# Exploration and Apprenticeship Learning in Reinforcement Learning

**Pieter Abbeel** PABBEEL@CS.STANFORD.EDU
**Andrew Y. Ng** ANG@CS.STANFORD.EDU
Computer Science Department, Stanford University Stanford, CA 94305, USA

## Abstract

We consider reinforcement learning in systems with unknown dynamics. Algorithms such as $E^3$ (Kearns and Singh, 2002) learn near-optimal policies by using "exploration policies" to drive the system towards poorly modeled states, so as to encourage exploration. But this makes these algorithms impractical for many systems; for example, on an autonomous helicopter, overly aggressive exploration may well result in a crash. In this paper, we consider the apprenticeship learning setting in which a teacher demonstration of the task is available. We show that, given the initial demonstration, no explicit exploration is necessary, and we can attain near-optimal performance (compared to the teacher) simply by repeatedly executing "exploitation policies" that try to maximize rewards. In finite-state MDPs, our algorithm scales polynomially in the number of states; in continuous-state linear dynamical systems, it scales polynomially in the dimension of the state. These results are proved using a martingale construction over relative losses.

## 1. Introduction

The Markov Decision Processes (MDPs) formalism provides a powerful set of tools for modeling and solving control problems, and many algorithms exist for finding (near) optimal solutions for a given MDP (see, e.g., Bertsekas & Tsitsiklis, 1996; Sutton & Barto, 1998). To apply these algorithms to control problems in which the dynamics are not known in advance, the parameters of the MDP typically need to be learned from observations of the system.

A key problem in learning an MDP's parameters is that of *exploration*: How can we ensure that all relevant parts of the MDP are visited sufficiently often that we manage to collect accurate statistics for their state transition probabilities? The state-of-the-art answer to this problem is the $E^3$-algorithm (Kearns & Singh, 2002) (and variants/extensions: Kearns & Koller, 1999; Kakade, Kearns & Langford, 2003; Brafman & Tennenholtz, 2002). These algorithms guarantee that near-optimal performance will be obtained in time polynomial in the number of states of the system. The basic idea of $E^3$ is that it will repeatedly apply an "exploration policy," i.e., one that tries to visit state-action pairs whose transition dynamics are still inaccurately modeled. After a polynomial number of iterations, it will deem itself to have modeled enough of the MDP accurately. Then, it will apply an "exploitation policy," which (given the current MDP model) tries to maximize the sum of rewards obtained over time. In the original $E^3$ work (Kearns & Singh, 2002), the algorithm would explicitly use an exploration policy until the model was considered accurate enough, after which it switched to an exploitation policy. In later variants such as (Brafman & Tennenholtz, 2002) this choice of exploration vs. exploitation policy was made less explicitly, but through a reward scheme reminiscent of "optimism in the face of uncertainty," (e.g., Kaelbling, Littman & Moore, 1996). However, the algorithm still tends to end up generating (and using) exploration policies in its initial stage.

To achieve its performance guarantees, the $E^3$-family of algorithms demand that we run exploration policies on the unknown system until we have an accurate model for the entire MDP (or at least for the "reachable" parts of it). The strong bias towards exploration makes the policies generated by the $E^3$-family often unacceptable for running on a real system. Consider for example running $E^3$ on an autonomous helicopter. This would require executing policies that aggressively explore different parts of the state-space, including parts of it that would lead to crashing the helicopter.[1] As a second example, if the system to be controlled is a chemical plant, $E^3$-generated policies may well cause an explosion in the plant through its aggressive exploration of the entire state space. Despite the strong theoretical results, for many robotics and other applications, we do not believe that $E^3$ is a practical algorithm.

In this paper, we consider the apprenticeship learning setting, in which we have available an initial teacher demonstration of the task to be learned. For example, we may

Appearing in *Proceedings of the $22^{nd}$ International Conference on Machine Learning*, Bonn, Germany, 2005. 

[1] Indeed, in our work on an autonomous helicopter flight, our first crash occurred during (manual flight) exploration, when a human pilot was over-aggressive in exploring the boundaries of the flight envelope (moving the control sticks through their extreme ranges), which placed excessive strain on the rotor head assembly and caused it to disintegrate in mid-air.

have a human pilot give us an initial demonstration of helicopter flight. Given this initial training data with which to learn the dynamics, we show that it suffices to only execute exploitation policies (ones that try to do as well as possible, given the current model of the MDP). More specifically, we propose the following algorithm:

1. Have a teacher demonstrate the task to be learned, and record the state-action trajectories of the teacher's demonstration.
2. Use all state-action trajectories seen so far to learn a dynamics model for the system. For this model, find a (near) optimal policy using any reinforcement learning (RL) algorithm.
3. Test that policy by running it on the real system. If the performance is as good as the teacher's performance, stop. Otherwise, *add the state-action trajectories from the (unsuccessful) test to the training set*, and go back to step 2.

Note that the algorithm we described uses a greedy policy with respect to the current estimated model at every iteration. So there is never an explicit exploration step. In practice, exploitation policies tend to be more benign, and thus we believe this is a significantly more palatable algorithm for many applications. For example, unlike $E^3$, this is a procedure that can much more safely and confidently be tried on an autonomous helicopter.[2] Further, if we are designing a controller for a client and each experiment consumes a non-trivial amount of time/resources, we believe it is much more palatable to tell them that the next policy we try will represent our best attempt at solving their problem—i.e., an exploitation policy that represents our current best attempt at controlling the system—rather than that we will be repeatedly running expensive experiments to slowly gather more and more data about the MDP.

We note that the algorithm proposed above also parallels a reasonably common practice in applied control, in which some initial policy is used to collect data and build a model for a simulator. Then, if subsequently a controller is found that works in simulation but not in real-life, the designer tries (usually manually) to adjust the simulator to make it correctly predict the failure of this policy. If machine learning is used to build the simulator, then a natural way to modify the simulator after observing an unsuccessful policy is to add the data obtained from the unsuccessful policy to the training set. Thus, our work can also be viewed as formally analyzing, and thereby attempting to cast light on, the conditions under which a procedure like this can be expected to lead to a good policy.

Previous work has shown the effectiveness of using teacher or expert demonstrations (called apprenticeship learning, also imitation learning, and learning by watching) in various ways for control. Schaal and Atkeson (1994) and Smart and Kaelbling (2000) both give examples where learning is significantly faster when bootstrapping from a teacher. Their methods are somewhat related in spirit, but different in detail from ours (e.g., Smart and Kaelbling, 2000, uses model-free Q-learning, and does not learn the MDP parameters), and had no formal guarantees.

Other examples include Sammut et al. (1992); Kuniyoshi, Inaba & Inoue (1994); Demiris & Hayes (1994); Amit & Mataric (2002); and Pomerleau (1989), which apply supervised learning to learn a parameterized policy from the demonstrations. In these examples, neither the reward function nor the system dynamics need to be specified since a policy is learned directly as a mapping from the states to the actions. This approach has been applied successfully in a variety of applications, but may require careful selection of an appropriate policy class parameterization, and generally lacks strong performance guarantees. Abbeel and Ng (2004) uses the demonstrations to remove the need for explicitly specifying a reward function; there, the system dynamics were assumed to be known.

In what follows, we prove that, with high probability, our algorithm given above terminates with a policy whose performance is comparable to (or better than) the teacher. In the case of discrete state MDPs, the algorithm scales at most polynomially in the number of states. In the case of linearly parameterized dynamical systems, we use a martingale over relative losses to show that the algorithm scales at most polynomially in the dimension of the state space.

Due to space constraints, most proofs are omitted from this paper or given only as sketches. The complete proofs are given in the full paper Abbeel and Ng (2005).

## 2. Preliminaries

A Markov decision process (MDP) is a tuple $(S, \mathcal{A}, T, H, D, R)$, where S is a set of states; $\mathcal{A}$ is a set of actions/inputs; $T = \{P(\cdot|s,a)\}_{s,a}$ is a set of state transition probabilities (here, $P(\cdot|s,a)$ is the state transition distribution upon taking action $a$ in state $s$); $H$ is the horizon time of the MDP, so that the MDP terminates after $H$ steps;[3] $D$ is a distribution over states from which the initial state $s_0$ is drawn; and $R : S \mapsto \mathbb{R}$ is the reward function, which we assume to be non-negative and bounded by $R_{\max}$. A policy $\pi$ is a mapping from states $S$ to a probability distribution over the set of actions $\mathcal{A}$. The utility of a policy $\pi$ in an MDP $M$ is given by $U_M(\pi) = \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi, M]$. Here the expectation is over all possible state trajectories in the MDP $M$.

Specifying an MDP therefore requires specifying each item

[2] For example, in our autonomous helicopter work, no exploitation policy that we have ever used—out of many dozens—has ever deliberately jerked the helicopter back-and-forth in the manner described in footnote 1.

[3] Any infinite horizon MDP with discounted rewards can be $\epsilon$-approximated by a finite horizon MDP, using a horizon $H_\epsilon = \lceil \log_\gamma(\epsilon(1-\gamma)/R_{\max}) \rceil$.

of the tuple $(S, \mathcal{A}, T, H, D, R)$. In practice, the state transitions probabilities $T$ are usually the most difficult element of this tuple to specify, and must often be learned from data. More precisely, the state space $S$ and action space $\mathcal{A}$ are physical properties of the system being controlled, and thus easily specified. $R$ (and $H$) is typically given by the task specification (or otherwise can be learned from a teacher demonstration, as in Abbeel & Ng, 2004). Finally, $D$ is usually either known or can straightforwardly be estimated from data. Thus, in the sequel, we will assume that $S, \mathcal{A}, H, D$ and $R$ are given, and focus exclusively on the problem of learning the state transition dynamics $T$ of the MDP.

Consider an MDP $M = (S, \mathcal{A}, T, H, D, R)$, and suppose we have some approximation $\hat{T}$ of the transition probabilities. Thus, $\hat{M} = (S, \mathcal{A}, \hat{T}, H, D, R)$ is our approximation to $M$. The Simulation Lemma (stated below) shows that so long as $\hat{T}$ is close to $T$ on states that are visited with high probability by a policy $\pi$, then the utility of $\pi$ in $\hat{M}$ is close to the utility of $\pi$ in $M$. (Related results are also given in Kearns & Singh, 2002; Kearns & Koller, 1999; Kakade, Kearns & Langford, 2003; Brafman & Tennenholtz, 2002.)

**Lemma 1** (Simulation Lemma)**.** *Let any $\epsilon, \eta \geq 0$ be given. Let an MDP $M = (S, \mathcal{A}, T, H, D, R)$ be given. Let $\hat{M} = (S, \mathcal{A}, \hat{T}, H, D, R)$ be another MDP which only differs from $M$ in its transition probabilities. Let $\pi$ be a policy over the state-action sets $S, \mathcal{A}$, so that $\pi$ can be applied to both $M$ and $\hat{M}$. Assume there exists a set of state-action pairs $\overline{SA}_\eta \subseteq S \times \mathcal{A}$ such that the following holds*

$$\begin{array}{ll} \text{(i)} & \forall (s,a) \in \overline{SA}_\eta, \;\; d_{\mathrm{var}}(P(\cdot|s,a), \hat{P}(\cdot|s,a)) \leq \epsilon, \\ \text{(ii)} & P(\{(s_t, a_t)\}_{t=0}^{H} \subseteq \overline{SA}_\eta | \pi, M) \geq 1 - \eta. \end{array}$$

*(Above, $d_{\mathrm{var}}$ denotes variational distance.[4]) Then we have*

$$|U_M(\pi) - U_{\hat{M}}(\pi)| \leq H^2 \epsilon R_{\max} + \eta H R_{\max}.$$

Consider the special case where every state-action pair $(s, a) \in S \times \mathcal{A}$ satisfies condition (i), in other words, $\overline{SA}_\eta = S \times \mathcal{A}$ and thus condition (ii) is satisfied for $\eta = 0$. Then the Simulation Lemma tells us that accurate transition probabilities are sufficient for accurate policy evaluation. The Simulation Lemma also shows that not necessarily all state-action pairs' transition probabilities need to be accurately modeled: it is sufficient to accurately model a subset of state-action pairs $\overline{SA}_\eta$ such that the probability of leaving this set $\overline{SA}_\eta$ under the policy $\pi$ is sufficiently small.

Let there be some event that has probability bounded away from zero. Suppose we would like to observe that event some minimum number of times in a set of IID experiments. The following lemma allows us to prove bounds on how often we need to repeat the experiment to see that event at least the desired number of times (with high probability).

**Lemma 2.** *Let any $\delta > 0$ and $a > 0$ be given. Let $\{X_i\}_{i=1}^m$ be IID Bernoulli($\phi$) random variables. Then for $\sum_{i=1}^m X_i \geq a$ to hold with probability at least $1 - \delta$, it suffices that $m \geq \frac{2}{\phi}(a + \log \frac{1}{\delta})$.*

## 3. Problem description

The problems we are concerned with in this paper are control tasks that can be described by an MDP $M = (S, \mathcal{A}, T, H, D, R)$. However the system dynamics $T$ are unknown. Everything else in the specification of the MDP is assumed to be known. We consider two specific classes of state-action spaces and transition probabilities, which we will refer to as discrete dynamics and linearly parameterized dynamics respectively.

- Discrete dynamics: The sets $S$ and $\mathcal{A}$ are finite sets. The system dynamics $T$ can be described by a set of transition probabilities $P(s'|s, a)$, which denote the probability of the next-state being $s'$ given the current state is $s$ and the current action is $a$. More specifically we have a multinomial distribution $P(\cdot|s, a)$ over the set of states $S$ for all state-action pairs $(s, a) \in S \times \mathcal{A}$.
- Linearly parameterized dynamics: The sets $S = \mathbb{R}^{n_S}$ and $\mathcal{A} = \mathbb{R}^{n_\mathcal{A}}$ are now continuous. We assume the system obeys the following dynamics:[5]

$$x_{t+1} = A\phi(x_t) + Bu_t + w_t, \tag{1}$$

where $\phi(\cdot) : \mathbb{R}^{n_S} \to \mathbb{R}^{n_S}$. Thus, the next-state is a linear function of some (possibly non-linear) features of the current state (plus noise). This generalizes the familiar LQR model from classical control (Anderson & Moore, 1989) to non-linear settings. For example, the (body-coordinates) helicopter model used in (Ng et al., 2004) was of this form, with a particular choice of non-linear $\phi$, and the unknown parameters $A$ and $B$ were estimated from data. The process noise $\{w_t\}_t$ is IID with $w_t \sim \mathcal{N}(0, \sigma^2 I_{n_S})$. Here $\sigma^2$ is a fixed, known, constant. We also assume that $\|\phi(s)\|_2 \leq 1$ for all $s$, and that the inputs $u_t$ satisfy $\|u_t\|_2 \leq 1$.[6]

## 4. Algorithm

Let $\pi_T$ be the policy of a teacher. Although it is natural to think of $\pi_T$ as a good policy for the MDP, we do not assume this to be the case. Let any $\alpha > 0$ be given. Our algorithm (with parameters $N_T$ and $k_1$) is as follows:

[4]Let $P(\cdot), Q(\cdot)$ be two probability distributions over a set $\mathcal{X}$, then the variational distance $d_{\mathrm{var}}(P, Q)$ is defined as follows: $d_{\mathrm{var}}(P, Q) = \frac{1}{2}\int_{x \in \mathcal{X}} |P(x) - Q(x)| dx$.

[5]We chose to adhere to the most commonly used notation for continuous systems. I.e., states are represented by $x$, inputs by $u$ and the system matrices by $A$ and $B$. We use script $\mathcal{A}$ for the set of actions and standard font $A$ for the system matrix.

[6]The generalizations to unknown $\sigma^2$, to non-diagonal noise covariances, and to non-linear features over the inputs ($B\psi(u_t)$ replacing $Bu_t$) offer no special difficulties.

1. Run $N_T$ trials under the teacher's policy $\pi_T$. Save the state-action trajectories encountered during these trials. Compute $\hat{U}_M(\pi_T)$—an estimate of the utility of the teacher's policy $\pi_T$ for the real system $M$—by averaging the sum of rewards accumulated in each of the $N_T$ trials. Initialize $i = 1$.
2. Using all state-action trajectories saved so far, estimate the system dynamics $T$ using maximum likelihood estimation for the discrete dynamics case, and regularized linear regression for the linearly parameterized dynamics case (as described below). Call the estimated dynamics $\hat{T}^{(i)}$.
3. Find a $\alpha/8$ optimal policy[7] for the MDP $\hat{M}^{(i)} = (S, \mathcal{A}, \hat{T}^{(i)}, H, D, R)$. Call this policy $\pi^{(i)}$.
4. Evaluate the utility of the policy $\pi^{(i)}$ on the real system $M$. More specifically, run the policy $\pi^{(i)}$ for $k_1$ trials on the system $M$. Let $\hat{U}_M(\pi^{(i)})$ be the average sum of rewards accumulated in the $k_1$ trials. Save the state-action trajectories encountered during these trials.
5. If $\hat{U}_M(\pi^{(i)}) \geq \hat{U}_M(\pi_T) - \alpha/2$, return $\pi^{(i)}$ and exit. Otherwise set $i = i + 1$ and go back to step 2.

In the $i^{\text{th}}$ iteration of the algorithm, a policy is found using an estimate $\hat{T}^{(i)}$ of the true system dynamics $T$. For the discrete dynamics, the estimate used in the algorithm is the maximum likelihood estimates for each of the multinomial distributions $P(\cdot|s, a)$. For the linearly parameterized dynamics, the model parameters $A, B$ are estimated via regularized linear regression. In particular the $k^{\text{th}}$ rows of $A$ and $B$ are estimated by[8] $\arg\min_{A_{k,:}, B_{k,:}} \sum_j (x_{\text{next}}^{(j)} - (A_{k,:}\phi(x_{\text{curr}}^{(j)}) + B_{k,:}u_{\text{curr}}^{(j)}))^2 + \frac{1}{\kappa^2}(\|A_{k,:}\|_2^2 + \|B_{k,:}\|_2^2)$, where $j$ indexes over all state-action-state triples $\{(x_{\text{curr}}^{(j)}, u_{\text{curr}}^{(j)}, x_{\text{next}}^{(j)})\}_j$ occurring after each other in the trajectories observed for the system.

## 5. Main theorem

The following theorem gives performance and running time guarantees for the algorithm described in Section 4.[9]

**Theorem 3.** *Let an MDP $M = (S, \mathcal{A}, T, H, D, R)$ be given, except for its transition probabilities $T$. Let the system either be a discrete dynamics system or a linearly parameterized dynamical system as defined in Section 3. Let any $\alpha > 0, \delta > 0$ be given. Let $\pi_T$ be the teacher's policy, and let $\pi$ be the policy returned by the algorithm defined above. Let $N$ denote the number of iterations of the main loop of the algorithm until the exit condition is met. Let $\mathcal{T} = (H, R_{\max}, |S|, |\mathcal{A}|)$ for the discrete case, and let $\mathcal{T} = (H, R_{\max}, n_S, n_{\mathcal{A}}, \|A\|_{\text{F}}, \|B\|_{\text{F}})$ for the linearly parameterized dynamics case. Then for*

$$U_M(\pi) \geq U_M(\pi_T) - \alpha, \quad (2)$$
$$N = O(\text{poly}(\tfrac{1}{\alpha}, \tfrac{1}{\delta}, \mathcal{T})) \quad (3)$$

*to hold with probability at least $1 - \delta$, it suffices that*

$$N_T = \Omega(\text{poly}(\tfrac{1}{\alpha}, \tfrac{1}{\delta}, \mathcal{T})), \quad (4)$$
$$k_1 = \Omega(\text{poly}(\tfrac{1}{\alpha}, \tfrac{1}{\delta}, \mathcal{T})). \quad (5)$$

Note that Eqn. (2) follows from the termination condition of our algorithm and assuming we choose $k_1$ and $N_T$ large enough such that the utilities of the policies $\{\pi^{(i)}\}_i$ and $\pi_T$ are sufficiently accurately evaluated in $M$.

The proof of this theorem is quite lengthy, and will make up most of the remainder of this paper. We now give a high-level sketch of the proof ideas. Our proof is based on showing the following two facts:

1. After we have collected sufficient data from the teacher, the estimated model is accurate for evaluating the utility of the teacher's policy in every iteration of the algorithm. (Note this does not merely require that the model has to be accurate after the $N_T$ trials under the teacher's policy, but also has to stay accurate after extra data is collected from testing the policies $\{\pi^{(i)}\}_i$.)
2. One can visit inaccurately modeled state-action pairs only a "small" number of times until all state-action pairs are accurately modeled.

We now sketch how these two facts can be proved. After we have collected sufficient data from the teacher, the state-action pairs that are visited often under the teacher's policy are modeled well. From the Simulation Lemma we know that an accurate model of the state-action pairs visited often under the teacher's policy is sufficient for accurate evaluation of the utility of the teacher's policy. This establishes (1.). Every time an inaccurate state-action pair is visited, the data collected for that state-action pair can be used to improve the model. However the model can be improved only a "small" number of times until it is accurate for all state-action pairs. This establishes (2.).

We now explain how these two facts can be used to bound the number of iterations of our algorithm. Consider the policy $\pi^{(i)}$ found in iteration $i$ of the algorithm. This policy $\pi^{(i)}$ is the optimal policy[10] for the current model. When

[7] A policy $\pi_1$ is an $\epsilon$-optimal policy for an MDP $M$ if $U_M(\pi_1) \geq \max_\pi U_M(\pi) - \epsilon$.

[8] We use matlab-like notation. $A_{k,:}$ denotes the $k^{\text{th}}$ row of $A$.

[9] The performance guarantees in the theorem are stated with respect to the teacher's demonstrated performance. However, the proof requires only that the initial dynamical model be accurate for at least one good policy. Thus, for example, it is sufficient to observe a few good teacher demonstrations along with many bad demonstrations (ones generated via a highly sub-optimal policy); or even only bad demonstrations that manage to visit good parts of the state space.

[10] For simplicity of exposition in this informal discussion, we assume $\pi^{(i)}$ is optimal, rather than near-optimal. The formal results in this paper do not use this assumption.

finding this policy $\pi^{(i)}$ in the model we could have chosen the teacher's policy. So the policy $\pi^{(i)}$ performs at least as well as the teacher's policy in the current model. Now if in the real system the utility of the policy $\pi^{(i)}$ is significantly lower than the teacher's utility (which is the case as long as the algorithm continues), then the model incorrectly predicted that $\pi^{(i)}$ was better than the teacher's policy. From (1.) we have that the model correctly evaluates the utility of the teacher's policy. Thus the model must have evaluated the policy $\pi^{(i)}$ inaccurately. Using the (contrapositive of) the Simulation Lemma, we get that the policy $\pi^{(i)}$ must be visiting (with probability bounded away from 0) state-action pairs that are not very accurately modeled. So when running the policy $\pi^{(i)}$ we can collect training data that allow us to improve the model. Now from (2.) we have that visiting inaccurately modeled state-action pairs can only happen a small number of times until the dynamics is learned, thus giving us a bound on the number of iterations of the algorithm.

The theorem will be proved for the discrete dynamics case in Section 6 and for the linearly parameterized dynamics case in Section 7.

## 6. Discrete state space systems

In this section we prove Theorem 3 for the case of discrete dynamics.

The Hoeffding inequality gives a bound on the number of samples that are sufficient to estimate the expectation of a (bounded) random variable. In our algorithm, we want to guarantee that the model is accurate (for the teacher's policy) not only when we have seen the samples from the teacher, but also any time after additional samples are collected. The following lemma, which is a direct consequence of Hoeffding's inequality (as shown in the long version), gives such a bound.

**Lemma 4.** *Let any $\epsilon > 0, \delta > 0$ be given. Let $X_i$ be IID $k$-valued multinomial random variables, with distribution denoted by $P$. Let $\hat{P}_n$ denote the $n$ sample estimate of $P$. Then for $\max_{n \geq N} d_{\mathrm{var}}(P(\cdot), \hat{P}_n(\cdot)) \leq \epsilon$ to hold with probability $1-\delta$, it suffices that $N \geq \frac{k^2}{4\epsilon^2}\log\frac{k^2}{\delta\epsilon}$.*

Lemma 4 will serve two important purposes. In the proof of Lemma 5 it is used to bound the number of trajectories needed under the teacher's policy to guarantee that frequently visited state-action pairs are accurately modeled in all models $\{\hat{M}^{(i)}\}_i$. This corresponds to establishing Fact (1.) of the proof outline in Section 5. In the proof of Lemma 6 it is used to bound the total number of times a state-action pair can be visited that is not accurately modeled. This latter fact corresponds exactly to establishing Fact (2.) of the proof outline in Section 5.[11]

[11]Fact (2.) follows completely straightforwardly from Lemma 4, so rather than stating it as a separate lemma, we will instead derive it within the proof of Lemma 6.

**Lemma 5.** *Let any $\alpha > 0, \delta > 0$ be given. Assume we use the algorithm as described in Section 4. Let $N_T$ satisfy the following condition $N_T \geq \frac{4096|S|^3|\mathcal{A}|H^5R_{\max}^3}{\alpha^3}\log\frac{32H^2R_{\max}|S|^3|\mathcal{A}|}{\delta\alpha}$. Then with probability $1-\delta$ we have that $\forall i \geq N_T \;\; |U_{\hat{M}^{(i)}}(\pi_T) - U_M(\pi_T)| \leq \alpha/8$.*

*Proof (sketch).* Let $\epsilon > 0, \eta > 0$. Let $SA_\xi \subseteq S \times A$ be the set of state-action pairs such that the probability of seeing any specific state-action pair $(s,a) \in SA_\xi$ under the policy $\pi_T$ in a single trial of duration $H$ is at least $\frac{\eta}{|S||\mathcal{A}|}$. From Lemma 4 and Lemma 2 we have that for any $(s,a) \in SA_\xi$ for

$$\forall i \geq N_T \;\; d_{\mathrm{var}}(P(\cdot|s,a), \hat{P}^{(i)}(\cdot|s,a)) \leq \epsilon \tag{6}$$

to hold with probability $1-\delta'-\delta''$, it is sufficient to have

$$N_T \geq \frac{2|S||\mathcal{A}|}{\eta}\left(\frac{|S|^2}{4\epsilon^2}\log\frac{|S|^2}{\delta'\epsilon} + \log\frac{1}{\delta''}\right). \tag{7}$$

Taking a union bound over all state-action pairs $(s,a) \in SA_\xi$ (note $|SA_\xi| \leq |S||\mathcal{A}|$) gives that for Eqn. (6) to hold for all $(s,a) \in SA_\xi$ with probability $1 - |S||\mathcal{A}|\delta' - |S||\mathcal{A}|\delta''$, it suffices that Eqn. (7) is satisfied. We also have from the definition of $SA_\xi$ that $\mathrm{P}(\{(s_t,a_t)\}_{t=0}^H \subseteq \overline{SA_\xi}|\pi_T) \geq 1-\eta$. Thus the Simulation Lemma gives us that

$$\forall i \;\; |U_{\hat{M}^{(i)}}(\pi_T) - U_M(\pi_T)| \leq H^2\epsilon R_{\max} + \eta H R_{\max}.$$

Now choose $\epsilon = \frac{1}{2}\frac{\alpha/8}{H^2R_{\max}}$, $\eta = \frac{1}{2}\frac{\alpha/8}{HR_{\max}}$ and $\delta' = \delta'' = \frac{\delta}{2|S||\mathcal{A}|}$ to get the lemma. □

Lemma 5 shows that, after having seen the teacher sufficiently often, the learned model will be accurate for evaluating the utility of the teacher's policy. Moreover, no later data collection (no matter under which policy the data is collected) can make the model inaccurate for evaluation of the utility of the teacher's policy. I.e., $U_{\hat{M}^{(i)}}(\pi_T)$ will be close to $U_M(\pi_T)$ for all $i$.

**Lemma 6.** *Let any $\alpha > 0, \delta > 0$ be given. Let*

$$N_{\mathrm{ubound}} = \frac{32HR_{\max}}{\alpha}\Big(\log\frac{4}{\delta} + \frac{16^2H^4R_{\max}^2|S|^3|\mathcal{A}|}{4\alpha^2}\log\frac{64H^2R_{\max}|S|^3|\mathcal{A}|}{\alpha\delta}\Big). \tag{8}$$

*Assume in the algorithm described in Section 4 we use*

$$k_1 \geq \frac{16^2H^2R_{\max}^2}{2\alpha^2}\log\frac{8N_{\mathrm{ubound}}}{\delta}, \tag{9}$$

$$N_T \geq \frac{4096|S|^3|\mathcal{A}|H^5R_{\max}^3}{\alpha^3}\log\frac{256H^2R_{\max}|S|^3|\mathcal{A}|}{\delta\alpha}. \tag{10}$$

*Let $N$ denote the number of iterations of the algorithm until it terminates. Then we have that with probability $1-\delta$ the following hold*

(i) $$N \leq N_{ubound}, \tag{11}$$

(ii) $$\forall i = 1:N \;\; |U_{\hat{M}^{(i)}}(\pi_T) - U_M(\pi_T)| \leq \frac{\alpha}{8}, \tag{12}$$

(iii) $$\forall i = 1:N \;\; |\hat{U}_M(\pi^{(i)}) - U_M(\pi^{(i)})| \leq \frac{\alpha}{16}, \tag{13}$$

(iv) $$|\hat{U}_M(\pi_T) - U_M(\pi_T)| \leq \frac{\alpha}{16}. \tag{14}$$

*Proof (sketch).* From Lemma 5 and from the Hoeffding inequality we have that for Eqn. (12), (13) and (14) to hold (for all $i \leq N_{\text{ubound}}$) with probability $1 - \frac{\delta}{2}$, it suffices that Eqn. (10) and Eqn. (9) are satisfied.

Now since the algorithm only exits in iteration $N$, we must have for all $i = 1 : N-1$ that

$$\hat{U}_M(\pi^{(i)}) < \hat{U}_M(\pi_T) - \alpha/2. \quad (15)$$

Combining Eqn. (15), (12), (13) and (14) and the fact that $\pi^{(i)}$ is $\alpha/8$-optimal for $\hat{M}^{(i)}$ we get

$$\forall i = 1 : N-1 \;\; U_{\hat{M}^{(i)}}(\pi^{(i)}) \geq U_M(\pi^{(i)}) + \alpha/8. \quad (16)$$

In words: when the algorithm continues (in iterations $i = 1 : N-1$), the model overestimated the utility of $\pi^{(i)}$. Using the contrapositive of the Simulation Lemma with $\epsilon = \frac{1}{2}\frac{\alpha/8}{H^2 R_{\max}}$ we get that for all $i = 1 : N-1$ the policy $\pi^{(i)}$ must be visiting a state-action pair $(s, a)$ that satisfies

$$d_{\text{var}}(P(\cdot|s,a), \hat{P}^{(i)}(\cdot|s,a)) > \frac{\alpha}{16H^2R_{\max}} \quad (17)$$

with probability at least $\frac{\alpha}{16HR_{\max}}$. From Lemma 2 and Lemma 4 we get that if the algorithm had run for a number of iterations $N_{\text{ubound}}$ then with probability $1 - \frac{\delta}{2}$ all state-actions pairs would satisfy

$$d_{\text{var}}(P(\cdot|s,a), \tilde{P}^{(N)}(\cdot|s,a)) \leq \frac{\alpha}{16H^2R_{\max}}. \quad (18)$$

On the other hand we showed above that if the algorithm does not exit in iteration $i$, there must be a state-action pair satisfying Eqn. (17), which contradicts Eqn. (18). Thus $N_{\text{ubound}}$ gives an upper bound on the number of iterations of the algorithm. □

The proof of Theorem 3 for the case of discrete dynamics is a straightforward consequence of Lemma 6.

*Proof of Theorem 3 for discrete dynamics.* First note that the conditions on $N_T$ and $k_1$ of Lemma 6 are satisfied in Theorem 3. So Lemma 6 proves the bound on the number of iterations as stated in Eqn. (3). Now it only remains to prove that at termination, Eqn. (2) holds. We have from the termination condition that $\hat{U}(\pi) \geq \hat{U}(\pi_T) - \alpha/2$. Now using Eqn. (13) and Eqn. (14) we get $U_\pi \geq U_{\pi_T} - \frac{5}{8}\alpha$, which implies Eqn. (2). □

## 7. Linearly parameterized dynamical systems

In this section we prove Theorem 3 for the case of linearly parameterized dynamics described in Eqn. (1). As pointed out in Section 5, the performance guarantee of Eqn. (2) follows from the termination condition of our algorithm and assuming we choose $k_1$ and $N_T$ large enough such that the utility of the policies $\{\pi^{(i)}\}_i$ and $\pi_T$ are sufficiently accurately evaluated in $M$. This leaves us to prove the bound on the number of iterations of the algorithm. As explained more extensively in Section 5, there are two main parts to this proof. In Section 7.2 we establish the first part: the estimated model is accurate for evaluating the utility of the teacher's policy in every iteration of the algorithm. In Section 7.3 we establish the second part: one can visit inaccurately modeled states only a "small" number of times (since every such visit improves the model). In Section 7.4 we combine these two results to prove Theorem 3 for the case of linearly parameterized dynamical systems.

### 7.1. Preliminaries

The following proposition will allow us to relate accuracy of the expected value of the next-state to variational distance for the next-state distribution. This will be important for using the Simulation Lemma, which is stated in terms of variational distance.

**Proposition 7.** *We have*

$$d_{\text{var}}(\mathcal{N}(\mu_1, \sigma^2 I_n), \mathcal{N}(\mu_2, \sigma^2 I_n)) \leq \frac{1}{\sqrt{2\pi}\sigma}\|\mu_1 - \mu_2\|_2.$$

### 7.2. Accuracy of the model for the teacher's policy

Given a set of state-action trajectories, the system matrices $A, B$ are estimated by solving $n_S$ separate regularized linear regression problems, one corresponding to each row of $A$ and $B$. After appropriately relabeling variables and data, each of these regularized linear regression problems is of the form

$$\min_\theta \sum_i (y^{(i)} - \theta^\top z^{(i)})^2 + \frac{\|\theta\|_2^2}{\kappa^2}. \quad (19)$$

Here $\theta \in \mathbb{R}^{n_S + n_\mathcal{A}}$ corresponds to a row in $A$ and $B$, and the norm bounds on $u$ and $\phi(x)$ result in $\|z\|_2 \leq \sqrt{2}$. The relabeled data points are kept in the same order as they were collected. The training data collected from the teacher's demonstration is indexed from 1 to $m = N_T H$. The additional training data collected when testing the policies $\{\pi^{(j)}\}_{j=1}^N$ is indexed from $m+1$ to $\tilde{m} = N_T H + k_1 N H$. The data is generated according to a true model $M$ as described in Section 4. In the notation of Eqn. (19), this means there is some $\theta^*$ such that

$$\forall i \;\; y^{(i)} = \theta^{*\top} z^{(i)} + w^{(i)}, \quad (20)$$

where the $\{w^{(i)}\}_i$ are IID, with $w^{(i)} \sim \mathcal{N}(0, \sigma^2)$. The data generation process that we just described will be referred to as "data generated according to Eqn. (20)" from here on. Note that the training data $\{z^{(i)}\}_i$ in this setup are *non-IID*. The teacher's policy $\pi_T$ induces a distribution over states $x_t$ and inputs $u_t$ at all times $t$. However these distributions need not be the same for different times $t$, making the data non-IID. Moreover, the training data indexed from $m+1$ to $\tilde{m}$ is obtained from various policies and the resulting data generation process is very difficult to model. As a consequence, our analysis will consider the worst-case scenario where an adversary can choose the additional training data indexed from $m+1$ to $\tilde{m}$.

For $1 \leq k \leq N_T H + k_1 NH$ let the following equations define $\hat{\theta}^{(k)}$ and $\mathrm{loss}^{(k)}(\theta)$:

$$\begin{aligned} \mathrm{loss}^{(k)}(\theta) &= \textstyle\sum_{i=1}^{k} (y^{(i)} - \theta^\top z^{(i)})^2 + \frac{1}{\kappa^2}\|\theta\|_2^2, \\ \hat{\theta}^{(k)} &= \arg\min_\theta \ \mathrm{loss}^{(k)}(\theta). \end{aligned} \tag{21}$$

The following lemma establishes that a "small" number of samples from the teacher's policy $\pi_T$ is sufficient to guarantee an accurate model $\hat{\theta}^{(k)}$ for all time steps $k = N_T H$ to $N_T H + k_1 NH$.

**Lemma 8.** *Let any $\delta > 0, \epsilon > 0, \eta > 0$ be given. Consider data $\{y^{(i)}, z^{(i)}\}_{i=1}^{N_T H + k_1 NH}$ generated as described in Eqn. (20). Let $\{\hat{\theta}^{(k)}\}_k$ be defined as in Eqn. (21). Let $\{\tilde{y}^{(t)}, \tilde{z}^{(t)}\}_{t=1}^{H}$ be data generated from one trial under $\pi_T$ (and appropriately relabeled as described in paragraph above). Then for*

$$P(\max_{t \in 1:H} |\theta^\top \tilde{z}^{(t)} - \theta^{*\top} \tilde{z}^{(t)}| > \epsilon) \leq \eta \tag{22}$$

*to hold with probability $1-\delta$ for all $\theta \in \{\hat{\theta}^{(k)}\}_{k=N_T H}^{N_T H + k_1 NH}$, it suffices that*

$$N_T = \Omega\left(\mathrm{poly}(\tfrac{1}{\epsilon}, \tfrac{1}{\eta}, \tfrac{1}{\delta}, H, \|\theta^*\|_2, n_S, n_{\mathcal{A}}, k_1, N)\right).$$

If $\theta$ satisfies Eqn. (22) then it is accurate for data generated under the teacher's policy and we refer to it as accurate in the discussion below; otherwise it is referred to as inaccurate. We now sketch the key ideas in the proof of Lemma 8. A full proof is provided in Abbeel and Ng (2005). The proof proceeds in four steps.

**Step 1.** For any inaccurate parameter $\theta$ we establish that with high probability the following holds

$$\mathrm{loss}^{(N_T H)}(\theta) > \mathrm{loss}^{(N_T H)}(\theta^*) + \Omega(N_T). \tag{23}$$

I.e., the true parameter $\theta^*$ outperforms an inaccurate parameter $\theta$ by a margin of $\Omega(N_T)$ after seeing $N_T$ trajectories from the teacher. The key idea is that the expected value of the loss difference $\mathrm{loss}^{(N_T H)}(\theta) - \mathrm{loss}^{(N_T H)}(\theta^*)$ is of order $N_T$ for inaccurate $\theta$. Our proof establishes the concentration result for this non-IID setting by looking at a martingale over the differences in loss at every step and uses Azuma's inequality to prove the sum of these differences is close to its expectated value with high probability.

**Step 2.** Let $\mathrm{loss}_{\mathrm{adv}}^{(k)}(\theta) = \sum_{i=N_T H+1}^{k} (y^{(i)} - \theta^\top z^{(i)})^2$ be the additional loss incurred over the additional data points $\{z^{(i)}\}_{i=N_T H+1}^{k}$. We establish that for any $-a < 0$,

$$P(\exists k > N_T H : \mathrm{loss}_{\mathrm{adv}}^{(k)}(\theta) < \mathrm{loss}_{\mathrm{adv}}^{(k)}(\theta^*) - a) \leq \exp(-\frac{a}{\sigma^2}).$$

In words, the probability of $\theta$ ever outperforming $\theta^*$ by a margin $a$ on the additional data is exponentially small in $a$. The proof considers the random walk $\{Z_k\}_k$

$$Z_k = \mathrm{loss}_{\mathrm{adv}}^{(k)}(\theta) - \mathrm{loss}_{\mathrm{adv}}^{(k)}(\theta^*).$$

Crudely speaking we exploit the fact that no matter how an adversary chooses each additional data point $z^{(i)}$ as a function of the history up to time $i-1$, the random walk $\{Z_k\}_k$ has a positive bias. More precisely, we use the Optional Stopping Theorem on the martingale $Y_k = \exp(\frac{-1}{2\sigma^2} Z_k)$.[12]

**Step 3.** Let $\theta$ be an inaccurate parameter. From Step 1 we have that the optimal $\theta^*$ outperforms $\theta$ by a margin $\Omega(N_T)$ after having seen the initial data points $\{z^{(i)}, y^{(i)}\}_{i=1}^{N_T H}$. Step 2 says that the probability for $\theta$ to ever make up for this margin $\Omega(N_T)$ is exponentially small in $N_T$. Our proof combines these two results to show that a "small" number of samples $N_T$ from the teacher is sufficient to guarantee (with high probability) that $\theta^*$ has a smaller loss than $\theta$ in every iteration, and thus $\theta \notin \{\hat{\theta}^{(k)}\}_{k=N_T H}^{N_T H + k_1 NH}$.

**Step 4.** Our proof uses a covering argument to extend the result that $\theta \notin \{\hat{\theta}^{(k)}\}_{k=N_T H}^{N_T H + k_1 NH}$ for one specific inaccurate $\theta$ from Step 3 to hold for all inaccurate parameters $\theta$ simultaneously. As a consequence, the estimated parameters $\hat{\theta}^{(k)}$ throughout all iterations $k$ ($N_T H \leq k \leq N_T H + k_1 NH$) must be accurate. Which establishes Lemma 8.

**Theorem 9.** *Let any $\delta > 0, \alpha > 0$ be given. Let $\{\hat{M}^{(i)}\}_{i=1}^{N}$ be the models estimated throughout $N$ iterations of the algorithm for the linearly parameterized dynamics case, as described in Section 4. Then for $|U_{\hat{M}^{(i)}}(\pi_T) - U_M(\pi_T)| \leq \alpha$ to hold for all $i \in 1 : N$ with probability $1-\delta$, it suffices that $N_T = \Omega\left(\mathrm{poly}(\frac{1}{\alpha}, \frac{1}{\delta}, H, R_{\max}, \|A\|_{\mathrm{F}}, \|B\|_{\mathrm{F}}, n_S, n_{\mathcal{A}}, k_1, N)\right)$.*

*Proof (idea).* From Prop. 7 and Lemma 8 we conclude that the estimated models $\{\hat{M}^{(i)}\}_{i=1}^{N}$ are close to the true model in variational distance with high probability for states visited under the teacher's policy. Using the Simulation Lemma gives the resulting accuracy of utility evaluation. □

Theorem 9 shows that a "small" number of samples from the teacher's policy $\pi_T$ is sufficient to guarantee accurate models $\hat{M}_i^{(i)}$ throughout all iterations of the algorithm. An accurate model here means that the utility of the teacher's policy $\pi_T$ is accurately evaluated in that model, i.e., $U_{\hat{M}^{(i)}}(\pi_T)$ is close to $U_M(\pi_T)$.

### 7.3. Bound on the number of inaccurate states visits

Based on the online learning results for regularized linear regression in Kakade and Ng (2005), we can show the following result.

---

[12] **Definition**(Martingale.) Let $(\Omega, \mathcal{F}, P)$ be a probability space with a a filtration $\mathcal{F}_0, \mathcal{F}_1, \cdots$. Suppose that $X_0, X_1, \cdots$ are random variables such that for all $i \geq 0$, $X_i$ is $\mathcal{F}_i$-measurable. The sequence $X_0, X_1, \cdots$ is a martingale provided, for all $i \geq 0$, we have that $\mathrm{E}[X_{i+1}|\mathcal{F}_i] = X_i$. Due to space constraints we can not expand on these concepts here. We refer the reader to, e.g., (Durrett, 1995; Billingsley, 1995; Williams, 1991), for more details on martingales and stopping times.

**Lemma 10.** *Let any $\mu > 0, \delta > 0$ be given. For the algorithm described in Section 4 we have with probability $1-\delta$ that the number of times a state-action pair $(x,u)$ is encountered such that $\|(A\phi(x)+Bu) - (\hat{A}^{(i)}\phi(x)+\hat{B}^{(i)}u)\|_2 > \mu$ is bounded by $N_\mu = O(k_1\sqrt{k_1 N}(\log k_1 N)^3 \text{poly}(\|A\|_F, \|B\|_F, n_S, n_{\mathcal{A}}, \log\frac{1}{\delta}, H, \frac{1}{\mu}))$.*

Due to space constraints, we refer the reader to the long version for the proof (Abbeel & Ng, 2005). Lemma 10 is key to proving the bound on the number of iterations in the algorithm.

### 7.4. Proof of Theorem 3 for linearly parameterized dynamical systems

*Proof (rough sketch).* The conditions in Eqn. (4), (5) ensure that $\hat{U}_M(\pi_T), \{\hat{U}_M(\pi^{(i)}\}_i$ are accurately evaluated with high probability (by the Hoeffding inequality) and Eqn. (4) also ensures that $\{U_{\hat{M}^{(i)}}(\pi_T)\}_i$ are accurate estimates of $U_M(\pi_T)$ (by Theorem 9). Using the Simulation Lemma and the same reasoning as in the proof of Lemma 6 gives us that if the algorithm does not terminate in step 4 of the algorithm, then the policy $\pi^{(i)}$ must be visiting a state-action pair $(x,u)$ that satisfies

$$d_{\text{var}}(P(\cdot|x,u), \hat{P}^{(i)}(\cdot|x,u)) > \frac{\alpha}{16H^2R_{\max}} \tag{24}$$

with probability at least $\frac{\alpha}{16HR_{\max}}$. If $(x,u)$ satisfies Eqn. (24) then we must have (using Prop. 7) that

$$\|(A\phi(x)+Bu) - (\hat{A}^{(i)}\phi(x)+\hat{B}^{(i)}u)\|_2 > \frac{\sqrt{2\pi}\sigma\alpha}{16H^2R_{\max}}.$$

From Lemma 10 this can happen only

$$O(k_1\sqrt{k_1N}(\log k_1N)^3\text{poly}(\|A\|_F, \|B\|_F, n_S, n_{\mathcal{A}}, \log\frac{1}{\delta}, H, R_{\max}, \frac{1}{\alpha})) \tag{25}$$

times in $N$ iterations of the algorithm. On the other hand, if the algorithm continues, we have from above that such an error must be encountered (with high probability)

$$\Omega(\frac{\alpha}{HR_{\max}}N) \tag{26}$$

times. Note that the lower bound on the number of state-action pairs encountered with large error in Eqn. (26) grows faster in $N$ than the upper bound in Eqn. (25).[13] Once the lower bound is larger than the upper bound we have a contradiction. Thus from Eqn. (26) and (25) we can conclude that after a number of iterations as given by Eqn. (3) the algorithm must have terminated with high probability. Also, since we chose $k_1, N_T$ such that $\{\hat{U}_M(\pi^{(i)}\}_i$ and $\hat{U}_M(\pi_T)$ are accurately evaluated, Eqn. (2) holds when the algorithm terminates. □

[13]In this proof sketch we ignore a dependence of $k_1$ on $N$. See the long version for a formal proof.

## References


Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. *Proc. ICML*.

Abbeel, P., & Ng, A. Y. (2005). Exploration and apprenticeship learning in reinforcement learning. (Full paper.) `http://www.cs.stanford.edu/~pabbeel`.

Amit, R., & Mataric, M. (2002). Learning movement sequences from demonstration. *Proc. ICDL*.

Anderson, B., & Moore, J. (1989). *Optimal control: Linear quadratic methods*. Prentice-Hall.

Bertsekas, D. P., & Tsitsiklis, J. (1996). *Neuro-dynamic programming*. Athena Scientific.

Billingsley, P. (1995). *Probability and Measure*. Wiley Interscience.

Brafman, R. I., & Tennenholtz, M. (2002). R-max, a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*.

Demiris, J., & Hayes, G. (1994). A robot controller using learning by imitation.

Durrett, R. (1995). *Probability:Theory and Examples*. Duxbury Press.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *JAIR*.

Kakade, S., Kearns, M., & Langford, J. (2003). Exploration in metric state spaces. *Proc. ICML*.

Kakade, S., & Ng, A. Y. (2005). Online bounds for Bayesian algorithms. *NIPS 17*.

Kearns, M., & Koller, D. (1999). Efficient reinforcement learning in factored MDPs. *Proc. IJCAI*.

Kearns, M., & Singh, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning journal*.

Kuniyoshi, Y., Inaba, M., & Inoue, H. (1994). Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *T-RA*, *10*, 799–822.

Ng, A. Y., Coates, A., Diel, M., Ganapathi, V., Schulte, J., Tse, B., Berger, E., & Liang, E. (2004). Inverted autonomous helicopter flight via reinforcement learning. *International Symposium on Experimental Robotics*.

Pomerleau, D. (1989). Alvinn: An autonomous land vehicle in a neural network. *NIPS 1*.

Sammut, C., Hurst, S., Kedzier, D., & Michie, D. (1992). Learning to fly. *Proc. ICML*.

Schaal, S., & Atkeson, C. G. (1994). Robot learning by nonparametric regression. *Proc. IROS*.

Smart, W. D., & Kaelbling, L. P. (2000). Practical reinforcement learning in continuous spaces. *Proc. ICML*.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. MIT Press.

Williams, D. (1991). *Probability with Martingales*. Cambridge Mathematical Textbooks.