) & \cdots & \beta(b_no) \end{bmatrix} \begin{bmatrix} \Pr[o|b_1] & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \Pr[o|b_n] \end{bmatrix} \\
=&\Pr[F_{t+1}|B_to] \begin{bmatrix} \Pr[o|b_1] & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \Pr[o|b_n] \end{bmatrix} && \text{(by definition of } \beta(b_io)\text{)} \\
=&\Pr[oF_{t+1}|B_t]. && \text{(by Bayes rule)}
\end{aligned}$$

Algorithm 1: Learning low rank distributions using exact conditional probabilities.

```
Set $B_0 = \Lambda_T = \{\varphi\}$.
Set $B_t = \{0^t\}$ where $0^t$ is $(0,\ldots,0)$ with $t$ zeroes for all $t \in \{1,\ldots,T\}$.
Set $\Lambda_t = \{0\}$ or $\{1\}$ to ensure $\Pr[\Lambda_t|B_t] \neq 0$ for all $t \in \{0,\ldots,T-1\}$. [10]
for round $1,2,\ldots$ do
    Choose $\widehat{A}_{o,t}$ for each $o \in \mathscr{O}$ and $t \in [T-1]$ to be any matrix that satisfies
        $\Pr[\Lambda_{t+1}|B_{t+1}]\widehat{A}_{o,t} = \Pr[o\Lambda_{t+1}|B_t]$    (3.9)
    Let $\overline{\Pr}$ be a function defined on observation sequence $(x_1 \ldots x_t)$ for any $t \in [T]$ as,
        $\overline{\Pr}[x_1,\ldots,x_t,\Lambda_t] = \Pr[\Lambda_t|B_t]\widehat{A}_{x_t,t-1}\ldots\widehat{A}_{x_1,0}$    (3.10)
    Sample $n$ sequences $(x_1,\ldots x_t)$ for each length $t \in [T]$ and check if any one of
      these $nT$ sequences is a counterexample, i.e., it satisfies
        $\overline{\Pr}[x_1,\ldots,x_t,\Lambda_t] \neq \Pr[x_1,\ldots,x_t,\Lambda_t]$
    if we find such a counterexample $(x_1,\ldots,x_t)$ then
        Use Proposition 17 to find a time step $\tau \in [t]$, a new test future $\lambda' \in F_\tau$, and a
          new representative history $b' \in H_\tau$. Update $\Lambda_\tau := \Lambda_\tau \cup \{\lambda'\}$ and
          $B_\tau := B_\tau \cup \{b'\}$.
    else
        return $\{\widehat{A}_{o,t}\}_{o\in\mathscr{O},t\in[T-1]}$
```

Since oF_{t+1} is a subset of F_t, by repeatedly applying this equation, we get

$$\Pr[F_T|B_T]A_{x_T,T-1}A_{x_{T-1},T-2}\ldots A_{x_1,0} = \Pr[x_TF_T \mid B_{T-1}]A_{x_{T-1},T-2}\ldots A_{x_1,0} = \Pr[x_1x_2\ldots x_TF_T|B_0]$$

Noting $\Pr[F_T|B_T] = 1$ and $\Pr[x_1x_2\ldots x_TF_T|B_0] = \Pr[x_1x_2\ldots x_T\varphi|\varphi] = \Pr[x_1x_2\ldots x_T]$ as $F_T = B_0 = \{\varphi\}$ completes the proof. □

[10] either $\Pr[0|B_t]$ or $\Pr[1 \mid B_t]$ must be nonzero.

3.4.1 Algorithm

We now present our algorithm (Algorithm 1). The user furnishes ε, the accuracy with which the distribution is to be learned; and δ, a confidence parameter. The parameter n depends on the input and is detailed in the proof of Theorem 3.

As discussed in Section 3.3, the algorithm iteratively builds a set of histories $B_t \subset H_t$ and a set of futures $\Lambda_t \subset F_t$ (for each t), to span the column/row space of $\Pr[F_t \mid H_t]$, respectively. Via Proposition 16, if we can find such sets, they would provide an efficient representation of the distribution. We refer to B_t and Λ_t as *representative* histories and *test* futures, respectively, and as we grow these sets, we maintain the invariant that the matrix $\Pr[\Lambda_t \mid B_t]$ is square and invertible.

We start with B_t, Λ_t of size 1. Then, we repeat the following: motivated by the evolving equation in Proposition 16, we use Equation (3.9) to compute estimates $\widehat{A_{o,t}}$ using our current representative histories and test futures. This may be an under-determined linear system, but Proposition 16 guarantees that it has a solution, and we take $\widehat{A_{o,t}}$ to be any such solution. We use these operators to define our estimate for the distribution, given in Equation (3.10), via iterated multiplication of the operators. Then, we sample several sequences from the distribution and check if any of them certify that our estimate is incorrect, i.e., serve as a counterexample. If we do find a counterexample, then the algorithm finds a time step τ, a new history $b' \in H_\tau$ and a new future $\lambda' \in F_\tau$ that increases the rank of $\Pr[F_\tau \mid B_\tau]$ (this step is described in Proposition 17 below). This can only happen rT times if the distribution has rank r. On the other hand, if we do not find a counterexample, then we simply output our current estimate.

3.4.2 Analysis

We first show how to use a counterexample to improve our set of representative histories and test futures.

Proposition 17 (Finding representative histories and test futures)**.** *If $x_1 \ldots x_t$ is a counterexample,* *that is, it satisfies the following:*

$$\overline{\Pr}[x_1,\ldots,x_t,\Lambda_t] \neq \Pr[x_1,\ldots,x_t,\Lambda_t] \tag{3.11}$$

then we can find a new test future $\lambda' \in F_\tau$ and representative history $b' \in H_\tau$ for $\tau \in [t]$ in at most poly(r,T) time such that $\operatorname{rank}(\Pr[\Lambda_\tau \cup \{\lambda'\}|B_\tau \cup \{b'\}]) = \operatorname{rank}(\Pr[\Lambda_\tau|B_\tau]) + 1$.

Proof. For clarity, in the poof, we abuse notation and do not explicitly mention the sequence length when writing the operator $A_{o,t}$, i.e., we use A_{x_t} instead of $A_{x_t,t-1}$. First, we find a time $\tau \in [t]$ where the following equations hold:

$$\Pr[x_1 \ldots x_\tau \Lambda_\tau] = \Pr[\Lambda_\tau|B_\tau]\widehat{A}_{x_\tau} \ldots \widehat{A}_{x_1}$$
$$\Pr[x_1 \ldots x_\tau x_{\tau+1} \Lambda_{\tau+1}] \neq \Pr[\Lambda_{\tau+1}|B_{\tau+1}]\widehat{A}_{x_{\tau+1}}\widehat{A}_{x_\tau} \ldots \widehat{A}_{x_1}$$

Such a τ must exist because (a) the first equation is true for $\tau = 0$ by definition, and (b) the second equation is true for $\tau = t-1$ because of the counterexample property (Equation (3.11)). Now, we can simplify the equations above by substituting the vector $\mathbf{v} := (\Pr[x_1 \ldots x_\tau])^{-1}\widehat{A}_{x_\tau} \ldots \widehat{A}_{x_1}$ which gives

$$\Pr[\Lambda_\tau|x_1 \ldots x_\tau] = \Pr[\Lambda_\tau|B_\tau]\mathbf{v} \tag{3.12}$$
$$\Pr[x_{\tau+1}\Lambda_{\tau+1}|x_1 \ldots x_\tau] \neq \Pr[\Lambda_{\tau+1}|B_{\tau+1}]\widehat{A}_{x_{\tau+1}}\mathbf{v} = \Pr[x_{\tau+1}\Lambda_{\tau+1}|B_\tau]\mathbf{v}, \tag{3.13}$$

where the last step holds by definition of $\widehat{A}_{x_{\tau+1}}$ (Equation (3.9)). Let $x_{\tau+1}\lambda_{\tau+1}$ index the row of Equation (3.13) where equality does not hold. Define $\lambda' = x_{\tau+1}\lambda_{\tau+1}$ and $b' = x_1 \ldots x_\tau$. We show that the equations above imply that the row vector $\Pr[\lambda'|B'_\tau] := \Pr[x_{\tau+1}\lambda_{\tau+1}|B'_\tau]$ is linearly independent of the rows of $\Pr[\Lambda_\tau|B'_\tau]$. This is enough to prove our claim that $\operatorname{rank}(\Pr[\Lambda'_\tau|B'_\tau]) = \operatorname{rank}(\Pr[\Lambda_\tau|B_\tau]) + 1$.

We establish linear independence by contradiction. Assume that $\Pr[\lambda' \mid B'_\tau]$ is in the span of the rows of $\Pr[\Lambda_\tau \mid B'_\tau]$. Then, there exists a vector $\mathbf{w}$ such that:

$$\Pr[x_{\tau+1}\lambda_{\tau+1}|B'_\tau] = \mathbf{w}^\top \Pr[\Lambda_\tau|B'_\tau]\,. \tag{3.14}$$

Then, we reach a contradiction as

$$\begin{aligned}
\Pr[x_{\tau+1}\lambda_{\tau+1}|x_1\dots x_\tau] &= \mathbf{w}^\top \Pr[\Lambda_\tau|x_1\dots x_\tau] \\
&= \mathbf{w}^\top \Pr[\Lambda_\tau|B_\tau]\mathbf{v} \\
&= \Pr[x_{\tau+1}\lambda_{\tau+1}|B_\tau]\mathbf{v} \\
&\neq \Pr[x_{\tau+1}\lambda_{\tau+1}|x_1\dots x_\tau]
\end{aligned}$$

where the first and third equality follows from linear dependence (Equation (3.14)), the second equality follows from Equation (3.12), and the last inequality follows from Equation (3.13). □

Finally, we need a technical lemma which allows us to estimate the TV distance using conditional samples. This lemma implies that if our algorithm does not find a violation, then with high probability our estimate is close to the true distribution in TV distance.

Proposition 18 (Substitute for TV oracle)**.** *Let* $\Pr[\cdot]$ *and* $\widehat{\Pr}[\cdot]$ *be two probability distributions over observation sequences of length* T*. Suppose that for all* $t \in \{0,\dots,T\}$ *and observations* $o \in \mathscr{O}$

$$\mathbb{E}_{x_1,\dots,x_t\sim\Pr[\cdot]}\left[\left|\widehat{\Pr}[o|x_1,\dots,x_t] - \Pr[o|x_1,\dots,x_t]\right|\right] \leq \varepsilon\,.$$

Then

$$TV(\Pr,\widehat{\Pr}) = \frac{1}{2}\sum_{x_1,\dots,x_T} |(\Pr[x_{1:T}] - \widehat{\Pr}[x_{1:T}])| \leq \frac{(T+1)|O|\varepsilon}{2}$$

Since, $\overline{\Pr}[\cdot]$ might not be a probability distribution, we need to apply this proposition to a probability distribution $\widehat{\Pr}[\cdot]$ that is close to $\overline{\Pr}[\cdot]$, which can be obtained by a simple construction. These details and the proof of Proposition 18 are relatively straightforward and deferred to [55].

Algorithm 2: Learning low rank distributions using conditional samples.

```
for sequence length $t = 0, 1, 2, \ldots, T$ do
    Build set $B_t = \{b_1, \ldots, b_n\}$ of $n$ observation sequences of length $t$ using Lemma 20.
    Build empirical estimates $\widehat{q}(bo)$ and $\widehat{\Sigma}_{B_t}$ (defined in Equation (3.17) and Equation (3.16)) for each history $b \in B_t$, observations $o \in \mathscr{O}$ with $m$ conditional samples (see [55] for details).
    Compute SVD of $\widehat{\Sigma}_{B_t}$.
    Let $\widehat{V}_t$ be the matrix of eigenvectors corresponding to eigenvalues $> \Delta/2$.
    Compute coefficients $\widehat{\beta}(b'_i o)$ for each observation $o \in \mathscr{O}$ and sequence $b'_i \in B_{t-1}$ by solving:
        $$\widehat{\beta}(b'_i o) = \text{argmin}_z \|\widehat{\Sigma}_{B_t} z - \widehat{q}(b'_i o)\|_2^2 + \lambda \|z\|_2^2.$$
    Compute model parameters $\widehat{A}_{o,t-1}$ for each observation $o \in \mathscr{O}$:
        $$\widehat{A}_{o,t-1} = \widehat{V}_t \widehat{V}_t^\top \left[\widehat{\beta}(b'_1 o) \quad \cdots \quad \widehat{\beta}(b'_n o)\right] \begin{bmatrix} \widehat{\Pr}[o|b'_1] & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & \widehat{\Pr}[o|b'_n] \end{bmatrix} \widehat{V}_{t-1} \widehat{V}_{t-1}^\top. \tag{3.15}$$
Return model parameters $\{\widehat{A}_{o,t}\}$.
```

3.5 Learning with conditional samples

In this section, we prove Theorem 4

Theorem 4 (Learning with conditional samples)**.** *Let* $\Pr[\cdot]$ *be any rank r distribution over observation sequences of length T. Assume distribution* $\Pr[\cdot]$ *has fidelity* Δ^*. *Pick any* $0 < \varepsilon, \delta < 1$. *Then Algorithm 2 with access to a conditional sampling oracle runs in* poly(*r, T, O,* $1/\Delta^*$*,* $1/\varepsilon$*,* $\log(1/\delta)$) *time and returns an efficiently represented approximation* $\widehat{\Pr}[\cdot]$ *satisfying* $\mathrm{TV}(\Pr, \widehat{\Pr}) \leq \varepsilon$ *with probability at least* $1 - \delta$.

Throughout this section, we use the same notation as Section 3.3, the set of futures $F_t := \mathscr{O}^{T-t}$.

3.5.1 Algorithm

Algorithm pseudocode is displayed in Algorithm 2. The user furnishes ε, the accuracy with which the distribution is to be learned; δ, a confidence parameter; Δ^*, the fidelity of the distribution and r, the rank of the distribution. The parameters Δ, λ, n and m are detailed in the proof of Theorem 4 in [55].

As with the previous algorithm, Algorithm 2 relies on the efficient representation provided by Proposition 16. First, the algorithm finds basis histories B_t for each $t \in [T]$. As discussed in Section 3.3, under the fidelity assumption, this is not particularly challenging and can be done by sampling from the distribution. The remaining steps in the algorithm constitute a specialized technique for estimating the operators $A_{o,t-1}$ specified in Proposition 16.

Our estimate $\widehat{A}_{o,t-1}$ is based on the formula for $A_{o,t-1}$ given in Equation (3.8) and involves three components: (a) projection onto (an estimate of) the row space of $\Pr[F_t \mid B_t]$, (b) estimates of coefficients $\beta(bo)$ and (c) estimates of probabilities $\Pr[o \mid b]$, where the latter two are for $b_i \in B_{t-1}$. Item (c) is straightforward using conditional samples. For item (a), we define the "preconditioned matrix"

$$\Sigma_{B_t} := \Pr[F_t \mid B_t]^\top D_t^{-1} \Pr[F_t \mid B_t], \tag{3.16}$$

where D_t is a $|F_t| \times |F_t|$ diagonal matrix with $d_t(f) = \frac{1}{|B_t|} \sum_{b \in B_t} \Pr[f \mid b]$ on the diagonal. We show in [55], how this matrix can be estimated using conditional samples. We project onto the principal subspace of the estimated matrix, i.e., onto the span of the eigenvectors with eigenvalue larger than $\Delta/2$. These projections help with error propagation, as it eliminates errors that leave the principal subspace.

For item (b), we estimate the coefficients $\beta(bo)$ for $b \in B_{t-1}$ via linear regression. Using our preconditioner and the definition of the coefficients, we can see that the coefficients satisfy:

$$q(bo) = \Sigma_{B_t}\beta(bo) \qquad \text{where} \qquad q(bo) := \Pr[F_t \mid B_t]^\top D_t^{-1} \Pr[F_t \mid bo] \tag{3.17}$$

As with Σ_{B_t}, $q(bo)$ can also be estimated using conditional samples (via the approach in [55]). Moreover, our basis B_t will ensure that $\|\beta(bo)\|_2$ is bounded by a universal constant, so we can use ridge regression to find estimates $\widehat{\beta}(bo)$. Then we can plug these into Equation (3.15) to obtain estimates $\widehat{A}_{o,t-1}$. We return these matrices as the representation of our estimated distribution.

3.5.2 Analysis

In the previous setting, when we had access to exact conditional probability oracle, the main challenge was finding the bases. In contrast, now that we can only obtain samples, even if we know the bases, we can only learn operators $A_{o,t}$ approximately. As discussed in Section 3.3, controlling estimation errors will require the notion of *robust bases*, which we define next.

Definition 19 (Robust bases)**.** Bases $\{B_t\}_{t\in[T]}$ for distribution $\Pr[\cdot]$ are Δ-*robust* if for every $t \in [T]$:

$$\sigma_+\left(\Pr[F_t|B_t]^\top D_t^{-1} \Pr[F_t|B_t]\right) \geq \Delta$$

where $\sigma_+(M)$ denotes the minimum non-zero eigenvalue of M and D_t is a diagonal matrix of size $|F_t| \times |F_t|$ with entries $d_t(f) := \frac{1}{|B_t|}\sum_{b\in B_t}\Pr[f|b]$ on the diagonal.

A priori, it is unclear if such bases exists for arbitrary low rank distributions. Moreover, even if robust bases exist, how do we find them? Our first lemma show how to find robust bases for high fidelity distributions (Definition 13).

Lemma 20 (Finding robust bases)**.** *Assume distribution* $\Pr[\cdot]$ *has rank r and fidelity* Δ^**. Pick* $0 < \delta < 1$*. Let* $n = O(\Delta^{*-8}\log(r/\delta T))$ *and* $\Delta = \Omega(\Delta^{*-11/2}\log(r/\delta T))$*. For each* $t \in [T]$*, let* S_t

be a random sample of size n of observation sequences of length t from distribution $\Pr[\cdot]$. *Then, with probability* $1-\delta$, $\{S_t\}_{t\in[T]}$ *form* Δ*-robust bases for* $\Pr[\cdot]$.

We provide a proof in [55]. According to this lemma, a random sample from a high fidelity distribution forms a robust basis for each t. With access to a Δ-robust basis B_t, we turn to the issues of estimation and error analysis. First we study estimation of the preconditioned quantities $q(bo)$ and Σ_{B_t} used by the algorithm. Note that all entries of these vectors and matrices are of the following form, where $b^* \in B_t$ and x is a history of length t:

$$s(b^*,x) = \sum_{f\in F_t} \frac{\Pr[f|b^*]\Pr[f|x]}{d(f)}\,.$$

We show such quantities can be estimated efficiently using conditional samples.

Lemma 21 (Estimating preconditioned quantities)**.** *Let* $\{B_t\}_{t\in[T]}$ *be bases for distribution* $\Pr[\cdot]$ *where* $\max_{0\le t\le T}|B_t| \le n$. *Pick any* $0<\varepsilon,\delta<1$. *Fix* $b^*\in B_t$ *and* $x\in H_t$. *Then we can build estimate* $\widehat{s}(b^*,x)$ *in poly*$(n,|\mathscr{O}|,T,1/\varepsilon,\log(1/\delta))$ *time such that with probability* $1-\delta$,

$$|s(b^*,x)-\widehat{s}(b^*,x)| \le \varepsilon\,.$$

We provide the estimation algorithm and a proof in [55]. Using this lemma, we can estimate the operators $A_{o,t}$ via Equation (3.15). The next lemma provides a precise characterization of the estimation error for these operators.

Lemma 22 (Estimating operators)**.** *Assume the distribution* $\Pr[\cdot]$ *has rank r and that* $\{B_t\}_{t\in[T]}$ *are* Δ*-robust bases. Pick* $0<\varepsilon,\delta<1$. *Then, we can learn approximations* $\widehat{A}_{o,t}$ *for all observations* $o\in\mathscr{O}$ *and* $t\in[T]$ *in poly(r,* $|\mathscr{O}|$*, T,* $1/\varepsilon$*,* $1/\Delta$*,* $\log(1/\delta)$*) time such that with probability* $1-\delta$*, for any unit vector v*

$$(\widehat{A}_{o,t}-A_{o,t})v = \beta(B_{t+1})\alpha(o,v)+V_{t+1}^{\perp}\alpha^{\perp}(o,v),$$

where $\beta(B_{t+1})$ *is a matrix with columns* $\beta(b)$ *for* $b \in B_{t+1}$, $V_{t+1}^{\perp}$ *is a matrix whose columns form an orthonormal basis for the kernel of* $\Pr[F_{t+1}|B_{t+1}]^{\top} D_{t+1}^{-1} \Pr[F_{t+1}|B_{t+1}]$, *and the vectors* $\alpha(o,v)$ *and* $\alpha^{\perp}(o,v)$ *are* ℓ_1 *bounded, i.e.,*

$$\max(\|\alpha(o,v)\|_1, \|\alpha^{\perp}(o,v)\|_1) \leq \varepsilon .$$

We provide the proof in [55]. As noted in Section 3.3, the main technical challenge is in analyzing how the estimation error propagates to errors in induced distributions. Using this structured error, we can show how to bound the TV distance between the induced distributions.

Lemma 23 (Perturbation argument)**.** *Assume for each sequence length* $t \in [T]$ *and observation* $o \in \mathscr{O}$, *we have an operator* $\widehat{A}_{o,t}$ *which is close to* $A_{o,t}$ *as defined above in Lemma 22. Let* $\widehat{\Pr}[\cdot]$ *be a function over observation sequences of length* T *given by*

$$\widehat{\Pr}[x_1 \dots x_T] = \widehat{A}_{x_T,T-1} \dots \widehat{A}_{x_1,0}$$

Then, the functions $\Pr[\cdot]$ *and* $\widehat{\Pr}[\cdot]$ *are close in TV distance:*

$$\mathrm{TV}(\Pr, \widehat{\Pr}) \leq 2|\mathscr{O}|T\varepsilon$$

This makes up the most technical component of our proof, and we give the formal proofs in [55]. Together with previous lemmas, this proves our main theorem, Theorem 4.

3.6 Discussion

In this chapter we show how interactive access to hidden Markov models (and more generally low rank distributions) can circumvent computational barriers to efficient learning. In particular, we show that all low rank distributions with a certain fidelity property can be efficiently learned assuming access to a conditional sampling oracle. In [55], we show that fidelity captures the assumptions considered in prior work on (non-interactive) learning of HMMs, specifically:

- Parity with noise admits bases B_t each of cardinality 2 with fidelity $(1-2\alpha^2)/2$, where α is the noise parameter.

- Full rank HMMs, where $\mathbb{T}$ and $\mathbb{O}$ are full column rank, admit bases of size O with fidelity bounded by the minimum singular value of the second moment matrix $\Pr[\mathbf{x}_2 = \cdot, \mathbf{x}_1 = \cdot]$. This parameter also appears polynomially in the analysis of [46].

- The overcomplete setting of [88], where sequences of length $\log S$ are used for estimation, admits bases of size S with fidelity $1/\mathrm{poly}(S)$, matching their parameters.

Despite this, the reliance on the fidelity parameter is the main limitation of our results. We believe this dependence is not necessary, which leads to the main open problem, Open Problem 11. We close the chapter with some final remarks regarding this open problem.

As we have mentioned previously, although fidelity greatly simplifies the basis finding aspect of our algorithm, it is not necessary for this part and refer the reader to [55] where we give a general algorithm for basis finding. Indeed the only place where fidelity is required is in our error propagation analysis, where our techniques require that operators $\widehat{A}_{o,t}$ are estimated in ℓ_2 norm. In the general case, we will only be able to learn operators in the directions for which the preconditioned matrix has large eigenvalues, and ideally we should be able to ignore the directions with small eigenvalues. This strategy would work if we can show that ignoring the small directions preserves the low rank property, which is the linear-algebraic analog of approximating an HMM by one with fewer states. Unfortunately, we do not know if the latter holds, and we believe this is the key challenge to resolving Open Problem 11. We look forward to further progress on this problem.

Acknowledgements. Chapter 3, in part is currently being prepared for submission for publication of the material. Sham M. Kakade, Akshay Krishnamurthy, Gaurav Mahajan and Cyril Zhang. *Learning Hidden Markov Models Using Conditional Samples*. The dissertation author was the primary investigator and author of this material.

Chapter 4

Understanding algorithms in practice

4.1 Preliminaries

A (finite) Markov Decision Process (MDP) $M = (\mathscr{S}, \mathscr{A}, P, r, \gamma, \rho)$ is specified by: a finite state space $\mathscr{S}$; a finite action space $\mathscr{A}$; a transition model P where $P(s'|s,a)$ is the probability of transitioning into state s' upon taking action a in state s; a reward function $r : \mathscr{S} \times \mathscr{A} \to [0,1]$ where $r(s,a)$ is the immediate reward associated with taking action a in state s; a discount factor $\gamma \in [0,1)$; a starting state distribution ρ over $\mathscr{S}$.

A deterministic, stationary policy $\pi : \mathscr{S} \to \mathscr{A}$ specifies a decision-making strategy in which the agent chooses actions adaptively based on the current state, i.e., $a_t = \pi(s_t)$. The agent may also choose actions according to a stochastic policy $\pi : \mathscr{S} \to \Delta(\mathscr{A})$ (where $\Delta(\mathscr{A})$ is the probability simplex over $\mathscr{A}$), and, overloading notation, we write $a_t \sim \pi(\cdot|s_t)$.

A policy induces a distribution over trajectories $\tau = (s_t, a_t, r_t)_{t=0}^{\infty}$, where s_0 is drawn from the starting state distribution ρ, and, for all subsequent timesteps t, $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$. The value function $V^\pi : \mathscr{S} \to \mathbb{R}$ is defined as the discounted sum of future rewards starting at state s and executing π, i.e.

$$V^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | \pi, s_0 = s\right],$$

where the expectation is with respect to the randomness of the trajectory τ induced by π in M.

Since we assume that $r(s,a) \in [0,1]$, we have $0 \leq V^\pi(s) \leq \frac{1}{1-\gamma}$. We overload notation and define $V^\pi(\rho)$ as the expected value under the initial state distribution ρ, i.e.

$$V^\pi(\rho) := \mathbb{E}_{s_0 \sim \rho}[V^\pi(s_0)].$$

The action-value (or Q-value) function $Q^\pi : \mathscr{S} \times \mathscr{A} \to \mathbb{R}$ and the *advantage* function $A^\pi : \mathscr{S} \times \mathscr{A} \to \mathbb{R}$ are defined as:

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) | \pi, s_0 = s, a_0 = a\right], \quad A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s).$$

The goal of the agent is to find a policy π that maximizes the expected value from the initial state, i.e. the optimization problem the agent seeks to solve is:

$$\max_\pi V^\pi(\rho), \tag{4.1}$$

where the max is over all policies. The famous theorem of [16] shows there exists a policy $\pi^\star$ which simultaneously maximizes $V^\pi(s_0)$, for all states $s_0 \in \mathscr{S}$.

Policy Parameterizations. This work studies ascent methods for the optimization problem:

$$\max_{\theta \in \Theta} V^{\pi_\theta}(\rho),$$

where $\{\pi_\theta | \theta \in \Theta\}$ is some class of parametric (stochastic) policies. We consider a number of different policy classes. The first two are *complete* in the sense that any stochastic policy can be represented in the class. The final class may be restrictive. These classes are as follows:

- *Direct parameterization:* The policies are parameterized by

$$\pi_\theta(a|s) = \theta_{s,a}, \tag{4.2}$$

where $\theta \in \Delta(\mathscr{A})^{|\mathscr{S}|}$, i.e. θ is subject to $\theta_{s,a} \geq 0$ and $\sum_{a \in \mathscr{A}} \theta_{s,a} = 1$ for all $s \in \mathscr{S}$ and $a \in \mathscr{A}$.

- *Softmax parameterization:* For unconstrained $\theta \in \mathbb{R}^{|\mathscr{S}||\mathscr{A}|}$,

$$\pi_\theta(a|s) = \frac{\exp(\theta_{s,a})}{\sum_{a' \in \mathscr{A}} \exp(\theta_{s,a'})}. \tag{4.3}$$

  The softmax parameterization is also complete.

- *Restricted parameterizations:* We also study parametric classes $\{\pi_\theta | \theta \in \Theta\}$ that may not contain all stochastic policies. In particular, we pay close attention to both log-linear policy classes and neural policy classes (see [6]). Here, the best we may hope for is an agnostic result where we do as well as the best policy in this class.

While the softmax parameterization is the more natural parametrization among the two complete policy classes, it is also informative to consider the direct parameterization.

Non-Concavity. It is worth explicitly noting that $V^{\pi_\theta}(s)$ is non-concave in θ for both the direct and the softmax parameterizations, so the standard tools of convex optimization are not applicable. For completeness, we formalize this as follows (with a proof in [6], along with an example in Figure 4.1):

Lemma 24. *There is an MDP M (described in Figure 4.1) such that the optimization problem $V^{\pi_\theta}(s)$ is not concave for both the direct and softmax parameterizations.*

Policy gradients. In order to introduce these methods, it is useful to define the discounted state visitation distribution $d^\pi_{s_0}$ of a policy π as:

$$d^\pi_{s_0}(s) := (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr^\pi(s_t = s | s_0), \tag{4.4}$$

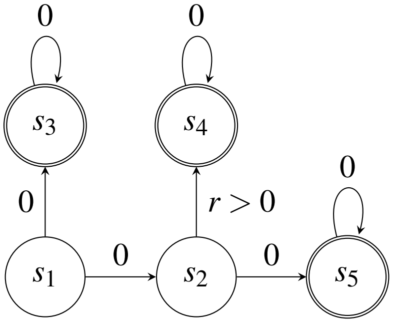

Figure 4.1. (Non-concavity example) A deterministic MDP corresponding to Lemma 24 where $V^{\pi_\theta}(s)$ is not concave. Numbers on arrows represent the rewards for each action.

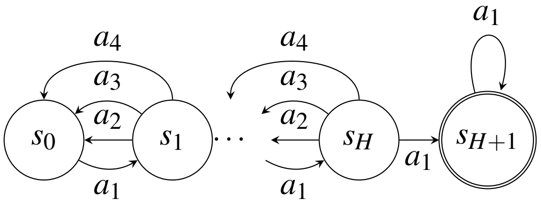

Figure 4.2. (Vanishing gradient example) A deterministic, chain MDP of length $H+2$. We consider a policy where $\pi(a|s_i) = \theta_{s_i,a}$ for $i = 1,2,\ldots,H$. Rewards are 0 everywhere other than $r(s_{H+1},a_1) = 1$. See Proposition 28.

where $\Pr^\pi(s_t = s|s_0)$ is the state visitation probability that $s_t = s$, after we execute π starting at state s_0. Again, we overload notation and write:

$$d_\rho^\pi(s) = \mathbb{E}_{s_0\sim\rho}\left[d_{s_0}^\pi(s)\right] ,$$

where d_ρ^π is the discounted state visitation distribution under initial distribution ρ.

The policy gradient functional form (see e.g. [99, 91]) is then:

$$\nabla_\theta V^{\pi_\theta}(s_0) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{s_0}^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\big[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a)\big]. \tag{4.5}$$

Furthermore, if we are working with a differentiable parameterization of $\pi_\theta(\cdot|s)$ that explicitly constrains $\pi_\theta(\cdot|s)$ to be in the simplex, i.e. $\pi_\theta \in \Delta(\mathscr{A})^{|\mathscr{S}|}$ for all θ, then we also have:

$$\nabla_\theta V^{\pi_\theta}(s_0) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{s_0}^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\big[\nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s,a)\big]. \tag{4.6}$$

Note the above gradient expression (Equation 4.6) does not hold for the direct parameterization, while Equation 4.5 is valid. [1]

[1] This is due to $\sum_a \nabla_\theta \pi_\theta(a|s) = 0$ not explicitly being maintained by the direct parameterization.

The performance difference lemma. The following lemma is helpful throughout:

Lemma 25. *(The performance difference lemma [54]) For all policies π, π' and states s_0,*

$$V^{\pi}(s_0) - V^{\pi'}(s_0) \quad = \quad \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{s_0}^{\pi}}\mathbb{E}_{a\sim\pi(\cdot|s)}\left[A^{\pi'}(s,a)\right].$$

For completeness, we provide a proof in [6].

The distribution mismatch coefficient. We often characterize the difficulty of the exploration problem faced by our policy optimization algorithms when maximizing the objective $V^{\pi}(\mu)$ through the following notion of *distribution mismatch coefficient*.

Definition 26 (Distribution mismatch coefficient)**.** Given a policy π and measures $\rho, \mu \in \Delta(\mathscr{S})$, we refer to $\left\|\frac{d_\rho^\pi}{\mu}\right\|_\infty$ as the *distribution mismatch coefficient* of π relative to μ. Here, $\frac{d_\rho^\pi}{\mu}$ denotes componentwise division.

We often instantiate this coefficient with μ as the initial state distribution used in a policy optimization algorithm, ρ as the distribution to measure the sub-optimality of our policy (this is the start state distribution of interest), and where π above is often chosen to be $\pi^\star \in \text{argmax}_{\pi\in\Pi} V^{\pi}(\rho)$, given a policy class Π.

Notation. Following convention, we use $V^\star$ and $Q^\star$ to denote $V^{\pi^\star}$ and $Q^{\pi^\star}$ respectively. For iterative algorithms which obtain policy parameters $\theta^{(t)}$ at iteration t, we let $\pi^{(t)}$, $V^{(t)}$ and $A^{(t)}$ denote the corresponding quantities parameterized by $\theta^{(t)}$, i.e. $\pi_{\theta^{(t)}}$, $V^{\theta^{(t)}}$ and $A^{\theta^{(t)}}$, respectively. For vectors u and v, we use $\frac{u}{v}$ to denote the componentwise ratio; $u \geq v$ denotes a componentwise inequality; we use the standard convention where $\|v\|_2 = \sqrt{\sum_i v_i^2}$, $\|v\|_1 = \sum_i |v_i|$, and $\|v\|_\infty = \max_i |v_i|$.

Table 4.1. Iteration Complexities with Exact Gradients for the Tabular Case: A summary of the number of iterations required by different algorithms to find a policy π such that $V^\star(s_0) - V^\pi(s_0) \leq \varepsilon$ for some fixed s_0, assuming access to *exact policy gradients*.

Algorithm	Iteration complexity
Projected Gradient Ascent on Simplex (Thm 5)	$O\left(\frac{D_\infty^2 \lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert}{(1-\gamma)^6\varepsilon^2}\right)$
Policy Gradient, softmax parameterization (Thm 6)	asymptotic
Policy Gradient + log barrier regularization, softmax parameterization (Cor 33)	$O\left(\frac{D_\infty^2 \lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert^2}{(1-\gamma)^6\,\varepsilon^2}\right)$
Natural Policy Gradient (NPG), softmax parameterization (Thm 8)	$\frac{2}{(1-\gamma)^2\varepsilon}$

4.2 Our results

This chapter focuses on first-order and quasi second-order policy gradient methods which directly work in the space of some parameterized policy class (rather than value-based approaches). We characterize the computational, approximation, and sample size properties of these methods in the context of a discounted Markov Decision Process (MDP). We focus on: 1) *tabular policy parameterizations*, where there is one parameter per state-action pair so the policy class is complete in that it contains the optimal policy, and 2) *function approximation*, where we have a restricted class or parametric policies which may not contain the globally optimal policy. Note that policy gradient methods for discrete action MDPs work in the space of stochastic policies, which permits the policy class to be differentiable. We now discuss our contributions in the both of these contexts.

Tabular case: We consider three algorithms: two of which are first order methods, projected gradient ascent (on the simplex) and gradient ascent (with a softmax policy parameterization); and the third algorithm, natural policy gradient ascent, can be viewed as a quasi second-order method (or preconditioned first-order method). Table 4.1 summarizes our main results in this case: upper bounds on the number of iterations taken by these algorithms to find an ε-optimal policy, when we have access to exact policy gradients.

Arguably, the most natural starting point for an analysis of policy gradient methods is to consider directly doing gradient ascent on the policy simplex itself and then to project back onto the simplex if the constraint is violated after a gradient update; we refer to this algorithm as projected gradient ascent on the simplex. Using a notion of gradient domination [79], our results provably show that any first-order stationary point of the value function results in an approximately optimal policy, under certain regularity assumptions; this allows for a global convergence analysis by directly appealing to standard results in the non-convex optimization literature.

A more practical and commonly used parameterization is the softmax parameterization, where the simplex constraint is explicitly enforced by the exponential parameterization, thus avoiding projections. This work provides the first global convergence guarantees using only first-order gradient information for the widely-used softmax parameterization. Our first result for this parameterization establishes the asymptotic convergence of the policy gradient algorithm; the analysis challenge here is that the optimal policy (which is deterministic) is attained by sending the softmax parameters to infinity.

In order to establish a finite time convergence rate to optimality for the softmax parameterization, we then consider a *log barrier* regularizer and provide an iteration complexity bound that is polynomial in all relevant quantities. Our use of the log barrier regularizer is critical to avoiding the issue of gradients becomingly vanishingly small at suboptimal near-deterministic policies, an issue of significant practical relevance. The log barrier regularizer can also be viewed as using a *relative* entropy regularizer; here, we note the general approach of entropy based regularization is common in practice (e.g. see [100, 68, 77, 3, 7]). One notable distinction, which we discuss later, is that our analysis is for the log barrier regularization rather than the entropy regularization.

For these aforementioned algorithms, our convergence rates depend on the optimization measure having coverage over the state space, as measured by the *distribution mismatch coefficient* D_∞ (see Table 4.1 caption). In particular, for the convergence rates shown in Table 4.1 (for the aforementioned algorithms), we assume that the optimization objective is the expected (discounted) cumulative value where the initial state is sampled under some distribution, and D_∞ is a measure of the coverage of this initial distribution. Furthermore, we provide a lower bound that shows such a dependence is unavoidable for first-order methods, even when exact gradients are available.

We then consider the Natural Policy Gradient (NPG) algorithm [53] (also see [15, 78]), which can be considered a quasi second-order method due to the use of its particular preconditioner, and provide an iteration complexity to achieve an ε-optimal policy that is at most $\frac{2}{(1-\gamma)^2\varepsilon}$ iterations, improving upon the previous related results of [35, 41] (see Section 4.3). Note the convergence rate has *no* dependence on the number of states or the number of actions, nor does it depend on the distribution mismatch coefficient D_∞. We provide a simple and concise proof for the convergence rate analysis by extending the approach developed in [35], which uses a mirror descent style of analysis [73, 27] and also handles the non-concavity of the policy optimization problem.

This fast and dimension free convergence rate shows how the variable preconditioner in the natural gradient method improves over the standard gradient ascent algorithm. The dimension free aspect of this convergence rate is worth reflecting on, especially given the widespread use of the natural policy gradient algorithm along with variants such as the Trust Region Policy Optimization (TRPO) algorithm [84]; our results may help to provide analysis of a more general family of entropy based algorithms (see for example [76]).

Function Approximation: We now summarize our results with regards to policy gradient methods in the setting where we work with a restricted policy class, which may not contain the optimal policy. In this sense, these methods can be viewed as approximate methods. Table 4.2 provides

Table 4.2. Overview of Approximate Methods: The suboptimality, $V^\star(s_0) - V^\pi(s_0)$, after T iterations for various approximate algorithms, which use different notions of approximation error (sample complexities are not directly considered but instead may be thought of as part of ε_1 and $\varepsilon_{\text{stat}}$).

Algorithm	**Suboptimality after T Iterations**	**Relevant Quantities**
Approx. Value/Policy Iteration [17]	$\frac{\varepsilon_\infty}{(1-\gamma)^2} + \frac{\gamma^T}{(1-\gamma)^2}$	ε_∞: ℓ_∞ error of values
Approx. Value/Policy Iteration, with concentrability [72, 10]	$\frac{C_\infty \varepsilon_1}{(1-\gamma)^2} + \frac{\gamma^T}{(1-\gamma)^2}$	ε_1: an ℓ_1 average error C_∞: concentrability (max density ratio)
Conservative Policy Iteration [54]	$\frac{D_\infty \varepsilon_1}{(1-\gamma)^2} + \frac{1}{(1-\gamma)\sqrt{T}}$	ε_1: an ℓ_1 average error D_∞: max density ratio to opt., $D_\infty \leq C_\infty$
Natural Policy Gradient [6]	$\sqrt{\frac{\kappa \varepsilon_{\text{stat}} + D_\infty \varepsilon_{\text{approx}}}{(1-\gamma)^3}} + \frac{1}{(1-\gamma)\sqrt{T}}$	$\varepsilon_{\text{stat}}$: excess risk $\varepsilon_{\text{approx}}$: approx. error κ: a condition number D_∞: max density ratio to opt., $D_\infty \leq C_\infty$

a summary along with the comparisons to some relevant approximate dynamic programming methods.

A long line of work in the function approximation setting focuses on mitigating the worst-case "ℓ_∞" guarantees that are inherent to approximate dynamic programming methods [17] (see the first row in Table 4.2). The reason to focus on average case guarantees is that it supports the applicability of *supervised machine learning* methods to solve the underlying approximation problem. This is because supervised learning methods, like classification and regression, typically have bounds on the expected error under a distribution, as opposed to worst-case guarantees over all possible inputs.

The existing literature largely consists of two lines of provable guarantees that attempt to mitigate the explicit ℓ_∞ error conditions of approximate dynamic programming: those methods which utilize a problem dependent parameter (the concentrability coefficient [72]) to provide more refined dynamic programming guarantees (e.g. see [72, 92, 10, 36]) and those which work with a restricted policy class, making incremental updates, such as Conservative Policy Iteration (CPI) [54, 82], Policy Search by Dynamic Programming (PSDP) [14], and MD-MPI [41]. Both styles of approaches give guarantees based on worst-case density ratios, i.e. they depend on a maximum ratio between two different densities over the state space. As discussed in[81], the assumptions in the latter class of algorithms are substantially weaker, in that the worst-case density ratio only depends on the state visitation distribution of an optimal policy (also see Table 4.2 caption and Section 4.3).

With regards to function approximation, our main contribution is in providing performance bounds that, in some cases, have milder dependence on these density ratios. We precisely quantify an *approximation/estimation* error decomposition relevant for the analysis of the natural gradient method; this decomposition is stated in terms of the *compatible function approximation error* as introduced in [91]. More generally, we quantify our function approximation results in terms of a precisely quantified transfer error notion, based on approximation error under *distribution shift*. Table 4.2 shows a special case of our convergence rates of NPG, which is governed by four quantities: $\varepsilon_{\text{stat}}$, $\varepsilon_{\text{approx}}$, κ, and D_∞.

Let us discuss the important special case of log-linear policies (i.e. policies that take the softmax of linear functions in a given feature space) where the relevant quantities are as follows: $\varepsilon_{\text{stat}}$ is a bound on the excess risk (the estimation error) in fitting linearly parameterized value functions, which can be driven to 0 with more samples (at the usual statistical rate of $O(1/\sqrt{N})$ where N is the number of samples); $\varepsilon_{\text{approx}}$ is the usual notion of average squared approximation error where the target function may not be perfectly representable by a linear function; κ can be upper bounded with an inverse dependence on the minimal eigenvalue of the feature covariance matrix of the fitting measure (as such it can be viewed as a dimension dependent quantity but not necessarily state dependent); and D_∞ is as before.

For the realizable case, where all policies have values which are linear in the given features (such as in linear MDP models of [52, 101, 49]), we have that the approximation error $\varepsilon_{\text{approx}}$ is 0. Here, our guarantees yield a fully polynomial and sample efficient convergence guarantee, provided the condition number κ is bounded. Importantly, there always exists a good (universal) initial measure that ensures κ is bounded by a quantity that is only polynomial in the dimension of the features, d, as opposed to an explicit dependence on the size of the (infinite) state space (see [6]). Such a guarantee would not be implied by algorithms which depend on the coefficients C_∞ or D_∞.[2]

Our results are also suggestive that a broader class of incremental algorithms — such as CPI [54], PSDP [14], and MD-MPI [41] which make small changes to the policy from one iteration to the next — may also permit a sharper analysis, where the dependence of worst-case density ratios can be avoided through an appropriate approximation/estimation decomposition; this is an interesting direction for future work (a point which we return to in Section 4.6). One significant advantage of NPG is that the explicit parametric policy representation in NPG (and other policy gradient methods) leads to a succinct policy representation in comparison to CPI, PSDP, or related boosting-style methods [82], where the representation complexity of the policy of the latter class of methods grows linearly in the number of iterations (since these methods add one policy to the ensemble per iteration). This representation complexity is likely why the latter class of algorithms are less widely used in practice.

4.3 Related work

We now discuss related work, roughly in the order which reflects our presentation of results in the previous section.

For the direct policy parameterization in the tabular case, we make use of a gradient domination-like property, namely any first-order stationary point of the policy value is approxi-

[2] Bounding C_∞ would require a restriction on the dynamics of the MDP (see [28] and Section 4.3). Bounding D_∞ would require an initial state distribution that is constructed using knowledge of $\pi^\star$, through $d^{\pi^\star}$. In contrast, κ can be made $O(d)$, with an initial state distribution that only depends on the geometry of the features (and does not depend on any other properties of the MDP). See [6].

mately optimal up to a distribution mismatch coefficient. A variant of this result also appears in Theorem 2 of [82], which itself can be viewed as a generalization of the approach in [54]. In contrast to CPI [54] and the more general boosting-based approach in [82], we phrase this approach as a Polyak-like gradient domination property [79] in order to directly allow for the transfer of any advances in non-convex optimization to policy optimization in RL. More broadly, it is worth noting the global convergence of policy gradients for Linear Quadratic Regulators [37] also goes through a similar proof approach of gradient domination.

Empirically, the recent work of [7] studies entropy based regularization and shows the value of regularization in policy optimization, even with exact gradients. This is related to our use of the log barrier regularization.

For our convergence results of the natural policy gradient algorithm in the tabular setting, there are close connections between our results and the works of [35, 41]. [35] provides provable online regret guarantees in changing MDPs utilizing experts algorithms (also see [75, 1]); as a special case, their MDP Experts Algorithm is equivalent to the natural policy gradient algorithm with the softmax policy parameterization. While the convergence result due to [35] was not specifically designed for this setting, it is instructive to see what it implies due to the close connections between optimization and regret [27, 86]. The Mirror Descent-Modified Policy Iteration (MD-MPI) algorithm [41] with negative entropy as the Bregman divergence results is an identical algorithm as NPG for softmax parameterization in the tabular case; Corollary 3 [41] applies to our updates, leading to a bound worse by a $1/(1-\gamma)$ factor and also has logarithmic dependence on $|\mathscr{A}|$. Our proof for this case is concise and may be of independent interest. Also worth noting is the Dynamic Policy Programming of [13], which is an actor-critic algorithm with a softmax parameterization; this algorithm, even though not identical, comes with similar guarantees in terms of its rate (it is weaker in terms of an additional $1/(1-\gamma)$ factor) than the NPG algorithm.

We now turn to function approximation, starting with a discussion of iterative algorithms which make incremental updates in which the next policy is effectively constrained to be close to the previous policy, such as in CPI and PSDP [14]. Here, the work in [82] show how CPI is part of broader family of boosting-style methods. Also, with regards to PSDP, the work in [81] shows how PSDP actually enjoys an improved iteration complexity over CPI, namely $O(\log 1/\varepsilon_{\text{opt}})$ vs. $O(1/\varepsilon_{\text{opt}}^2)$. It is worthwhile to note that both NPG and projected gradient ascent are also incremental algorithms.

We now discuss the approximate dynamic programming results characterized in terms of the concentrability coefficient. Broadly we use the term approximate dynamic programming to refer to fitted value iteration, fitted policy iteration and more generally generalized policy iteration schemes such as classification-based policy iteration as well, in addition to the classical approximate value/policy iteration works. While the approximate dynamic programming results typically require ℓ_∞ bounded errors, which is quite stringent, the notion of concentrability (originally due to [71, 72]) permits sharper bounds in terms of average case function approximation error, provided that the concentrability coefficient is bounded (e.g. see [72, 92, 10, 64]). [28] provide a more detailed discussion on this quantity. Based on this problem dependent constant being bounded, [72, 92], [10] and [64] provide meaningful sample size and error bounds for approximate dynamic programming methods, where there is a data collection policy (under which value-function fitting occurs) that induces a concentrability coefficient. In terms of the concentrability coefficient C_∞ and the "distribution mismatch coefficient" D_∞ in Table 4.2 , we have that $D_\infty \leq C_\infty$, as discussed in [81] (also see the table caption). Also, as discussed in [28], a finite concentrability coefficient is a restriction on the MDP dynamics itself, while a bounded D_∞ does not require any restrictions on the MDP dynamics. The more refined quantities defined by [36] (for the approximate policy iteration result) partially alleviate some of these concerns, but their assumptions still implicitly constrain the MDP dynamics, like the finiteness of the concentrability coefficient.

Assuming bounded concentrability coefficient, there are a notable set of provable average case guarantees for the MD-MPI algorithm [41] (see also [13, 83]), which are stated in terms of various norms of function approximation error. MD-MPI is a class of algorithms for approximate planning under regularized notions of optimality in MDPs. Specifically, [41] analyze a family of actor-critic style algorithms, where there are both approximate value functions updates and approximate policy updates. As a consequence of utilizing approximate value function updates for the critic, the guarantees of [41] are stated with dependencies on concentrability coefficients.

When dealing with function approximation, computational and statistical complexities are relevant because they determine the effectiveness of approximate updates with finite samples. With regards to sample complexity, the work in [92, 10] provide finite sample rates (as discussed above), further generalized to actor-critic methods in [13, 83]. In our policy optimization approach, the analysis of both computational and statistical complexities are straightforward, since we can leverage known statistical and computational results from the stochastic approximation literature; in particular, we use the stochastic projected gradient ascent to obtain a simple, linear time method for the critic estimation step in the natural policy gradient algorithm.

In terms of the algorithmic updates for the function approximation setting, our development of NPG bears similarity to the natural actor-critic algorithm [78], for which some asymptotic guarantees under finite concentrability coefficients are obtained in [19]. While both updates seek to minimize the compatible function approximation error, we perform streaming updates based on stochastic optimization using Monte Carlo estimates for values. In contrast [78] utilize Least Squares Temporal Difference methods [24] to minimize the loss. As a consequence, their updates additionally make linear approximations to the value functions in order to estimate the advantages; our approach is flexible in allowing for wide family of smoothly differentiable policy classes (including neural policies).

Finally, we remark on some concurrent works. The work of [18] provides gradient domination-like conditions under which there is (asymptotic) global convergence to the optimal policy. Their results are applicable to the projected gradient ascent algorithm; they are not applicable to gradient ascent with the softmax parameterization (see the discussion in Section 4.5 herein for the analysis challenges). [18] also provide global convergence results beyond MDPs. Also, [67] provide an analysis of the TRPO algorithm [84] with neural network parameterizations, which bears resemblance to our natural policy gradient analysis. In particular, [67] utilize ideas from both [35] (with a mirror descent style of analysis) along with [26] (to handle approximation with neural networks) to provide conditions under which TRPO returns a near optimal policy. [67] do not explicitly consider the case where the policy class is not complete (i.e when there is approximation). Another related work of [87] considers the TRPO algorithm and provides theoretical guarantees in the tabular case; their convergence rates with exact updates are $O(1/\sqrt{T})$ for the (unregularized) objective function of interest; they also provide faster rates on a modified (regularized) objective function. They do not consider the case of infinite state spaces and function approximation. The closely related recent papers [1, 2] also consider closely related algorithms to the Natural Policy Gradient approach studied here, in an infinite horizon, average reward setting. Specifically, the EE-POLITEX algorithm is closely related to the Q-NPG algorithm which we study in [6], though our approach is in the discounted setting. We adopt the name Q-NPG to capture its close relationship with the NPG algorithm, with the main difference being the use of function approximation for the Q-function instead of advantages. We refer the reader to [6] for more discussion of the technical differences between the two works.

4.4 Warmup: Constrained tabular parameterization

Our starting point is, arguably, the simplest first-order method: we directly take gradient ascent updates on the policy simplex itself and then project back onto the simplex if the constraints are violated after a gradient update. This algorithm is projected gradient ascent on the direct policy parametrization of the MDP, where the parameters are the state-action probabilities, i.e. $\theta_{s,a} = \pi_\theta(a|s)$ (see (4.2)). As noted in Lemma 24, $V^{\pi_\theta}(s)$ is non-concave in the parameters π_θ. Here, we first prove that $V^{\pi_\theta}(\mu)$ satisfies a Polyak-like gradient domination condition [79], and this tool helps in providing convergence rates. The basic approach was also used in the analysis of CPI [54]; related gradient domination-like lemmas also appeared in [82].

It is instructive to consider this special case due to the connections it makes to the non-convex optimization literature. We also provide a lower bound that rules out algorithms whose runtime appeals to the curvature of saddle points (e.g. [74, 40, 50]).

For the direct policy parametrization where $\theta_{s,a} = \pi_\theta(a|s)$, the gradient is:

$$\frac{\partial V^\pi(\mu)}{\partial \pi(a|s)} = \frac{1}{1-\gamma} d_\mu^\pi(s) Q^\pi(s,a), \tag{4.7}$$

using (4.5). In particular, for this parameterization, we may write $\nabla_\pi V^\pi(\mu)$ instead of $\nabla_\theta V^{\pi_\theta}(\mu)$.

4.4.1 Gradient domination

Informally, we say a function $f(\theta)$ satisfies a gradient domination property if for all $\theta \in \Theta$,

$$f(\theta^\star) - f(\theta) = O(G(\theta)),$$

where $\theta^\star \in \operatorname{argmax}_{\theta' \in \Theta} f(\theta')$ and where $G(\theta)$ is some suitable scalar notion of first-order stationarity, which can be considered a measure of how large the gradient is (see [58, 23, 11]). Thus if one can find a θ that is (approximately) a first-order stationary point, then the parameter θ will be near optimal (in terms of function value). Such conditions are a standard device to establishing global convergence in non-convex optimization, as they effectively rule out the presence of bad critical points. In other words, given such a condition, quantifying the convergence rate for a specific algorithm, like say projected gradient ascent, will require quantifying the rate of its convergence to a first-order stationary point, for which one can invoke standard results from the optimization literature.

The following lemma shows that the direct policy parameterization satisfies a notion of gradient domination. This is the basic approach used in the analysis of CPI [54]; a variant of this lemma also appears in [82]. We give a proof for completeness.

Even though we are interested in the value $V^\pi(\rho)$, it is helpful to consider the gradient with respect to another state distribution $\mu \in \Delta(\mathscr{S})$.

Lemma 27 (Gradient domination)**.** *For the direct policy parameterization (as in* (4.2)*), for all state distributions $\mu, \rho \in \Delta(\mathscr{S})$, we have*

$$\begin{aligned} V^\star(\rho) - V^\pi(\rho) &\leq \left\| \frac{d_\rho^{\pi^\star}}{d_\mu^\pi} \right\|_\infty \max_{\bar{\pi}} \, (\bar{\pi} - \pi)^\top \nabla_\pi V^\pi(\mu) \\ &\leq \frac{1}{1-\gamma} \left\| \frac{d_\rho^{\pi^\star}}{\mu} \right\|_\infty \max_{\bar{\pi}} \, (\bar{\pi} - \pi)^\top \nabla_\pi V^\pi(\mu), \end{aligned}$$

where the max is over the set of all policies, i.e. $\bar{\pi} \in \Delta(\mathscr{A})^{|\mathscr{S}|}$.

Before we provide the proof, a few comments are in order with regards to the performance measure ρ and the optimization measure μ. Subtly, note that although the gradient is with respect to $V^\pi(\mu)$, the final guarantee applies to *all* distributions ρ. The significance is that even though we may be interested in our performance under ρ, it may be helpful to optimize under the distribution μ. To see this, note the lemma shows that a sufficiently small gradient magnitude in the feasible directions implies the policy is nearly optimal in terms of its value, but only if the state distribution of π, i.e. d_μ^π, adequately covers the state distribution of some optimal policy $\pi^\star$. Here, it is also worth recalling the theorem of [16] which shows there exists a single policy $\pi^\star$ that is simultaneously optimal for all starting states s_0. Note that the hardness of the exploration problem is captured through the distribution mismatch coefficient (Definition 26).

of Lemma 27. By the performance difference lemma (Lemma 25),

$$\begin{aligned} V^\star(\rho) - V^\pi(\rho) &= \frac{1}{1-\gamma} \sum_{s,a} d_\rho^{\pi^\star}(s) \pi^\star(a|s) A^\pi(s,a) \\ &\leq \frac{1}{1-\gamma} \sum_{s,a} d_\rho^{\pi^\star}(s) \max_{\bar{a}} A^\pi(s,\bar{a}) \\ &= \frac{1}{1-\gamma} \sum_s \frac{d_\rho^{\pi^\star}(s)}{d_\mu^\pi(s)} \cdot d_\mu^\pi(s) \max_{\bar{a}} A^\pi(s,\bar{a}) \\ &\leq \frac{1}{1-\gamma} \left(\max_s \frac{d_\rho^{\pi^\star}(s)}{d_\mu^\pi(s)} \right) \sum_s d_\mu^\pi(s) \max_{\bar{a}} A^\pi(s,\bar{a}), \end{aligned} \tag{4.8}$$

where the last inequality follows since $\max_{\bar{a}} A^{\pi}(s,\bar{a}) \geq 0$ for all states s and policies π. We wish to upper bound (4.8). We then have:

$$\begin{aligned}
\sum_s \frac{d_\mu^\pi(s)}{1-\gamma} \max_{\bar{a}} A^\pi(s,\bar{a}) &= \max_{\bar{\pi}\in\Delta(\mathscr{A})^{|\mathscr{S}|}} \sum_{s,a} \frac{d_\mu^\pi(s)}{1-\gamma} \bar{\pi}(a|s) A^\pi(s,a) \\
&= \max_{\bar{\pi}\in\Delta(\mathscr{A})^{|\mathscr{S}|}} \sum_{s,a} \frac{d_\mu^\pi(s)}{1-\gamma} (\bar{\pi}(a|s) - \pi(a|s)) A^\pi(s,a) \\
&= \max_{\bar{\pi}\in\Delta(\mathscr{A})^{|\mathscr{S}|}} \sum_{s,a} \frac{d_\mu^\pi(s)}{1-\gamma} (\bar{\pi}(a|s) - \pi(a|s)) Q^\pi(s,a) \\
&= \max_{\bar{\pi}\in\Delta(\mathscr{A})^{|\mathscr{S}|}} (\bar{\pi}-\pi)^\top \nabla_\pi V^\pi(\mu)
\end{aligned}$$

where the first step follows since $\max_{\bar{\pi}}$ is attained at an action which maximizes $A^\pi(s,\cdot)$ (per state); the second step follows as $\sum_a \pi(a|s) A^\pi(s,a) = 0$; the third step uses $\sum_a (\bar{\pi}(a|s) - \pi(a|s)) V^\pi(s) = 0$ for all s; and the final step follows from the gradient expression (see (4.7)). Using this in (4.8),

$$\begin{aligned}
V^\star(\rho) - V^\pi(\rho) &\leq \left\| \frac{d_\rho^{\pi^\star}}{d_\mu^\pi} \right\|_\infty \max_{\bar{\pi}\in\Delta(\mathscr{A})^{|\mathscr{S}|}} (\bar{\pi}-\pi)^\top \nabla_\pi V^\pi(\mu) \\
&\leq \frac{1}{1-\gamma} \left\| \frac{d_\rho^{\pi^\star}}{\mu} \right\|_\infty \max_{\bar{\pi}\in\Delta(\mathscr{A})^{|\mathscr{S}|}} (\bar{\pi}-\pi)^\top \nabla_\pi V^\pi(\mu).
\end{aligned}$$

where the last step follows due to $\max_{\bar{\pi}\in\Delta(\mathscr{A})^{|\mathscr{S}|}} (\bar{\pi}-\pi)^\top \nabla_\pi V^\pi(\mu) \geq 0$ for any policy π and $d_\mu^\pi(s) \geq (1-\gamma)\mu(s)$ (see (4.4)). □

In a sense, the use of an appropriate μ circumvents the issues of strategic exploration. It is natural to ask whether this additional term is necessary, a question which we return to. First, we provide a convergence rate for the projected gradient ascent algorithm.

4.4.2 Convergence rates for projected gradient ascent

Using this notion of gradient domination, we now give an iteration complexity bound for projected gradient ascent over the space of stochastic policies, i.e. over $\Delta(\mathscr{A})^{|\mathscr{S}|}$. The projected gradient ascent algorithm updates

$$\pi^{(t+1)} = P_{\Delta(\mathscr{A})^{|S|}}(\pi^{(t)} + \eta \nabla_\pi V^{(t)}(\mu)), \tag{4.9}$$

where $P_{\Delta(\mathscr{A})^{|\mathscr{S}|}}$ is the projection onto $\Delta(\mathscr{A})^{|\mathscr{S}|}$ in the Euclidean norm.

Theorem 5. *The projected gradient ascent algorithm* (4.9) *on* $V^\pi(\mu)$ *with stepsize* $\eta = \frac{(1-\gamma)^3}{2\gamma|\mathscr{A}|}$ *satisfies for all distributions* $\rho \in \Delta(\mathscr{S})$*,*

$$\min_{t<T} \left\{ V^\star(\rho) - V^{(t)}(\rho) \right\} \leq \varepsilon \quad \textit{whenever} \quad T > \frac{64\gamma|\mathscr{S}||\mathscr{A}|}{(1-\gamma)^6\varepsilon^2} \left\| \frac{d_\rho^{\pi^\star}}{\mu} \right\|_\infty^2 .$$

A proof is provided in [6]. The proof first invokes a standard iteration complexity result of projected gradient ascent to show that the gradient magnitude with respect to all feasible directions is small. More concretely, we show the policy is ε-stationary[3], that is, for all $\pi_\theta + \delta \in \Delta(\mathscr{A})^{|\mathscr{S}|}$ and $\|\delta\|_2 \leq 1$, $\delta^\top \nabla_\pi V^{\pi_\theta}(\mu) \leq \varepsilon$. We then use Lemma 27 to complete the proof.

Note that the guarantee we provide is for the best policy found over the T rounds, which we obtain from a bound on the average norm of the gradients. This type of a guarantee is standard in the non-convex optimization literature, where an average regret bound cannot be used to extract a single good solution, e.g. by averaging. In the context of policy optimization, this is not a serious limitation as we collect on-policy trajectories for each policy in doing sample-based gradient estimation, and these samples can be also used to estimate the policy's value. Note that the evaluation step is not required for every policy, and can also happen on a schedule, though we still need to evaluate $O(T)$ policies to obtain the convergence rates described here.

[3] See [6] for discussion on this definition.

4.4.3 Lower bound: Vanishing gradients and saddle points

To understand the necessity of the distribution mismatch coefficient in Lemma 27 and Theorem 5, let us first give an informal argument that some condition on the state distribution of π, or equivalently μ, is necessary for stationarity to imply optimality. For example, in a sparse-reward MDP (where the agent is only rewarded upon visiting some small set of states), a policy that does not visit *any* rewarding states will have zero gradient, even though it is arbitrarily suboptimal in terms of values. Below, we give a more quantitative version of this intuition, which demonstrates that even if π chooses all actions with reasonable probabilities (and hence the agent will visit all states if the MDP is connected), then there is an MDP where a large fraction of the policies π have vanishingly small gradients, and yet these policies are highly suboptimal in terms of their value.

Concretely, consider the chain MDP of length $H+2$ shown in Figure 4.2. The starting state of interest is state s_0 and the discount factor $\gamma = H/(H+1)$. Suppose we work with the direct parameterization, where $\pi_\theta(a|s) = \theta_{s,a}$ for $a = a_1, a_2, a_3$ and $\pi_\theta(a_4|s) = 1 - \theta_{s,a_1} - \theta_{s,a_2} - \theta_{s,a_3}$. Note we do not over-parameterize the policy. For this MDP and policy structure, if we were to initialize the probabilities over actions, say deterministically, then there is an MDP (obtained by permuting the actions) where all the probabilities for a_1 will be less than $1/4$.

The following result not only shows that the gradient is exponentially small in H, it also shows that many higher order derivatives, up to $O(H/\log H)$, are also exponentially small in H.

Proposition 28 (Vanishing gradients at suboptimal parameters)**.** *Consider the chain MDP of Figure 4.2, with $H+2$ states, $\gamma = H/(H+1)$, and with the direct policy parameterization (with $3|\mathscr{S}|$ parameters, as described in the text above). Suppose θ is such that $0 < \theta < 1$ (componentwise) and $\theta_{s,a_1} < 1/4$ (for all states s). For all $k \leq \frac{H}{40\log(2H)} - 1$, we have $\|\nabla_\theta^k V^{\pi_\theta}(s_0)\| \leq (1/3)^{H/4}$, where $\nabla_\theta^k V^{\pi_\theta}(s_0)$ is a tensor of the k_{th} order derivatives of $V^{\pi_\theta}(s_0)$ and the norm is the operator norm of the tensor.[4] Furthermore, $V^\star(s_0) - V^{\pi_\theta}(s_0) \geq (H+1)/8 - (H+1)^2/3^H$.*

[4] The operator norm of a k_{th}-order tensor $J \in \mathbb{R}^{d^{\otimes k}}$ is defined as $\sup_{u_1,\ldots,u_k \in \mathbb{R}^d \,:\, \|u_i\|_2 = 1} \langle J, u_1 \otimes \ldots \otimes u_d \rangle$.

This lemma also suggests that results in the non-convex optimization literature, on escaping from saddle points, e.g. [74, 40, 50], do not directly imply global convergence due to that the higher order derivatives are small.

Remark 29. (Exact vs. Approximate Gradients) The chain MDP of Figure 4.2, is a common example where *sample* based estimates of gradients will be 0 under random exploration strategies; there is an exponentially small in H chance of hitting the goal state under a random exploration strategy. Note that this lemma is with regards to *exact* gradients. This suggests that even with exact computations (along with using exact higher order derivatives) we might expect numerical instabilities.

Remark 30. (Comparison with the upper bound) The lower bound does not contradict the upper bound of Theorem 27 (where a small gradient is turned into a small policy suboptimality bound), as the distribution mismatch coefficient, as defined in Definition 26, could be infinite in the chain MDP of Figure 4.2, since the start-state distribution is concentrated on one state only. More generally, for any policy with $\theta_{s,a_1} < 1/4$ in all states s, $\left\| \frac{d_\rho^{\pi^\star}}{d_\rho^{\pi_\theta}} \right\|_\infty = \Omega(4^H)$.

Remark 31. (Comparison with information-theoretic lower bounds) The lower bound here is *not information theoretic*, in that it does not present a hard problem instance for all algorithms. Indeed, exploration algorithms for tabular MDPs starting from E^3 [60], RMAX [25] and several subsequent works yield polynomial sample complexities for the chain MDP. Proposition 28 should be interpreted as a hardness result for the specific class of policy gradient like approaches that search for a policy with a small policy gradient, as these methods will find the initial parameters to be valid in terms of the size of (several orders of) gradients. In particular, it precludes any meaningful claims on global optimality, based just on the size of the policy gradients, without additional assumptions as discussed in the previous remark.

The proof is provided in [6]. The lemma illustrates that lack of good exploration can indeed be detrimental in policy gradient algorithms, since the gradient can be small either due to π being near-optimal, or, simply because π does not visit advantageous states often enough. In this sense, it also demonstrates the necessity of the distribution mismatch coefficient in Lemma 27.

4.5 Softmax tabular parameterization

We now consider the softmax policy parameterization (4.3). Here, we still have a non-concave optimization problem in general, as shown in Lemma 24, though we do show that global optimality can be reached under certain regularity conditions. From a practical perspective, the softmax parameterization of policies is preferable to the direct parameterization, since the parameters θ are unconstrained and standard unconstrained optimization algorithms can be employed. However, optimization over this policy class creates other challenges as we study in this section, as the optimal policy (which is deterministic) is attained by sending the parameters to infinity.

We study three algorithms for this problem. The first performs direct policy gradient ascent on the objective without modification, while the second adds a log barrier regularizer to keep the parameters from becoming too large, as a means to ensure adequate exploration. Finally, we study the natural policy gradient algorithm and establish a global optimality result with no dependence on the distribution mismatch coefficient or dimension-dependent factors.

For the softmax parameterization, the gradient takes the form:

$$\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_\mu^{\pi_\theta}(s) \pi_\theta(a|s) A^{\pi_\theta}(s,a) \tag{4.10}$$

(see [6] for a proof).

4.5.1 Asymptotic convergence, without regularization

Due to the exponential scaling with the parameters θ in the softmax parameterization, *any* policy that is nearly deterministic will have gradients close to 0. In spite of this difficulty, we provide a positive result that gradient ascent asymptotically converges to the global optimum for the softmax parameterization.

The update rule for gradient ascent is:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta V^{(t)}(\mu). \tag{4.11}$$

Theorem 6 (Global convergence for softmax parameterization)**.** *Assume we follow the gradient ascent update rule as specified in Equation* (4.11) *and that the distribution μ is strictly positive i.e. $\mu(s) > 0$ for all states s. Suppose $\eta \leq \frac{(1-\gamma)^3}{8}$, then we have that for all states s, $V^{(t)}(s) \to V^\star(s)$ as $t \to \infty$.*

Remark 32. (Strict positivity of μ and exploration) Theorem 6 assumed that optimization distribution μ was *strictly* positive, i.e. $\mu(s) > 0$ for all states s. We leave it is an open question of whether or not gradient ascent will globally converge if this condition is not met. The concern is that if this condition is not met, then gradient ascent may not globally converge due to that $d_\mu^{\pi_\theta}(s)$ effectively scales down the learning rate for the parameters associated with state s (see (4.10)).

The complete proof is provided in [6]. We now discuss the subtleties in the proof and show why the softmax parameterization precludes a direct application of the gradient domination lemma. In order to utilize the gradient domination property (in Lemma 27), we would desire to show that: $\nabla_\pi V^\pi(\mu) \to 0$. However, using the functional form of the softmax parameterization and (4.7), we have that:

$$\frac{\partial V^{\pi_\theta}(\mu)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_\mu^{\pi_\theta}(s) \pi_\theta(a|s) A^{\pi_\theta}(s,a) = \pi_\theta(a|s) \frac{\partial V^{\pi_\theta}(\mu)}{\partial \pi_\theta(a|s)}.$$

Hence, we see that even if $\nabla_\theta V^{\pi_\theta}(\mu) \to 0$, we are not guaranteed that $\nabla_\pi V^{\pi_\theta}(\mu) \to 0$.

We now briefly discuss the main technical challenges in the proof. The proof first shows that the sequence $V^{(t)}(s)$ is monotone increasing pointwise, i.e. for *every* state s, $V^{(t+1)}(s) \geq$

$V^{(t)}(s)$. This implies the existence of a limit $V^{(\infty)}(s)$ by the monotone convergence theorem. Based on the limiting quantities $V^{(\infty)}(s)$ and $Q^{(\infty)}(s,a)$, which we show exist, define the following limiting sets for each state s:

$$\begin{aligned} I_0^s &:= \{a|Q^{(\infty)}(s,a) = V^{(\infty)}(s)\} \\ I_+^s &:= \{a|Q^{(\infty)}(s,a) > V^{(\infty)}(s)\} \\ I_-^s &:= \{a|Q^{(\infty)}(s,a) < V^{(\infty)}(s)\}. \end{aligned}$$

The challenge is to then show that, for all states s, the set I_+^s is the empty set, which would immediately imply $V^{(\infty)}(s) = V^\star(s)$. The proof proceeds by contradiction, assuming that I_+^s is non-empty. Using that I_+^s is non-empty and that the gradient tends to zero in the limit, i.e. $\nabla_\theta V^{\pi_\theta}(\mu) \to 0$, we have that for all $a \in I_+^s$, $\pi^{(t)}(a|s) \to 0$ (see (4.10)). This, along with the functional form of the softmax parameterization, implies that there must be divergence (in magnitude) among the set of parameters associated with *some* action a at state s, i.e. that $\max_{a\in\mathscr{A}} |\theta_{s,a}^{(t)}| \to \infty$. The primary technical challenge in the proof is to then use this divergence, along with the dynamics of gradient ascent, to show that I_+^s is empty via a contradiction.

We leave it as a question for future work as to characterizing the convergence rate, which we conjecture is exponentially slow in some of the relevant quantities, such as in terms of the size of state space. Here, we turn to a regularization based approach to ensure convergence at a polynomial rate in all relevant quantities.

4.5.2 Polynomial convergence with log barrier regularization

Due to the exponential scaling with the parameters θ, policies can rapidly become near deterministic, when optimizing under the softmax parameterization, which can result in slow convergence. Indeed a key challenge in the asymptotic analysis in the previous section was to handle the growth of the absolute values of parameters as they tend to infinity. A common practical remedy for this is to use entropy-based regularization to keep the probabilities from getting too small [100, 68], and we study gradient ascent on a similarly regularized objective in this section. Recall that the relative-entropy for distributions p and q is defined as: $\mathrm{KL}(p,q) := \mathbb{E}_{x\sim p}[-\log q(x)/p(x)]$. Denote the uniform distribution over a set $\mathscr{X}$ by $\mathrm{Unif}_{\mathscr{X}}$, and define the following log barrier regularized objective as:

$$\begin{aligned} L_\lambda(\theta) &:= V^{\pi_\theta}(\mu) - \lambda\, \mathbb{E}_{s\sim \mathrm{Unif}_{\mathscr{S}}}\Big[\mathrm{KL}(\mathrm{Unif}_{\mathscr{A}}, \pi_\theta(\cdot|s))\Big] \\ &= V^{\pi_\theta}(\mu) + \frac{\lambda}{|\mathscr{S}|\,|\mathscr{A}|}\sum_{s,a}\log \pi_\theta(a|s) + \lambda \log|\mathscr{A}|, \end{aligned} \tag{4.12}$$

where λ is a regularization parameter. The constant (i.e. the last term) is not relevant with regards to optimization. This regularizer is different from the more commonly utilized entropy regularizer as in [68], a point which we return to in Remark 34.

The policy gradient ascent updates for $L_\lambda(\theta)$ are given by:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta L_\lambda(\theta^{(t)}). \tag{4.13}$$

Our next theorem shows that approximate first-order stationary points of the entropy-regularized objective are approximately globally optimal, provided the regularization is sufficiently small.

Theorem 7. *(Log barrier regularization) Suppose θ is such that:*

$$\|\nabla_\theta L_\lambda(\theta)\|_2 \le \varepsilon_{opt}$$

and $\varepsilon_{opt} \le \lambda/(2|\mathscr{S}|\,|\mathscr{A}|)$. Then we have that for all starting state distributions ρ:

$$V^{\pi_\theta}(\rho) \;\ge\; V^\star(\rho) - \frac{2\lambda}{1-\gamma}\left\|\frac{d_\rho^{\pi^\star}}{\mu}\right\|_\infty.$$

Proof. The proof consists of showing that $\max_a A^{\pi_\theta}(s,a) \le 2\lambda/(\mu(s)|\mathscr{S}|)$ for all states. To see that this is sufficient, observe that by the performance difference lemma (Lemma 25),

$$\begin{aligned}
V^\star(\rho) - V^{\pi_\theta}(\rho) &= \frac{1}{1-\gamma} \sum_{s,a} d_\rho^{\pi^\star}(s) \pi^\star(a|s) A^{\pi_\theta}(s,a) \\
&\leq \frac{1}{1-\gamma} \sum_{s} d_\rho^{\pi^\star}(s) \max_{a \in \mathscr{A}} A^{\pi_\theta}(s,a) \\
&\leq \frac{1}{1-\gamma} \sum_{s} 2 d_\rho^{\pi^\star}(s) \lambda / (\mu(s) |\mathscr{S}|) \\
&\leq \frac{2\lambda}{1-\gamma} \max_s \left(\frac{d_\rho^{\pi^\star}(s)}{\mu(s)} \right).
\end{aligned}$$

which would then complete the proof.

We now proceed to show that $\max_a A^{\pi_\theta}(s,a) \leq 2\lambda/(\mu(s)|\mathscr{S}|)$. For this, it suffices to bound $A^{\pi_\theta}(s,a)$ for any state-action pair s,a where $A^{\pi_\theta}(s,a) \geq 0$ else the claim is trivially true. Consider an (s,a) pair such that $A^{\pi_\theta}(s,a) > 0$. Using the policy gradient expression for the softmax parameterization,

$$\frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_\mu^{\pi_\theta}(s) \pi_\theta(a|s) A^{\pi_\theta}(s,a) + \frac{\lambda}{|\mathscr{S}|} \left(\frac{1}{|\mathscr{A}|} - \pi_\theta(a|s) \right). \tag{4.14}$$

The gradient norm assumption $\|\nabla_\theta L_\lambda(\theta)\|_2 \leq \varepsilon_{\text{opt}}$ implies that:

$$\begin{aligned}
\varepsilon_{\text{opt}} \geq \frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} &= \frac{1}{1-\gamma} d_\mu^{\pi_\theta}(s) \pi_\theta(a|s) A^{\pi_\theta}(s,a) + \frac{\lambda}{|\mathscr{S}|} \left(\frac{1}{|\mathscr{A}|} - \pi_\theta(a|s) \right) \\
&\geq \frac{\lambda}{|\mathscr{S}|} \left(\frac{1}{|\mathscr{A}|} - \pi_\theta(a|s) \right),
\end{aligned}$$

where we have used $A^{\pi_\theta}(s,a) \geq 0$. Rearranging and using our assumption $\varepsilon_{\text{opt}} \leq \lambda/(2|\mathscr{S}||\mathscr{A}|)$,

$$\pi_\theta(a|s) \geq \frac{1}{|\mathscr{A}|} - \frac{\varepsilon_{\text{opt}}|\mathscr{S}|}{\lambda} \geq \frac{1}{2|\mathscr{A}|}.$$

Solving for $A^{\pi_\theta}(s,a)$ in (4.14), we have:

$$\begin{aligned}
A^{\pi_\theta}(s,a) &= \frac{1-\gamma}{d_\mu^{\pi_\theta}(s)}\left(\frac{1}{\pi_\theta(a|s)}\frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} + \frac{\lambda}{|\mathscr{S}|}\left(1-\frac{1}{\pi_\theta(a|s)|\mathscr{A}|}\right)\right) \\
&\leq \frac{1-\gamma}{d_\mu^{\pi_\theta}(s)}\left(2|\mathscr{A}|\varepsilon_{\text{opt}} + \frac{\lambda}{|\mathscr{S}|}\right) \\
&\leq 2\frac{1-\gamma}{d_\mu^{\pi_\theta}(s)}\frac{\lambda}{|\mathscr{S}|} \\
&\leq 2\lambda/(\mu(s)|\mathscr{S}|),
\end{aligned}$$

where the penultimate step uses $\varepsilon_{\text{opt}} \leq \lambda/(2|\mathscr{S}|\,|\mathscr{A}|)$ and the final step uses $d_\mu^{\pi_\theta}(s) \geq (1-\gamma)\mu(s)$. This completes the proof. □

By combining the above theorem with standard results on the convergence of gradient ascent (to first order stationary points), we obtain the following corollary.

Corollary 33. *(Iteration complexity with log barrier regularization) Let* $\beta_\lambda := \frac{8\gamma}{(1-\gamma)^3} + \frac{2\lambda}{|\mathscr{S}|}$. *Starting from any initial* $\theta^{(0)}$*, consider the updates* (4.13) *with* $\lambda = \frac{\varepsilon(1-\gamma)}{2\left\|\frac{d_\rho^{\pi^\star}}{\mu}\right\|_\infty}$ *and* $\eta = 1/\beta_\lambda$*. Then for all starting state distributions* ρ*, we have*

$$\min_{t<T}\left\{V^\star(\rho) - V^{(t)}(\rho)\right\} \leq \varepsilon \quad \textit{whenever} \quad T \geq \frac{320|\mathscr{S}|^2|\mathscr{A}|^2}{(1-\gamma)^6\varepsilon^2}\left\|\frac{d_\rho^{\pi^\star}}{\mu}\right\|_\infty^2.$$

See [6] for the proof. The corollary shows the importance of balancing how the regularization parameter λ is set relative to the desired accuracy ε, as well as the importance of the initial distribution μ to obtain global optimality.

Remark 34. (Entropy vs. log barrier regularization) The more commonly considered regularizer is the entropy [68] (also see [7] for a more detailed empirical investigation), where the regularizer would be:

$$\frac{1}{|\mathscr{S}|}\sum_s H(\pi_\theta(\cdot|s)) = \frac{1}{|\mathscr{S}|}\sum_s\sum_a -\pi_\theta(a|s)\log\pi_\theta(a|s).$$

Note the entropy is far less aggressive in penalizing small probabilities, in comparison to the log barrier, which is equivalent to the relative entropy. In particular, the entropy regularizer is always bounded between 0 and $\log|\mathscr{A}|$, while the relative entropy (against the uniform distribution over actions), is bounded between 0 and infinity, where it tends to infinity as probabilities tend to 0. We leave it is an open question if a polynomial convergence rate [5] is achievable with the more common entropy regularizer; our polynomial convergence rate using the KL regularizer crucially relies on the aggressive nature in which the relative entropy prevents small probabilities (the proof shows that any action, with a positive advantage, has a significant probability for any near-stationary policy of the regularized objective).

4.5.3 Dimension-free convergence of Natural Policy Gradient Ascent

We now show the Natural Policy Gradient algorithm, with the softmax parameterization (4.3), obtains an improved iteration complexity. The NPG algorithm defines a Fisher information matrix (induced by π), and performs gradient updates in the geometry induced by this matrix as follows:

$$\begin{aligned} F_\rho(\theta) &= \mathbb{E}_{s\sim d_\rho^{\pi_\theta}} \mathbb{E}_{a\sim\pi_\theta(\cdot|s)} \Big[\nabla_\theta \log \pi_\theta(a|s) \Big(\nabla_\theta \log \pi_\theta(a|s)\Big)^\top\Big] \\ \theta^{(t+1)} &= \theta^{(t)} + \eta F_\rho(\theta^{(t)})^\dagger \nabla_\theta V^{(t)}(\rho), \end{aligned} \tag{4.15}$$

where $M^\dagger$ denotes the Moore-Penrose pseudoinverse of the matrix M. Throughout this section, we restrict to using the initial state distribution $\rho \in \Delta(\mathscr{S})$ in our update rule in (4.15) (so our optimization measure μ and the performance measure ρ are identical). Also, we restrict attention to states $s \in \mathscr{S}$ reachable from ρ, since, without loss of generality, we can exclude states that are not reachable under this start state distribution[6].

We leverage a particularly convenient form the update takes for the softmax parameteri-

[5] Here, ideally we would like to be poly in $|\mathscr{S}|$, $|\mathscr{A}|$, $1/(1-\gamma)$, $1/\varepsilon$, and the distribution mismatch coefficient, which we conjecture may not be possible.

[6] Specifically, we restrict the MDP to the set of states $\{s \in \mathscr{S} \ : \ \exists \pi \ \text{ such that } \ d_\rho^\pi(s) > 0\}$.

zation (see [53]). For completeness, we provide a proof in [6].

Lemma 35. *(NPG as soft policy iteration) For the softmax parameterization* (4.3)*, the NPG updates* (4.15) *take the form:*

$$\theta^{(t+1)} = \theta^{(t)} + \frac{\eta}{1-\gamma} A^{(t)} \quad \textit{and} \quad \pi^{(t+1)}(a|s) = \pi^{(t)}(a|s) \frac{\exp(\eta A^{(t)}(s,a)/(1-\gamma))}{Z_t(s)},$$

where $Z_t(s) = \sum_{a \in \mathscr{A}} \pi^{(t)}(a|s) \exp(\eta A^{(t)}(s,a)/(1-\gamma))$.

The updates take a strikingly simple form in this special case; they are identical to the classical multiplicative weights updates [39, 27] for online linear optimization over the probability simplex, where the linear functions are specified by the advantage function of the current policy at each iteration. Notably, there is no dependence on the state distribution $d_\rho^{(t)}$, since the pseudoinverse of the Fisher information cancels out the effect of the state distribution in NPG. We now provide a dimension free convergence rate of this algorithm.

Theorem 8 (Global convergence for NPG)**.** *Suppose we run the NPG updates* (4.15) *using* $\rho \in \Delta(\mathscr{S})$ *and with* $\theta^{(0)} = 0$*. Fix* $\eta > 0$*. For all* $T > 0$*, we have:*

$$V^{(T)}(\rho) \geq V^*(\rho) - \frac{\log|\mathscr{A}|}{\eta T} - \frac{1}{(1-\gamma)^2 T}.$$

In particular, setting $\eta \geq (1-\gamma)^2 \log|\mathscr{A}|$, we see that NPG finds an ε-optimal policy in a number of iterations that is at most:

$$T \leq \frac{2}{(1-\gamma)^2 \varepsilon},$$

which has no dependence on the number of states or actions, despite the non-concavity of the underlying optimization problem.

The proof strategy we take borrows ideas from the online regret framework in changing MDPs (in [35]); here, we provide a faster rate of convergence than the analysis implied by [35] or by [41]. We also note that while this proof is obtained for the NPG updates, it is known in the literature that in the limit of small stepsizes, NPG and TRPO updates are closely related (e.g. see [84, 76, 80]).

First, the following improvement lemma is helpful:

Lemma 36 (Improvement lower bound for NPG)**.** *For the iterates $\pi^{(t)}$ generated by the NPG updates* (4.15)*, we have for all starting state distributions μ*

$$V^{(t+1)}(\mu) - V^{(t)}(\mu) \geq \frac{(1-\gamma)}{\eta}\mathbb{E}_{s\sim\mu}\log Z_t(s) \geq 0.$$

Proof. First, let us show that $\log Z_t(s) \geq 0$. To see this, observe:

$$\begin{aligned}\log Z_t(s) &= \log \sum_a \pi^{(t)}(a|s)\exp(\eta A^{(t)}(s,a)/(1-\gamma)) \\ &\geq \sum_a \pi^{(t)}(a|s)\log\exp(\eta A^{(t)}(s,a)/(1-\gamma)) = \frac{\eta}{1-\gamma}\sum_a \pi^{(t)}(a|s)A^{(t)}(s,a) = 0.\end{aligned}$$

where the inequality follows by Jensen's inequality on the concave function $\log x$ and the final equality uses $\sum_a \pi^{(t)}(a|s)A^{(t)}(s,a) = 0$. Using $d^{(t+1)}$ as shorthand for $d_\mu^{(t+1)}$, the performance difference lemma implies:

$$\begin{aligned}V^{(t+1)}(\mu) - V^{(t)}(\mu) &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{(t+1)}}\sum_a \pi^{(t+1)}(a|s)A^{(t)}(s,a) \\ &= \frac{1}{\eta}\mathbb{E}_{s\sim d^{(t+1)}}\sum_a \pi^{(t+1)}(a|s)\log\frac{\pi^{(t+1)}(a|s)Z_t(s)}{\pi^{(t)}(a|s)} \\ &= \frac{1}{\eta}\mathbb{E}_{s\sim d^{(t+1)}}\mathrm{KL}(\pi_s^{(t+1)}||\pi_s^{(t)}) + \frac{1}{\eta}\mathbb{E}_{s\sim d^{(t+1)}}\log Z_t(s) \\ &\geq \frac{1}{\eta}\mathbb{E}_{s\sim d^{(t+1)}}\log Z_t(s) \geq \frac{1-\gamma}{\eta}\mathbb{E}_{s\sim\mu}\log Z_t(s),\end{aligned}$$

where the last step uses that $d^{(t+1)} = d_\mu^{(t+1)} \geq (1-\gamma)\mu$, componentwise (by (4.4)), and that

$\log Z_t(s) \geq 0.$ □

With this lemma, we now prove Theorem 8.

of Theorem 8. Since ρ is fixed, we use $d^\star$ as shorthand for $d_\rho^{\pi^\star}$; we also use π_s as shorthand for the vector of $\pi(\cdot|s)$. By the performance difference lemma (Lemma 25),

$$\begin{aligned}
V^{\pi^\star}(\rho) - V^{(t)}(\rho) &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\star}\sum_a \pi^\star(a|s)A^{(t)}(s,a) \\
&= \frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\sum_a \pi^\star(a|s)\log\frac{\pi^{(t+1)}(a|s)Z_t(s)}{\pi^{(t)}(a|s)} \\
&= \frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\left(\mathrm{KL}(\pi_s^\star||\pi_s^{(t)}) - \mathrm{KL}(\pi_s^\star||\pi_s^{(t+1)}) + \sum_a \pi^*(a|s)\log Z_t(s)\right) \\
&= \frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\left(\mathrm{KL}(\pi_s^\star||\pi_s^{(t)}) - \mathrm{KL}(\pi_s^\star||\pi_s^{(t+1)}) + \log Z_t(s)\right),
\end{aligned}$$

where we have used the closed form of our updates from Lemma 35 in the second step.

By applying Lemma 36 with $d^\star$ as the starting state distribution, we have:

$$\frac{1}{\eta}\mathbb{E}_{s\sim d^\star}\log Z_t(s) \leq \frac{1}{1-\gamma}\left(V^{(t+1)}(d^\star) - V^{(t)}(d^\star)\right)$$

which gives us a bound on $\mathbb{E}_{s\sim d^\star}\log Z_t(s)$.

Using the above equation and that $V^{(t+1)}(\rho) \geq V^{(t)}(\rho)$ (as $V^{(t+1)}(s) \geq V^{(t)}(s)$ for all states s by Lemma 36), we have:

$$
\begin{aligned}
V^{\pi^\star}(\rho) - V^{(T-1)}(\rho) &\le \frac{1}{T}\sum_{t=0}^{T-1}(V^{\pi^\star}(\rho) - V^{(t)}(\rho)) \\
&\le \frac{1}{\eta T}\sum_{t=0}^{T-1}\mathbb{E}_{s\sim d^\star}(\mathrm{KL}(\pi_s^\star||\pi_s^{(t)}) - \mathrm{KL}(\pi_s^\star||\pi_s^{(t+1)})) + \frac{1}{\eta T}\sum_{t=0}^{T-1}\mathbb{E}_{s\sim d^\star}\log Z_t(s) \\
&\le \frac{\mathbb{E}_{s\sim d^\star}\mathrm{KL}(\pi_s^\star||\pi^{(0)})}{\eta T} + \frac{1}{(1-\gamma)T}\sum_{t=0}^{T-1}\left(V^{(t+1)}(d^\star) - V^{(t)}(d^\star)\right) \\
&= \frac{\mathbb{E}_{s\sim d^\star}\mathrm{KL}(\pi_s^\star||\pi^{(0)})}{\eta T} + \frac{V^{(T)}(d^\star) - V^{(0)}(d^\star)}{(1-\gamma)T} \\
&\le \frac{\log|\mathscr{A}|}{\eta T} + \frac{1}{(1-\gamma)^2 T}.
\end{aligned}
$$

The proof is completed using that $V^{(T)}(\rho) \ge V^{(T-1)}(\rho)$. □

4.6 Discussion

This work provides a systematic study of the convergence properties of policy optimization techniques, both in the tabular and the function approximation settings. At the core, our results imply that the non-convexity of the policy optimization problem is not the fundamental challenge for typical variants of the policy gradient approach. This is evidenced by the global convergence results which we establish and that demonstrate the relative niceness of the underlying optimization problem. At the same time, our results highlight that insufficient exploration can lead to the convergence to sub-optimal policies, as is also observed in practice; technically, we show how this is an issue of conditioning. Conversely, we can expect typical policy gradient algorithms to find the best policy from amongst those whose state-visitation distribution is adequately aligned with the policies we discover, provided a distribution-shifted notion of approximation error is small.

In the tabular case, our results show that the nature and severity of the exploration / distribution mismatch term differs in different policy optimization approaches. For instance, we find that doing policy gradient in its standard form for both the direct and softmax parameterizations can be slow to converge, particularly in the face of distribution mismatch, even when policy gradients are computed exactly. Natural policy gradient, on the other hand, enjoys a fast dimension-free convergence when we are in tabular settings with exact gradients. On the other hand, for the function approximation setting, or when using finite samples, all algorithms suffer to some degree from the exploration issue captured through a conditioning effect.

With regards to function approximation, the guarantees herein are the first provable results that permit average case approximation errors, where the guarantees do not have explicit worst case dependencies over the state space. These worst case dependencies are avoided by precisely characterizing an approximation/estimation error decomposition, where the relevant approximation error is under distribution shift to an optimal policies measure. Here, we see that successful function approximation relies on two key aspects: good conditioning (related to exploration) and low distribution-shifted, approximation error. In particular, these results identify the relevant measure of the expressivity of a policy class, for the natural policy gradient.

With regards to sample size issues, we showed that simply using stochastic (projected) gradient ascent suffices for accurate policy optimization. However, in terms of improving sample efficiency and polynomial dependencies, there are number of important questions for future research, including variance reduction techniques along with data re-use.

There are number of compelling directions for further study. The first is in understanding how to remove the density ratio guarantees among prior algorithms; our results are suggestive that the incremental policy optimization approaches, including CPI [54], PSDP [14], and MD-MPI [41], may permit such an improved analysis. The question of understanding what representations are robust to distribution shift is well-motivated by the nature of our distribution-shifted, approximation error (the transfer error). Finally, we hope that policy optimization approaches can be combined with exploration approaches, so that, provably, these approaches can retain their robustness properties (in terms of their agnostic learning guarantees) while mitigating the need for a well conditioned initial starting distribution.

Acknowledgements. Chapter 4 contains a reprint of the material as it appears in Con-

ference on Learning Theory (COLT 2020). Alekh Agarwal, Sham M. Kakade, Jason D. Lee, Gaurav Mahajan. *Optimality and approximation with policy gradient methods in markov decision processes*. The dissertation author was the primary investigator and author of this paper.

Bibliography

[1] Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. POLITEX: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702, 2019.

[2] Yasin Abbasi-Yadkori, Nevena Lazic, Csaba Szepesvari, and Gellert Weisz. Exploration-enhanced politex. *arXiv preprint arXiv:1908.10479*, 2019.

[3] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.

[4] Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *arXiv preprint arXiv:2006.10814*, 2020.

[5] Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *33rd Conference on Learning Theory*, COLT 2020.

[6] Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. In *Journal of Machine Learning Research*, JMLR 2021.

[7] Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans, editors. *Understanding the impact of entropy on policy optimization*, 2019.

[8] Michael Alekhnovich. More on average case vs approximation complexity. In *Symposium on Foundations of Computer Science*, 2003.

[9] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of machine learning research*, 2014.

[10] András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

[11] Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

[12] Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin F Yang. Model-based reinforcement learning with value-targeted regression. *arXiv:2006.01107*, 2020.

[13] Mohammad Gheshlaghi Azar, Vicenç Gómez, and Hilbert J. Kappen. Dynamic policy programming. *J. Mach. Learn. Res.*, 13(1), November 2012.

[14] J. A. Bagnell, Sham M Kakade, Jeff G. Schneider, and Andrew Y. Ng. Policy search by dynamic programming. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 831–838. MIT Press, 2004.

[15] J. Andrew Bagnell and Jeff Schneider. Covariant policy search. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 1019–1024, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[16] Richard Bellman and Stuart Dreyfus. Functional approximations and dynamic programming. *Mathematical Tables and Other Aids to Computation*, 13(68):247–251, 1959.

[17] Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

[18] Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *CoRR*, abs/1906.01786, 2019.

[19] Shalabh Bhatnagar, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor–critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

[20] Robi Bhattacharjee and Gaurav Mahajan. Learning what to remember. In *33rd International Conference on Algorithmic Learning Theory*, ALT 2022.

[21] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology*, 1994.

[22] Avrim Blum and Joel Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 1995.

[23] Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.

[24] Justin A Boyan. Least-squares temporal difference learning. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 49–56. Morgan Kaufmann Publishers Inc., 1999.

[25] Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231, 2003.

[26] Qi Cai, Zhuoran Yang, Jason D. Lee, and Zhaoran Wang. Neural temporal-difference learning converges to global optima. *CoRR*, abs/1905.10027, 2019.

[27] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

[28] Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, 2019.

[29] Mary Cryan, Leslie Ann Goldberg, and Paul W Goldberg. Evolutionary trees can be learned in polynomial time in the two-state general Markov model. *SIAM Journal on Computing*, 2001.

[30] Shi Dong, Benjamin Van Roy, and Zhengyuan Zhou. Provably efficient reinforcement learning with aggregated states, 2020.

[31] Simon Du, Sham Kakade, Jason Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. In *38th International Conference on Machine Learning*, ICML 2021.

[32] Simon Du, Jason Lee, Gaurav Mahajan, and Ruosong Wang. Agnostic q-learning with function approximation in deterministic systems. In *Advances in Neural Information Processing Systems*, NeurIPS 2020.

[33] Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2020.

[34] Simon S Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudík, and John Langford. Provably efficient RL with rich observations via latent state decoding. In *International Conference on Machine Learning*, 2019.

[35] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

[36] Amir-massoud Farahmand, Csaba Szepesvári, and Rémi Munos. Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems*, pages 568–576, 2010.

[37] Maryam Fazel, Rong Ge, Sham M Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. *arXiv preprint arXiv:1801.05039*, 2018.

[38] Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

[39] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[40] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points - online stochastic gradient for tensor decomposition. *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, 2015.

[41] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. *arXiv preprint arXiv:1901.11275*, 2019.

[42] Max Hopkins, Daniel Kane, Shachar Lovett, and Gaurav Mahajan. Noise-tolerant, reliable active classification with comparison queries. In *33rd Conference on Learning Theory*, COLT 2020.

[43] Max Hopkins, Daniel Kane, Shachar Lovett, and Gaurav Mahajan. Realizable learning is all you need. In *35th Conference on Learning Theory*, COLT 2022.

[44] Max Hopkins, Daniel Kane, Shachar Lovett, and Gaurav Mahajan. Point location and active learning: Learning halfspaces almost optimally. In *IEEE 61st Annual Symposium on Foundations of Computer Science*, FOCS 2020.

[45] Max Hopkins, Daniel M Kane, Shachar Lovett, and Gaurav Mahajan. Do pac-learners learn the marginal distribution? *arXiv preprint arXiv:2302.06285*, 2023.

[46] Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences*, 2012.

[47] Herbert Jaeger. Observable operator models for discrete stochastic time series. *Neural computation*, 2000.

[48] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire. Contextual decision processes with low bellman rank are pac-learnable, 2016.

[49] 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.

[50] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1724–1732, 2017.

[51] Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *arXiv preprint arXiv:2102.00815*, 2021.

[52] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, 2020.

[53] S. Kakade. A natural policy gradient. In *NIPS*, 2001.

[54] Sham Kakade and John Langford. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2, pages 267–274, 2002.

[55] Sham M. Kakade, Akshay Krishnamurthy, Gaurav Mahajan, and Cyril Zhang. Learning hidden markov models using conditional samples. *arXiv preprint arXiv:2302.14753*, 2023.

[56] Daniel Kane, Sihan Liu, Shachar Lovett, and Gaurav Mahajan. Computational-statistical gap in reinforcement learning. In *35th Conference on Learning Theory*, COLT 2022.

[57] Daniel Kane, Sihan Liu, Shachar Lovett, Gaurav Mahajan, Csaba Szepesvári, and Gellért Weisz. Exponential hardness of reinforcement learning with linear function approximation. *arXiv preprint arXiv:2302.12940*, 2023.

[58] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

[59] Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Symposium on Theory of Computing*, 1994.

[60] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

[61] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[62] Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1848–1856, 2016.

[63] Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, 2020.

[64] Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Analysis of classification-based policy iteration algorithms. *The Journal of Machine Learning Research*, 17(1):583–612, 2016.

[65] Lihong Li. *A Unifying Framework for Computational Reinforcement Learning Theory*. PhD thesis, USA, 2009. AAI3386797.

[66] Michael L Littman, Richard S Sutton, and Satinder P Singh. Predictive representations of state. In *NIPS*, volume 14, page 30, 2001.

[67] Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural proximal/trust region policy optimization attains globally optimal policy. *CoRR*, abs/1906.10306, 2019.

[68] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[69] Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *Conference on Artificial Intelligence and Statistics*, 2020.

[70] Elchanan Mossel and Sébastien Roch. Learning nonsingular phylogenies and hidden markov models. In *Symposium on Theory of Computing*, 2005.

[71] Rémi Munos. Error bounds for approximate policy iteration. In *ICML*, volume 3, pages 560–567, 2003.

[72] Rémi Munos. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[73] Arkadii Semenovich Nemirovsky and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[74] Yurii Nesterov and Boris T. Polyak. Cubic regularization of newton method and its global performance. *Math. Program.*, pages 177–205, 2006.

[75] Gergely Neu, Andras Antos, András György, and Csaba Szepesvári. Online markov decision processes under bandit feedback. In *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., 2010.

[76] Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *CoRR*, abs/1705.07798, 2017.

[77] Jan Peters, Katharina Mülling, and Yasemin Altün. Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)*, pages 1607–1612. AAAI Press, 2010.

[78] Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomput.*, 71(7-9):1180–1190, 2008.

[79] B. T. Polyak. Gradient methods for minimizing functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

[80] Aravind Rajeswaran, Kendall Lowrey, Emanuel V. Todorov, and Sham M Kakade. Towards generalization and simplicity in continuous control. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6550–6561. Curran Associates, Inc., 2017.

[81] Bruno Scherrer. Approximate policy iteration schemes: A comparison. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14. JMLR.org, 2014.

[82] Bruno Scherrer and Matthieu Geist. Local policy search in a convex space and conservative policy iteration as boosted policy search. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2014.

[83] Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, and Matthieu Geist. Approximate modified policy iteration and its application to the game of tetris. *Journal of Machine Learning Research*, 16:1629–1676, 2015.

[84] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[85] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[86] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[87] Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and fa ster rates for regularized mdps, 2019.

[88] Vatsal Sharan, Sham M Kakade, Percy S Liang, and Gregory Valiant. Learning overcomplete HMMs. *Advances in Neural Information Processing Systems*, 2017.

[89] Geelon So, Gaurav Mahajan, and Sanjoy Dasgupta. Convergence of online k-means. In *25th International Conference on Artificial Intelligence and Statistics*, AISTATS 2022.

[90] Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, 2019.

[91] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 99, pages 1057–1063, 1999.

[92] Csaba Szepesvári and Rémi Munos. Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd international conference on Machine learning*, pages 880–887. ACM, 2005.

[93] L G Valiant and V V Vazirani. Np is as easy as detecting unique solutions. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, page 458–463, 1985.

[94] Yuanhao Wang, Ruosong Wang, and Sham M. Kakade. An exponential lower bound for linearly-realizable mdps with constant suboptimality gap, 2021.

[95] Gellért Weisz, Philip Amortila, and Csaba Szepesvári. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132, pages 1237–1264. PMLR, 2021.

[96] Gellért Weisz, András György, Tadashi Kozuno, and Csaba Szepesvári. Confident approximate policy iteration for efficient local planning in q^{π}-realizable mdps. In *Advances in Neural Information Processing Systems*, 2022.

[97] Gellért Weisz, Csaba Szepesvári, and András György. Tensorplan and the few actions lower bound for planning in mdps under linear realizability of optimal value functions. In *International Conference on Algorithmic Learning Theory*, pages 1097–1137. PMLR, 2022.

[98] Gellért Weisz, Philip Amortila, Barnabás Janzer, Yasin Abbasi-Yadkori, Nan Jiang, and Csaba Szepesvári. On query-efficient planning in mdps under linear realizability of the optimal state-value function, 2021.

[99] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[100] Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

[101] Lin Yang and Mengdi Wang. Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning*, 2019.

[102] Dong Yin, Botao Hao, Yasin Abbasi-Yadkori, Nevena Lazić, and Csaba Szepesvári. Efficient local planning with linear function approximation. In *International Conference on Algorithmic Learning Theory*, pages 1165–1192. PMLR, 2022.

[103] Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error, 2020.

[104] Dongruo Zhou, Jiafan He, and Quanquan Gu. Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning*, pages 12793–12802, 2021.