Breaking the Deadly Triad in Reinforcement Learning

Shangtong Zhang
St Catherine's College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity 2022

To my parents Wei and Yuehua

Acknowledgements

I feel indebted to my advisor Shimon Whiteson for his support and guidance during my DPhil study, without which this thesis would not have been possible. As a scientist, Shimon is visionary and always aims high. As an educator, Shimon is supportive and willing to start low. I cannot exaggerate enough what I have learned from Shimon, both research and beyond. I thank Emma Brunskill and Alessandro Abate for carefully examining this thesis. I thank the colleagues in WhiRL: Mingfei Sun, Matthew Fellows, Wendelin Boehmer, Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Maximilian Igl, Supratik Paul, Luisa Zintgraf, Anuj Mahajan, Jelena Luketina, Vitaly Kurin, Kristian Hartikainen, Matt Smith, Tarun Gupta, Zheng Xiong, Benjamin Ellis, Jacob Beck, Risto Vuorio, and Charline Le Lan. My study at Oxford would not have been so enjoyable without them. I thank Remi Tachet des Combes, Romain Laroche, and Harm van Seijen. My two internships at Microsoft would not have been so successful without their support. I thank Michael Mathieu, Sherjil Ozair, Srivatsan Srinivasan, Caglar Gulcehre, Ray Jiang, Tom Le Paine, Konrad Zolna, Richard Powell, David Choi, Wojciech M. Czarnecki, Nando de Freitas, and Oriol Vinyals for introducing me to AlphaStar during my stay at DeepMind. I thank Adam White and Hado van Hasselt. The collaboration with them during my internship at DeepMind is truly unforgettable. I also thank many other collaborators: Yi Wan, Bo Liu, Hengshuai Yao, Veronica Chelu, Richard Sutton, and Vivek Veeriah. Many chapters in this thesis would not have been possible without them.

No word can express my thanks enough to my wife Yijia Yu. It is her love and support that helped me go through all the hardness.

Abstract

Reinforcement Learning (RL) is a promising framework for solving sequential decision making problems emerging from agent-environment interactions via trial and error. *Off-policy learning* is one of the most important techniques in RL, which enables an RL agent to learn from agent-environment interactions generated by a policy (i.e, a decision making rule that an agent relies on to interact with the environment) that is different from the policy of interest. Arguably, this flexibility is key to applying RL to real-world problems. Off-policy learning, however, often leads to instability of RL algorithms, if combined with *function approximation* (i.e., using a parameterized function to represent quantities of interest) and *bootstrapping* (i.e., recursively constructing a learning target for an estimator by using the estimator itself), two arguably indispensable ingredients for large-scale RL applications. This instability, resulting from the combination of off-policy learning, function approximation, and bootstrapping, is the notorious *deadly triad* in RL.

In this thesis, we propose several novel RL algorithms theoretically addressing the deadly triad. The proposed algorithms cover a wide range of RL settings (e.g., both prediction and control, both value-based and policy-based methods, both discounted and average-reward performance metrics). By contrast, existing methods address this issue in only a few RL settings, where our methods also exhibit several advantages over existing ones, e.g., reduced variance, improved asymptotic performance guarantee. These improvements are made possible by the use of several advanced tools (e.g., target networks, differential value functions, density ratios, and truncated followon traces). Importantly, the proposed algorithms remain fully incremental and computationally efficient, making them readily available for large-scale RL applications.

Besides the theoretical contributions in breaking the deadly triad, we also make empirical contributions by introducing a bi-directional target net-

work that scales up residual algorithms, a family of RL algorithms that break the deadly triad in some restricted settings.

Preface

This thesis is based on several papers I authored. In particular, Chapters 3, 7, and 9 are based on Zhang et al. (2021d). Chapters 4 is based on Zhang et al. (2020d,e) and my contribution to Jiang et al. (2022a). Chapters 5 and 11 are based on Zhang and Whiteson (2022). Chapter 6 is based on Zhang et al. (2020c). Chapter 8 is based on my contribution to Zhang et al. (2021c). Chapter 10 is based on Zhang et al. (2020b). Chapter 12 is based on Zhang et al. (2019, 2020d). All contents of this thesis are based on my writing.

1. Zhang, S., Boehmer, W., and Whiteson, S. (2019). Generalized off-policy actor-critic. In *Advances in Neural Information Processing Systems*.
2. Zhang, S., Boehmer, W., and Whiteson, S. (2020b). Deep residual reinforcement learning. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems* (Best Paper Award).
3. Zhang, S., Liu, B., and Whiteson, S. (2020c). GradientDICE: Rethinking generalized offline estimation of stationary values. In *Proceedings of the International Conference on Machine Learning*.
4. Zhang, S., Liu, B., Yao, H., and Whiteson, S. (2020d). Provably convergent two-timescale off-policy actor-critic with function approximation. In *Proceedings of the International Conference on Machine Learning*.
5. Zhang, S., Veeriah, V., and Whiteson, S. (2020e). Learning retrospective knowledge with reverse reinforcement learning. In *Advances in Neural Information Processing Systems*.
6. Zhang, S., Wan, Y., Sutton, R. S., and Whiteson, S. (2021c). Average-reward off-policy policy evaluation with function approximation. In *Proceedings of the International Conference on Machine Learning*.

7. Zhang, S., Yao, H., and Whiteson, S. (2021d). Breaking the deadly triad with a target network. In *Proceedings of the International Conference on Machine Learning*.
8. Zhang, S. and Whiteson, S. (2022). Truncated emphatic temporal difference methods for prediction and control. *Journal of Machine Learning Research*
9. Jiang, R., Zhang, S., Chelu, V., White, A., and van Hasselt, H. (2022b). Learning expected emphatic traces for deep RL. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

During the development of this thesis, I also authored the following papers, which are not directly related to the topic of this thesis and thus are not included.

10. Zhang, S. and Whiteson, S. (2019). DAC: the double actor-critic architecture for learning options. In *Advances in Neural Information Processing Systems*.
11. Zhang, S., Liu, B., and Whiteson, S. (2021b). Mean-variance policy iteration for risk-averse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
12. Mathieu, M., Ozair, S., Srinivasan, S., Gulcehre, C., Zhang, S., Jiang, R., Paine, T. L., Zolna, K., Powell, R., Schrittwieser, J., Choi, D., Georgiev, P., Toyama, D. K., Huang, A., Ring, R., Babuschkin, I., Ewalds, T., Bordbar, M., Henderson, S., Colmenarejo, S. G., van den Oord, A., Czarnecki, W. M., de Freitas, N., and Vinyals, O. (2021). Starcraft II unplugged: Large scale offline reinforcement learning. In *Deep RL Workshop NeurIPS 2021*.
13. Zhang, S., des Combes, R. T., and Laroche, R. (2021a). Global optimality and finite sample analysis of softmax off-policy actor critic under state distribution mismatch. *arXiv preprint arXiv:2111.02997*.
14. Zhang, S., Tachet, R., and Laroche, R. (2022b). On the chattering of sarsa with linear function approximation. *arXiv preprint arXiv:2202.06828*.
15. Zhang, S., Laroche, R., van Seijen, H., Whiteson, S., and des Combes, R. T. (2022a). A deeper look at discounting mismatch in actor-critic algorithms. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*.

Contents

List of Figures

Chapter 1

Introduction

Arguably, an agent with artificial general intelligence should be able to sequentially make decisions towards certain goals in an efficient, scalable, and stable way. Reinforcement Learning (RL, Sutton and Barto (2018)) is a framework for studying sequential decision making problems via trial and error and has made tremendous success in various domains (see, e.g., Mnih et al. (2015); Silver et al. (2016); Jumper et al. (2021); Vinyals et al. (2019); Bellemare et al. (2020); Mirhoseini et al. (2021); Bastani et al. (2021); Degrave et al. (2022)). Different from many other frameworks that can be used for sequential decision making (e.g., dynamic programming (Bellman, 1966), optimal control (Kirk, 2004), and search (Russell and Norvig, 2002)), RL does not assume to know the internal dynamics of the environment, which makes RL readily applicable to a wide range of real-world problems, e.g., vehicle parking.

RL considers an agent-environment interface. At each time step, an agent makes a decision (i.e., an action) according to a policy, which is a mapping from the agent's current state (i.e., the set of the information that the agent currently has) to a probability distribution of available actions. The environment receives this action, emits a reward, and leads the agent to a successor state. Here the reward is a scalar signal. It is assumed, in the RL framework, that this reward signal is predesigned and faithfully conveys the goal we want the agent to achieve (Sutton, 2004). Consequently, all the agent needs to do is to adapt its policy to maximize the received reward signals in a long run. In the vehicle parking example, the state could be sensor data such as images from the backup camera and the velocity of the vehicle. The action could be to steer, to brake, or to accelerate. The reward could be a negative constant every second until the vehicle is successfully parked. It is worth mentioning that designing a reward signal to faithfully convey human intention is nontrivial, especially in real-world scenarios. For instance, in the vehicle parking example, one might also want to reflect safety constraints in the reward signal. And additional techniques

such as supervised learning might be necessary to determine whether the vehicle is properly parked. Reward design itself is an active research area (see, e.g., Abbeel and Ng (2004); Abel et al. (2021)), which, however, deviates from the main topic of this thesis. In this thesis, we assume that the desired reward signal is available and focus on the maximization of this reward signal. The RL community has developed numerous exciting ideas for this reward-maximization process. Arguably, bootstrapping, function approximation, and off-policy learning are among the most important ones.

The idea of *bootstrapping* (Sutton, 1988) is to recursively construct a learning target for an estimator by using the estimator itself. In the RL context, *bootstrapping* is an efficient method to address the credit assignment problem, i.e., to determine which actions should be credited for delayed rewards in the future. Temporal Difference learning (TD, Sutton (1988)) is an example of bootstrapping methods. In TD, the estimate of the *value* of a state, i.e., the estimate of the expected aggregated future rewards after visiting a state, is updated towards the summation of the immediate reward and the estimate of the value of the successor state. Since the successor state is immediately available in the next time step, TD is able to update the estimate every time step in an online manner. By contrast, without bootstrapping, it would not be possible to update the estimate of the value of a state until all future rewards are received, which could possibly take a long time. In the vehicle parking example, we might want to know the rewards an agent can receive starting at some position following some policy. Without bootstrapping, one straightforward way is to wait until the end of this trial of parking. With bootstrapping, we can, however, get an estimate immediately at the next second via summing up the immediate reward that the agent have received and our estimate of the rewards that the agent could receive in the future starting from the new position.

The idea of *function approximation* is to use a parameterized function to represent quantities of interest, e.g., the values of different states, in a compact way. A straightforward method to store the values of different states is to use a look-up table to store them separately. This tabular method is, however, memory-expensive when there are too many states. Even worse, this tabular method cannot easily provide generalization, i.e., it is hard to deduce the value of a new state based on the look-up table storing the value of known states. By contrast, function approximation adopts a feature function that maps each state to several numerical features. Those numerical features serve as the inputs to a parameterized function. And we adapt the parameters such that the parameterized function is able to approximate well the

quantities of interest, e.g., the values of states. As features are usually much less than states and parameters are shared across all states, function approximation exhibits great memory efficiency. More importantly, as features are usually correlated across states, function approximation easily provides generalization to unseen states. Consider, again, estimating the rewards a parking agent can receive starting from different positions in the vehicle parking example. If the states are the images from the backup camera, maintaining a look-up table using the image as key and the estimated reward as value is impractical because there are too many possible images. A more practical way might be to train a neural network (LeCun et al., 2015) to approximate the mapping from the images to the estimated rewards.

The idea of *off-policy* learning is to decouple the policy being learned (target policy) from the policy used for interaction with the environment (behavior policy) and thus allows them to be different from each other. By contrast, another learning paradigm, called *on-policy* learning, restricts the policy being learned to be the one that the agent uses to interact with the environment. This flexibility of off-policy learning helps improve the sample complexity of RL. For example, with off-policy learning, one can reuse existing experiences generated by other policies to learn the policy of interest, without requiring additional interaction with the environment. One can also learn multiple policies of interest simultaneously while interacting with the environment following a single behavior policy (cf. Sutton et al. (2011)). This flexibility is also key to solving the exploration and exploitation dilemma. On the one hand, we want the agent to interact with the environment in an exploratory way such that more information can be obtained. On the other hand, we also want the agent to interact with the environment in an exploitative way such that it can maximize the received rewards. With on-policy learning, it is nontrivial to require the policy to be both exploratory and exploitative. With off-policy learning, one can, however, use an exploratory behavior policy for interacting with the environment while learning an exploitative target policy. Consider again the vehicle parking example. We might have some premature parking policies and want to know its outcome. Off-policy learning allows us to evaluate those premature parking policies without actually deploying them, possibly with data from a single human driver. This greatly reduces the risk in the evaluation process.

It thus would be desirable if an RL algorithm can simultaneously combine all the three ingredients above. Such an algorithm, unfortunately, is usually not theoretically guaranteed to be stable, i.e., the output of such an algorithm can easily go to infinity. This instability is known as the notorious *deadly triad* (Chapter 11.3 of Sutton and

	discounted performance metric	average reward performance metric
value prediction	Sutton et al. (2008, 2009) Sutton et al. (2016) **Ours**	**Ours**
density ratio prediction	Nachum et al. (2019a) **Ours**	**Ours**
value-based control with a fixed behavior policy	Maei et al. (2010) **Ours**	**Ours**
value-based control with a changing behavior policy	**Ours**	**Ours**
policy-based control with a fixed behavior policy	**Ours**	N/A
policy-based control with a changing behavior policy	N/A	N/A

Table 1.1: Solution methods of the deadly triad issue in different RL settings. This table is an arguably exhaustive taxonomy of RL settings regarding the deadly triad. The exact definition of each setting is deferred to Chapter 2 for the ease of presentation.

Barto (2018)). When it comes to the deadly triad, we limit our discussion to RL algorithms that are computationally efficient, i.e., algorithms that have linear per-step computational and memory complexity w.r.t. the number of input features. There are indeed algorithms with squared complexity that can address the deadly triad, which are, however, hard to use in large scale applications due to the high demand of computation resources. Though there have been works theoretically addressing this deadly triad in a computationally efficient way (Sutton et al., 2008, 2009; Maei et al., 2010; Sutton et al., 2016), they consider only a few RL settings, namely, value-based methods under a discounted performance metric. **The major contribution of this thesis is to theoretically address the deadly triad issue in a wide range of RL settings, including both policy- and value-based methods and under various performance metrics.** Table 1.1 details how the settings this thesis considers differ from the settings previous works consider. Importantly, in the settings where solutions already exist, our methods also exhibit various advantages over existing ones, e.g., reduced variance, improved asymptotic performance guarantee. Furthermore, all our proposed convergent algorithms are also computationally efficient. Their per-step computation and memory complexity are linear with respect to the number of input features, making them readily available for large-scale RL applications.

The theoretical advances presented in this thesis are made possible via the use of several advanced tools, including target networks, differential value functions, density ratios, and truncated followon traces. The target network (Mnih et al., 2015) is a technique widely used by practitioners to *empirically* stabilize RL algorithms (e.g., Lillicrap et al. (2016); Fujimoto et al. (2018); Haarnoja et al. (2018)). The idea is to keep a slowly changing copy of the estimator, which is referred to as the target network, and use the copy to construct an update target for the estimator in bootstrapping. **This thesis is the first to theoretically identify target networks as effective tools for addressing the deadly triad.** Differential value functions are key to credit assignment when it comes to the average reward performance metric. **This thesis proposes the first convergent algorithm to learn differential value functions in the context of the deadly triad.** In RL, agents following different policies have different state distributions. The density ratio is the ratio between the state distribution under the target policy and that under the behavior policy. The density ratio is an effective tool for correcting the discrepancy between the target policy and the behavior policy and thus addressing the deadly triad. **This thesis proposes the first convergent algorithm to learn the density ratio in the context of the deadly triad and the average reward performance metric. This thesis also proposes the first convergent algorithm to learn a generalization of the density ratio for addressing the deadly triad.** Followon traces are first introduced by Sutton et al. (2016) in emphatic TD methods to address the deadly triad in value prediction problems. Followon traces, however, suffer from large variance and the convergence analysis of emphatic TD methods is only asymptotic (Yu, 2015). **By introducing the truncated followon traces, this thesis is the first to address the large variance of the followon trace in a theoretically grounded way. This thesis also provides both asymptotic and nonasymptotic convergence analysis for the resulting truncated emphatic TD methods for both prediction and control settings, in the context of the deadly triad.**

Besides theoretically breaking the deadly triad, we also make empirical contributions. In particular, we design a novel bidirectional target network that scales up residual algorithms (Baird, 1995). Residual algorithms are a class of RL algorithms that are able to theoretically address the deadly triad issue in some restricted settings and have been widely studied before the emergence of deep RL (i.e., the combination of RL and deep neural networks, cf. Mnih et al. (2015)). **By using the bidirec-**

tional target network, this thesis provides the first empirical success of residual algorithms in the context of deep RL.

Chapter 2

Background

2.1 Mathematical Conventions

In this section, we discuss the mathematical conventions we follow in this thesis.

All vectors are column. A matrix M (not necessarily symmetric) is said to be positive definite (p.d.) if there exists a constant $\lambda > 0$ such that $x^\top Mx \geq \lambda x^\top x$ holds for any x. It is well known that M is p.d. if and only if $M + M^\top$ is p.d. M is negative definite (n.d.) if and only if $-M$ is p.d. We overload 0 to denote the scalar zero, an all-zero vector, and an all-zero matrix. The notation 1 is also similarly overloaded. For a vector x and a p.d. matrix M, we use $\|x\|_M \doteq \sqrt{x^\top Mx}$ to denote the vector norm induced by M. We also use $\|\cdot\|_M$ to denote the corresponding induced matrix norm, i.e., for a matrix X, $\|X\|_M \doteq \max_{x \neq 0} \frac{\|Xx\|_M}{\|x\|_M}$. We use $\|\cdot\|$ as shorthand for $\|\cdot\|_I$ where I is the identity matrix, i.e., $\|\cdot\|$ is the standard ℓ_2-norm. We use $diag(x)$ to denote a diagonal matrix whose diagonal entry is x and write $\|\cdot\|_x$ as shorthand for $\|\cdot\|_{diag(x)}$ when $diag(x)$ is p.d.. We use $\|\cdot\|_\infty$ and $\|\cdot\|_1$ to denote the standard infinity norm and ℓ_1-norm respectively. We use $\langle \cdot, \cdot \rangle$ to denote the inner product in Euclidean spaces, i.e., $\langle x, y \rangle \doteq x^\top y$. We use functions and vectors interchangeably when it does not confuse, e.g., if f is a function from $\mathcal{S}$ to $\mathbb{R}$, we also use f to denote a vector in $\mathbb{R}^{|\mathcal{S}|}$, whose s-th element is $f(s)$.

2.2 Markov Decision Process

In this section, we discuss the mathematical model we use for RL in this thesis.

We consider an infinite horizon Markov Decision Process (MDP, see, e.g., Puterman (2014)) consisting of a finite *state space* $\mathcal{S}$, a finite *action space* $\mathcal{A}$, a bounded *reward* function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ such that $\forall s \in \mathcal{S}, a \in \mathcal{A}, |r(s,a)| \leq r_{max} < \infty$, a

transition function $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$, a *policy* $\pi : \mathcal{A} \times \mathcal{S} \to [0, 1]$, and an *initial distribution* $p_0 : \mathcal{S} \to [0, 1]$.

At time step 0, an initial state S_0 is sampled according to $p_0(\cdot)$. Here $p_0(\cdot)$ represents a probability distribution on $\mathcal{S}$ whose probability mass function is p_0. At time step $t = 0, 1, \ldots$, an action A_t is sampled according to $\pi(\cdot|S_t)$. Here $\pi(\cdot|S_t)$ represents a probability distribution on $\mathcal{A}$, where the probability mass for $a \in \mathcal{A}$ is $\pi(a|S_t)$. Then a reward $R_{t+1} \doteq r(S_t, A_t)$ is emitted and a successor state S_{t+1} is sampled according to $p(\cdot|S_t, A_t)$. Here $p(\cdot|S_t, A_t)$ represents a probability distribution on $\mathcal{S}$, where the probability mass for $s \in \mathcal{S}$ is $p(s|S_t, A_t)$. We consider two different settings for studying this MDP: the *discounted* setting and the *average-reward* setting. They differ from each other in how the performance of the policy π is measured.

In the discounted setting, we consider a *discount factor* $\gamma \in [0, 1)$ to trade off the importance of long term and short term rewards. To summarize the possible future rewards starting from a state s following the policy π, we define the *state value function* v_π as

$$v_\pi(s) \doteq \mathbb{E}_{\pi,p}[G_t|S_t = s],$$

where

$$G_t \doteq \sum_{i=0}^{\infty} \gamma^i R_{t+i+1}$$

is the *return* at time step t. Similarly, we use the *action value function* q_π to summarize the possible future rewards starting from a state-action pair (s, a) following the policy π:

$$q_\pi(s, a) \doteq \mathbb{E}_{\pi,p}[G_t|S_t = s, A_t = a].$$

The two value functions are related to each other as

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a).$$

We now arrive at our first performance metric, the *discounted total rewards* J_π, which is the expectation of the summation of the discounted future rewards starting from time step 0:

$$J_\pi \doteq \mathbb{E}_{s\sim p_0(\cdot)}[v_\pi(s)].$$

In the average-reward setting, we consider another performance metric, the *average reward* $\bar{J}_\pi$ (a.k.a. *gain*, see, e.g, Puterman (2014)), which is the average of the received rewards in a long run:

$$\bar{J}_\pi \doteq \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\pi,p}[R_t].$$

The average reward exists and is independent of p_0 if the following assumption holds:

Assumption 2.1. *The Markov chain in $\mathcal{S}$ induced by the policy π is ergodic.*

Assumption 2.1 is sufficient but not necessary. Weaker conditions (e.g., the Markov chain in $\mathcal{S}$ induced by the policy π is a unichain) can also be used. In this thesis, we use Assumption 2.1 for the ease of presentation. We similarly define the *differential state value function* $\bar{v}_\pi$ (see, e.g., Sutton and Barto (2018), a.k.a. *bias*, see, e.g., Puterman (2014)) and the *differential action value function* $\bar{q}_\pi$ as

$$\bar{v}_\pi(s) \doteq \mathbb{E}_{\pi,p}\left[\sum_{i=1}^{\infty}(R_{t+i} - \bar{J}_\pi)|S_t = s\right]$$

and

$$\bar{q}_\pi(s,a) \doteq \mathbb{E}_{\pi,p}\left[\sum_{i=1}^{\infty}(R_{t+i} - \bar{J}_\pi)|S_t = s, A_t = a\right],$$

which summarize the possible future excessive rewards between the immediate reward and the average reward. Both $\bar{v}_\pi$ and $\bar{q}_\pi$ are well-defined under Assumption 2.1. The two differential value functions are related to each other as

$$\bar{v}_\pi(s) = \sum_a \pi(a|s)\bar{q}_\pi(s,a).$$

We consider both *prediction* and *control* problems in both discounted and average-reward settings. The prediction problem refers to estimating the performance of a policy, i.e., computing $J_\pi, v_\pi, q_\pi, \bar{J}_\pi, \bar{v}_\pi, \bar{q}_\pi$. The control problem refers to finding a policy maximizing J_π or $\bar{J}_\pi$.

2.3 Bellman Equations

In this section, we discuss dynamic programming, one origin of RL.

Bellman equations (Bellman, 1966; Puterman, 2014) are powerful tools to study the prediction and control problems in both discounted and average-reward settings.

For the prediction problem in the discounted setting, we consider the following Bellman equation:

$$v = r_\pi + \gamma P_\pi v, \tag{2.1}$$

where $v \in \mathbb{R}^{|\mathcal{S}|}$ is the free variable, $r_\pi \in \mathbb{R}^{|\mathcal{S}|}$ is the reward vector induced by the policy π, i.e., the s-indexed element of r_π is $r_\pi(s) \doteq \sum_a \pi(a|s) r(s,a)$, and $P_\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is the state transition matrix induced by the policy, i.e., $P_\pi(s,s') \doteq \sum_a \pi(a|s) p(s'|s,a)$. The state value function v_π is the unique solution to (2.1). Similarly, we also have the Bellman equation for action value:

$$q = r + \gamma P_\pi q, \tag{2.2}$$

where $q \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ is the free variable, $P_\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ is overloaded to denote the state-action transition matrix as well for simplifying notations, i.e., $P_\pi((s,a),(s',a')) \doteq \pi(a'|s')p(s'|s,a)$. The action value function q_π is the unique solution to (2.2). Once v_π or q_π is known, computing J_π becomes straightforward using samples from p_0.

For the control problem in the discounted setting, we have two Bellman optimality equations:

$$q(s,a) = r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a) \max_{a'\in\mathcal{A}} q(s',a') \quad \forall (s,a), \tag{2.3}$$

$$v(s) = \max_a r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a) v(s') \quad \forall s, \tag{2.4}$$

where $q \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ and $v \in \mathbb{R}^{|\mathcal{S}|}$ are the free variables. (2.3) has a unique solution, which we refer to as q_*. (2.4) also has a unique solution, which we refer to as v_* Importantly, for any (π, s, a), we have

$$\begin{aligned} q_*(s,a) &\geq q_\pi(s,a), \\ v_*(s) &\geq v_\pi(s). \end{aligned}$$

It is then easy to see that J_π is maximized if the policy π is greedy w.r.t. $q_*(s,a)$, or equivalently, $r(s,a) + \sum_{s'} p(s'|s,a) v_*(s')$.

For the prediction problem in the average reward setting, we consider the following differential Bellman equation:

$$v = r_\pi - \bar{J}1 + P_\pi v, \tag{2.5}$$

where $v \in \mathbb{R}^{|\mathcal{S}|}$ and $\bar{J} \in \mathbb{R}$ are free variables. Since (2.5) has $|\mathcal{S}|+1$ free variables but only $|\mathcal{S}|$ constraints, it does not have a unique solution. It is well known (see, e.g., Puterman (2014)) that all the solutions to (2.5) form a set

$$\left\{(v, \bar{J}) \mid v = \bar{v}_\pi + c1, c \in \mathbb{R}, \bar{J} = \bar{J}_\pi\right\},$$

i.e., the solution to $\bar{J}$ is always the average reward, but the solutions to v can differ from the differential value function $\bar{v}_\pi$ by an arbitrary constant offset. Similarly, we also have the differential Bellman equation for action value:

$$q = r - \bar{J}1 + P_\pi q, \tag{2.6}$$

where $q \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ and $\bar{J} \in \mathbb{R}$ are free variables. The solution to $\bar{J}$ is always the reward rate $\bar{J}_\pi$, but the solutions to q can differ from the differential action value function $\bar{q}_\pi$ by an arbitrary constant offset.

For the control problem in the average reward setting, we consider the following differential Bellman optimality equation:

$$q(s,a) = r(s,a) - \bar{J} + \sum_{s'} p(s'|s,a) \max_{a' \in \mathcal{A}} q(s',a'), \quad \forall (s,a), \tag{2.7}$$

where $q \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ and $\bar{J} \in \mathbb{R}$ are free variables. Similarly, the only solution to $\bar{J}$ is $\bar{J}_* \doteq \sup_\pi \bar{J}_\pi$ and all the solutions to q differ from each other by a constant offset.

Now it becomes clear that both the prediction and control problems can be addressed by solving Bellman equations (2.1), (2.2), (2.3), (2.4), (2.5), (2.6), and (2.7). One classical approach for solving Bellman equations is Dynamic Programming (DP, Bellman (1966)). For example, to solve (2.1), DP considers the Bellman operator $\mathcal{T}_\pi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ defined as

$$\mathcal{T}_\pi v \doteq r_\pi + \gamma P_\pi v.$$

It is well-known that $\mathcal{T}_\pi$ is a contraction mapping and v_π is its unique fixed point according to the Banach fixed point theorem. Consequently, repeatedly applying $\mathcal{T}_\pi$ to any initial vector $v \in \mathbb{R}^{|\mathcal{S}|}$ converges to v_π. To solve (2.2), DP considers the action value Bellman operator $\mathcal{T}_\pi : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ defined as

$$\mathcal{T}_\pi q \doteq r + \gamma P_\pi q.$$

Here we have overloaded $\mathcal{T}_\pi$ for simplifying notations. The action value Bellman operator is a contraction mapping as well, and repeatedly applying $\mathcal{T}_\pi$ to any initial

vector $q \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ converges to q_π. To solve (2.3), DP considers the action value optimal Bellman operator $\mathcal{T}_* : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\to|\mathcal{S}\times\mathcal{A}|}$ defined as

$$(\mathcal{T}_* q)(s,a) \doteq r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a) \max_{a'\in\mathcal{A}} q(s',a').$$

$\mathcal{T}_*$ is a contraction mapping as well, and repeatedly applying $\mathcal{T}_*$ to any initial vector $q \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ converges to q_*. Similarly, to solve (2.4), DP considers the state value optimal Bellman operator $\mathcal{T}_* : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ defined as

$$(\mathcal{T}_* v)(s) \doteq \max_a r(s,a) + \gamma \sum_{s'} p(s'|s,a) v(s'), \tag{2.8}$$

where we have overloaded $\mathcal{T}_*$ for simplicity. Repeatedly applying $\mathcal{T}_*$ to any initial vector $v \in \mathbb{R}^{|\mathcal{S}|}$ converges to v_*.

It is, however, worth mentioning that applying $\mathcal{T}_\pi$ and $\mathcal{T}_*$ requires to know the transition function p, which is usually unknown in practice. Reinforcement Learning (RL, Sutton and Barto (2018)) is a framework to address this challenge. Instead of requiring knowing $p(s'|s,a)$ for any s' given (s,a), RL requires only a sample from $p(\cdot|s,a)$, which is usually readily available in practice. We use Temporal Difference learning (TD, Sutton (1988)) as a representative RL algorithm to demonstrate the difference between RL and DP. To solve (2.1), at time step t, TD updates a single element of v as

$$v(S_t) \leftarrow v(S_t) + \alpha_t \left(R_{t+1} + \gamma v(S_{t+1}) - v(S_t)\right),$$

where $\{\alpha_t\}$ is a sequence of learning rates. This TD update differs from the DP update of applying $\mathcal{T}_\pi$ to v mainly in the following aspects: (1) TD only updates a single element of v at each time step (i.e., $v(S_t)$), while DP updates all the elements of v at each iteration. (2) TD updates the element $v(S_t)$ incrementally towards $R_{t+1} + \gamma v(S_{t+1})$, controlled by the learning rate α_t, while DP completely rewrites v with $\mathcal{T}_\pi v$. (3) The computation of $\mathcal{T}_\pi v$ in DP requires to know P_π, while TD requires only S_{t+1}, a sample from $p(\cdot|S_t, A_t)$. To summarize, RL provides an incremental and stochastic way for implementing DP.

2.4 Bootstrapping

In this section, we discuss the first ingredient of the deadly triad.

Bootstrapping is perhaps the most important idea in RL. Consider, for example, computing v_π. By definition, $v_\pi(s) \doteq \mathbb{E}[G_t|S_t = s]$, so the most straightforward way

is to update $v(S_t)$ towards a realization of the random variable G_t, which is known as the Monte Carlo method. Obtaining a realization of G_t is, however, nontrivial. Recall that G_t is the summation of all possible futures rewards ($\sum_{i=0}^{\infty} \gamma^i R_{t+i+1}$). So in an infinite horizon MDP, we have to wait for sufficiently long time, say T, to make sure γ^T is sufficiently small such that we can truncate the summation as $\sum_{i=0}^{T} \gamma^i R_{t+i+1}$ without incurring large bias, or wait for a time step T such that $R_t \equiv 0$ for all $t \geq T$. Since T can be very large, the update to $v(S_t)$ is likely to be heavily delayed. Even worse, the variance of G_t can be very large.

Instead of waiting for a realization of G_t to construct an update target for $v(S_t)$, TD uses $R_{t+1} + \gamma v(S_{t+1})$ as the update target for $v(S_t)$, which is readily available at time step $t+1$. The TD update target is recursively constructed by using the estimate v itself, which is the central idea of bootstrapping (Sutton, 1988). The difference term $R_{t+1} + \gamma v(S_{t+1}) - v(S_t)$ is usually referred to as the TD error. Moreover, since the randomness of the TD update target comes from only time step $t+1$, it usually enjoys lower variance than the Monte Carlo update target G_t. Obviously there is no free lunch, the TD update target is biased since v is only an estimate of v_π. However, under mild conditions, one can prove that v converges to v_π almost surely (a.s.) under the TD update (Bertsekas and Tsitsiklis, 1996).

The idea of bootstrapping has been widely used in RL. For example, SARSA (Rummery and Niranjan, 1994) for prediction updates updates q as

$$q(S_t, A_t) \leftarrow q(S_t, A_t) + \alpha_t(R_{t+1} + \gamma q(S_{t+1}, A_{t+1}) - q(S_t, A_t)),$$

where q converges to q_π a.s. (Bertsekas and Tsitsiklis, 1996). To compute $\bar{J}_\pi$, differential TD (Sutton and Barto, 2018) updates v and $\bar{J}$ as

$$\begin{aligned} \bar{J} &\leftarrow \bar{J} + \alpha_t(R_{t+1} - \bar{J}), \\ v(S_t) &\leftarrow v(S_t) + \alpha_t(R_{t+1} - \bar{J} + v(S_{t+1}) - v(S_t)). \end{aligned}$$

where $\bar{J}$ converges to $\bar{J}_\pi$ and v converges to one solution of (2.5) (Zhang et al., 2021e). Similarly, differential SARSA for prediction updates q and $\bar{J}$ as

$$\begin{aligned} \bar{J} &\leftarrow \bar{J} + \alpha_t(R_{t+1} - \bar{J}), \\ q(S_t, A_t) &\leftarrow q(S_t, A_t) + \alpha_t(R_{t+1} - \bar{J} + q(S_{t+1}, A_{t+1}) - q(S_t, A_t)), \end{aligned}$$

where $\bar{J}$ converges to $\bar{J}_\pi$ and q converges to one solution of (2.6) (Zhang et al., 2021e).

2.5 Function Approximation

In this section, we discuss the second ingredient of the deadly triad.

When the state space $\mathcal{S}$ is too large, maintaining a look-up table (i.e., a vector $v \in \mathbb{R}^{|\mathcal{S}|}$) as an estimate of v_π becomes intractable. Even worse, such a look-up table v cannot easily provide generalization across states. For example, even if we have good estimates for all states except for s_0, i.e., $v(s) = v_\pi(s)$ holds for all $s \in \mathcal{S} \setminus \{s_0\}$, it is still nontrivial to deduce the value of the state s_0. To cope well with large state spaces and provide generalization across different states, function approximation is introduced.

Function approximation adopts a feature mapping $x : \mathcal{S} \to \mathbb{R}^K$, which maps each state s into a K-dimensional vector. In the case of *linear function approximation*, a weight vector $w \in \mathbb{R}^K$ is then adapted such that

$$x(s)^\top w \approx v_\pi(s), \quad \forall s.$$

As K is usually much smaller than $|\mathcal{S}|$, function approximation exhibits great memory efficiency when the state space is large. As $x(s)$ is usually correlated for different $s \in \mathcal{S}$, function approximation naturally provide generalization across different states.

With linear function approximation, TD updates the weight w iteratively as

$$w_{t+1} \doteq w_t + \alpha_t(R_{t+1} + \gamma x_{t+1}^\top w_t - x_t^\top w_t)x_t, \tag{2.9}$$

where we use $x_t \doteq x(S_t)$ as shorthand. The almost sure convergence of $\{w_t\}$ is obtained by Tsitsiklis and Roy (1996a) under mild conditions.

Let $X \in \mathbb{R}^{|\mathcal{S}| \times K}$ be the feature matrix, each row of which is $x(s)^\top$, and $d_\pi \in \mathbb{R}^{|\mathcal{S}|}$ be the stationary state distribution of the Markov chain induced by π, assuming it exists. We define a projection $\Pi_{d_\pi} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ such that

$$\Pi_{d_\pi} v \doteq X \arg\min_{w \in \mathbb{R}^K} \|Xw - v\|_{d_\pi}^2.$$

It is easy to see that Π_{d_π} is the projection onto the column space of X w.r.t. the norm induced by d_π. With simple algebraic manipulation, it is easy to verify that Π_{d_π} is linear and

$$\Pi_{d_\pi} = X(X^\top D_\pi X)^{-1} X^\top D_\pi,$$

where $D_\pi \doteq diag(d_\pi) \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ and we assume the inversion exists for now. Under mild assumptions, Tsitsiklis and Roy (1996a) show that the iterate $\{w_t\}$ generated by linear TD (2.9) satisfies

$$\lim_{t \to \infty} w_t = w_* \quad a.s.,$$

where w_* is the unique solution to

$$\|\Pi_{d_\pi} \mathcal{T}_\pi(Xw) - Xw\|_{d_\pi}^2 = 0, \tag{2.10}$$

the LHS of which is known as the *Mean Squared Projected Bellman Error* (MSPBE).

Linear function approximation can also be combined with many other RL algorithms and some of the combinations remain convergent as well. For example, to approximate $q_\pi(s, a)$, linear SARSA for prediction updates the weight w iteratively as

$$w_{t+1} \doteq w_t + \alpha_t(R_{t+1} + \gamma x_{t+1}^\top w_t - x_t^\top w_t)x_t, \tag{2.11}$$

where we have overloaded x to also denote a state-action feature mapping: $\mathcal{S} \times \mathcal{A} \to \mathbb{R}^K$ and $x_t \doteq x(S_t, A_t)$. We overload $d_\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ to denote the stationary state-action distribution of the chain induced by π as well, assuming it exists. Similarly, $D_\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ is overloaded to denote a diagonal matrix whose diagonal entry is $d_\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, $X \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times K}$ is overloaded to denote the state-action feature matrix, each row of which is $x(s, a)^\top$. The projection Π_{d_π} is also overloaded accordingly. Under mild conditions, the results in Tsitsiklis and Roy (1996a) can be used to show that the $\{w_t\}$ generated by linear SARSA for prediction (2.11) converges to the unique solution of (2.10) (with overloaded notations).

Linear differential TD (Tsitsiklis and Roy, 1999) updates the weight w and a scalar estimate $\bar{J}$ of the reward iteratively as

$$\begin{aligned}
\bar{J}_{t+1} &\doteq \bar{J}_t + \alpha_t(R_{t+1} - \bar{J}_t), \\
w_{t+1} &\doteq w_t + \alpha_t(R_{t+1} - \bar{J}_t + x_{t+1}^\top w_t - x_t^\top w_t).
\end{aligned}$$

To study the behavior of linear differential TD, we make the following assumption:

Assumption 2.2. *For any $c \in \mathbb{R}, w \in \mathbb{R}^K$, $Xw \neq c1$.*

Under Assumptions 2.1, 2.2, and other mild conditions, Tsitsiklis and Roy (1999) prove that

$$\begin{aligned}
\lim_{t\to\infty} \bar{J}_t &= \bar{J}_\pi, \\
\lim_{t\to\infty} w_t &= w_*,
\end{aligned}$$

where w_* is the unique solution of

$$\left\|\Pi_{d_\pi}(r_\pi - \bar{J}_\pi 1 + P_\pi X w) - Xw\right\|_{d_\pi}^2 = 0.$$

Linear differential SARSA for prediction and its convergence are the same as linear differential TD (with overloaded notations) and are thus omitted.

Besides linear function approximation, *nonlinear function approximation* can also be used, where the estimate v is a nonlinear w.r.t. the feature $x(s)$. We use $v_w(s)$ and $q_w(s,a)$ to denote state value estimate and action value estimate parameterized by w respectively.

2.6 Off-Policy Learning

In this section, we discuss the third ingredient of the deadly triad.

So far we have considered only a single policy π, which is both the policy used for action selection and the policy whose value function is to be estimated. This learning paradigm is called *on-policy* learning. One could of course consider two policies instead: one policy μ is used for action selection, and we estimate the value function of another policy π. This learning paradigm is called *off-policy* learning, and the policies μ and π are referred to as the *behavior policy* and the *target policy* respectively.

Arguably, off-policy learning is more flexible than on-policy learning. For example, while a single policy μ is used for action selection, one could learn the value function of multiple target policies $\pi_1, \pi_2, \dots$ simultaneously (see, e.g., Sutton et al. (2011)), which would be impossible in the on-policy learning setting. In some settings, action selection based on the target policy may not be able to cover the whole state space efficiently. So one might also want to consider off-policy learning and use a different behavior policy for action selection to explore the state space more efficiently.

In this thesis, we consider two different off-policy learning settings, the *Markovian* setting and the *i.i.d.* setting.

Definition 2.1 (The Markovian setting)**.** *The learning algorithm is presented with an infinite sequence* $(S_0, A_0, R_1, S_1, A_1, R_2, S_2, \dots)$*, where*

$$S_0 \sim p_0(\cdot), A_t \sim \mu_t(\cdot|S_t), R_{t+1} \doteq r(S_t, A_t), S_{t+1} \sim p(\cdot|S_t, A_t) \quad (t = 0, 1, \dots).$$

Here $\{\mu_t\}$ *is a sequence of behavior policies for action selection.*

In its simplest form, the behavior policy is fixed.

Definition 2.2 (The Markovian setting with a fixed behavior policy)**.** *The learning algorithm is presented with an infinite sequence* $(S_0, A_0, R_1, S_1, A_1, R_2, S_2, \dots)$*, where*

$$S_0 \sim p_0(\cdot), A_t \sim \mu(\cdot|S_t), R_{t+1} \doteq r(S_t, A_t), S_{t+1} \sim p(\cdot|S_t, A_t) \quad (t = 0, 1, \dots).$$

We use $d_{\mu_t} \in \mathbb{R}^{|\mathcal{S}|}$ to denote its stationary state distribution. If the behavior policy is fixed, we simply use $d_\mu \in \mathbb{R}^{|\mathcal{S}|}$.

Besides the Markovian setting, we sometimes work on the i.i.d. setting, which has two variants. When working with state value function v_π or $\bar{v}_\pi$, we consider the following i.i.d. setting:

Definition 2.3 (The i.i.d. setting)**.** *The learning algorithm is presented with an infinite sequence of tuples* $\{(S_k, A_k, R_k, S'_k)\}_{k=0,1,\dots}$*, where*

$$S_k \sim d_\mu(\cdot), A_k \sim \mu(\cdot|S_k), R_k \doteq r(S_k, A_k), S'_k \sim p(\cdot|S_k, A_k).$$

Here d_μ *is the stationary distribution of the chain induced by the behavior policy* μ*.*

When working with action value function q_π or $\bar{q}_\pi$, we consider the following i.i.d. setting:

Definition 2.4 (The behavior agnostic i.i.d. setting)**.** *The learning algorithm is presented with an infinite sequence of tuples* $\{(S^0_k, A^0_k, S_k, A_k, R_k, S'_k, A'_k)\}_{k=0,1,\dots}$*, where*

$$S^0_k \sim p_0(\cdot), A^0_k \sim \pi(\cdot|S^0_k), (S_k, A_k) \sim d_\mu(\cdot), R_k \doteq r(S_k, A_k),$$
$$S'_k \sim p(\cdot|S_k, A_k), A'_k \sim \pi(\cdot|S'_k).$$

Here $d_\mu : \mathcal{S} \times \mathcal{A} \to (0, 1)$ *is an arbitrary probability distribution on* $\mathcal{S} \times \mathcal{A}$ *that has full support. For simplifying notations, we sometimes use* $d_{p_0\pi}$ *to denote the joint distribution of* (S^0_k, A^0_k) *and* $d_{\mu p \pi}$ *to denote the joint distribution of* (S_k, A_k, S'_k, A'_k)*.*

In the Markovian setting in Definition 2.2, after the chain induced by μ has mixed, it reduces to the i.i.d. setting in Definition 2.3. In other words, if an algorithm is convergent in the i.i.d. setting in Definition 2.3, it is usually also convergent in the Markovian setting in Definition 2.2 (see, e.g., Sutton et al. (2009) and Wang et al. (2017a)). The convergence proof in the Markovian setting is, however, usually more involved. In this thesis, we study those two settings separately for the ease of presentation.

The i.i.d. setting in Definition 2.4 is usually referred to as the *behavior agnostic off-policy learning setting* (Nachum et al., 2019a) since it does not impose any condition on how d_μ is obtained: it can be the stationary state-action pair distribution of a single known behavior policy; it can also result from multiple unknown behavior policies. In practice, this d_μ is usually the marginal state-action pair distribution in a large dataset consisting of previously logged tuples $\{(s_k, a_k, r_k, s'_k)\}$ (see, e.g., Levine et al. (2020)).

The i.i.d. settings can be viewed as the batch RL setting (Levine et al., 2020) with infinitely many data. It is straightforward to extend the results in those settings to the canonical batch RL setting with a finite dataset, see, e.g., Nachum et al. (2019a). The results in the Markovian settings typically require an infinite trajectory. Extending those results to the batch RL setting is nontrivial, which we leave for future work.

We now give some examples of off-policy algorithms. We first consider the Markovian setting in Definition 2.2. Off-policy TD updates the look-up table v as

$$v(S_t) \leftarrow v(S_t) + \alpha_t \rho_t (R_{t+1} + \gamma v(S_{t+1}) - v(S_t)), \tag{2.12}$$

where

$$\rho_t \doteq \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$$

is the *importance sampling ratio*. It is proven that under mild conditions, v converges to v_π a.s. (Bertsekas and Tsitsiklis, 1996). Off-policy SARSA for prediction updates the look-up table q as

$$q(S_t, A_t) \leftarrow q(S_t, A_t) + \alpha_t (R_{t+1} + \gamma \rho_{t+1} q(S_{t+1}, A_{t+1}) - q(S_t, A_t)). \tag{2.13}$$

Off-policy expected SARSA for prediction updates the look-up table q as

$$q(S_t, A_t) \leftarrow q(S_t, A_t) + \alpha_t \left(R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) q(S_{t+1}, a) - q(S_t, A_t) \right). \tag{2.14}$$

It is proven that under mild conditions, q converges to q_π a.s. (Bertsekas and Tsitsiklis, 1996). Differential off-policy TD updates v and $\bar{J}$ as

$$\begin{aligned} \delta_t &\leftarrow R_{t+1} - \bar{J} + v(S_{t+1}) - v(S_t), \\ \bar{J} &\leftarrow \bar{J} + \alpha_t \rho_t \delta_t, \\ v(S_t) &\leftarrow v(S_t) + \alpha_t \rho_t \delta_t. \end{aligned}$$

It is proven that $\bar{J}$ converges to $\bar{J}_\pi$ a.s. and v converges to one solution of (2.5) (Wan et al., 2021). Differential off-policy SARSA for prediction updates q and $\bar{J}$ as

$$\begin{aligned} \delta_t &\leftarrow R_{t+1} - \bar{J} + \rho_{t+1} q(S_{t+1}, A_{t+1}) - q(S_t, A_t), \\ \bar{J} &\leftarrow \bar{J} + \alpha_t \delta_t, \\ q(S_t, A_t) &\leftarrow q(S_t, A_t) + \alpha_t \delta_t. \end{aligned} \tag{2.15}$$

The results in Wan et al. (2021) can be used to show that $\bar{J}$ converges to $\bar{J}_\pi$ a.s. and q converges to one solution of (2.6).

We then consider the Markovian setting (Definition 2.1) with changing behavior policies. Q-learning (Watkins, 1989) updates q as

$$q(S_t, A_t) \leftarrow q(S_t, A_t) + \alpha_t \left(R_{t+1} + \gamma \max_a q(S_{t+1}, a) - q(S_t, A_t) \right), \tag{2.16}$$

where q converges to q_* a.s. (Bertsekas and Tsitsiklis, 1996). Comparing (2.16) and (2.14), one can see that the Q-learning is essentially using a greedy policy w.r.t. the current q as the target policy in the off-policy expected SARSA for prediction. Since the estimate q keeps changing, the target policy changes every time step in Q-learning. In practice, the behavior policy μ_t in Q-learning also changes every time step. A common choice is an ϵ-greedy policy w.r.t. the current q, i.e., it selects a random action w.p. ϵ and selects a greedy action w.r.t. q w.p. $1 - \epsilon$. Differential Q-learning (Wan et al., 2021) updates q and $\bar{J}$ as

$$\begin{aligned} \delta_t &\leftarrow R_{t+1} - \bar{J} + \max_a q(S_{t+1}, a) - q(S_t, A_t), \\ \bar{J} &\leftarrow \bar{J} + \alpha_t \delta_t, \\ q(S_t, A_t) &\leftarrow q(S_t, A_t) + \alpha_t \delta_t, \end{aligned} \tag{2.17}$$

where $\bar{J}$ converges to $\bar{J}_*$ a.s. and q converges to one solution of (2.7) a.s. (Wan et al., 2021). The behavior policy of differential Q-learning is usually an ϵ-greedy policy as well.

2.7 The Deadly Triad

Given the benefits of bootstrapping, function approximation, and off-policy learning, it would be desirable if one single algorithm can simultaneously adopt all those three ingredients. Such an algorithm, unfortunately, is usually not guaranteed to be convergent, which is the notorious deadly triad (Chapter 11.3 of Sutton and Barto (2018)).

For example, in the Markovian setting in Definition 2.2, off-policy linear TD updates the weight w iteratively as

$$w_{t+1} \doteq w_t + \alpha_t \rho_t (R_{t+1} + \gamma x_{t+1}^\top w_t - x_t^\top w_t) x_t. \tag{2.18}$$

The divergence of (2.18) is, however, well-documented (Baird, 1995; Tsitsiklis and Roy, 1996a; Sutton and Barto, 2018). In the on-policy setting, the linear TD update (2.9) can be rewritten as

$$w_{t+1} - w_t = \alpha_t \left(x_t (\gamma x_{t+1} - x_t)^\top w_t + R_{t+1} x_t \right). \tag{2.19}$$

Since the chain is assumed to be ergodic, asymptotically, S_t will be distributed according to the stationary distribution d_π. Consequently, in the long run, the expectation of the terms $x_t(\gamma x_{t+1} - x_t)^\top$ and $R_{t+1}x_t$ are

$$A \doteq \sum_s d_\pi(s)x(s)\left(\gamma \sum_{s'} P_\pi(s,s')x(s') - x(s)\right)^\top = X^\top D_\pi(\gamma P_\pi - I)X$$

and

$$b \doteq \sum_s d_\pi(s)x(s)r_\pi(s) = X^\top D_\pi r_\pi.$$

If the learning rate α_t decays properly, the system described by (2.19) approximates the following Ordinary Differential Equation (ODE) in the long run:

$$\frac{\mathrm{d}w(t)}{\mathrm{d}t} = Aw(t) + b. \tag{2.20}$$

The system in (2.19) is discrete-time and stochastic. The ODE in (2.20) is continuous-time and deterministic. Studying the convergence of RL algorithms like (2.19) via studying the corresponding ODEs like (2.20) is a common practice in the RL community, see, e.g., Tsitsiklis and Roy (1996a, 1999); Sutton et al. (2008, 2009); Yu (2015), where theoretical results from the stochastic approximation community (e.g., Kushner and Yin (2003); Borkar (2009)) are usually used to formally connect the RL algorithms and the corresponding ODEs. Technically speaking, if the RL algorithm satisfies some conditions (e.g., the iterates generated by the RL algorithm is bounded almost surely, the learning rates used by the RL algorithm decays properly), then the iterates generated by the RL algorithm converge to an invariant set of the ODE almost surely. We refer the reader to Section A.2 for an example of such theoretical results, which we repeatedly use in this thesis, and to Kushner and Yin (2003); Borkar (2009) for more discussion in depth. In the following we provide some intuition about why this line of research is fruitful. First, the ODE is deterministic via averaging out the randomness of RL algorithms from the policy and the transition function. A deterministic system is usually easier to analyze than a stochastic system. Second, the ODE community exists much longer than the RL community. Via connecting RL algorithms with ODEs, the RL community can make use of powerful existing results in the ODE community.

It is well-known that an ODE like (2.20) is globally asymptotically stable if A is negative definite (see, e.g., Vidyasagar (2002)). Such negative definiteness of A is confirmed by Tsitsiklis and Roy (1996a) with the help of an important property

of stationary distribution $d_\pi^\top P_\pi = d_\pi^\top$. Consequently, the convergence of (2.19) is expected. In the off-policy setting, the corresponding A matrix of (2.18) is

$$X^\top D_\mu(\gamma P_\pi - I)X,$$

where $D_\mu \doteq diag(d_\mu)$. Such an A matrix is, however, not guaranteed to be negative definite because $d_\mu^\top P_\pi = d_\mu^\top$ usually does not hold. Consequently, the off-policy linear TD update (2.18) can possibly diverge. This lack of negative definiteness of the expected limiting update is the root cause of the deadly triad. Similarly, linear Q-learning updates the weight iteratively as

$$w_{t+1} \doteq w_t + \alpha_t \left(R_{t+1} + \gamma \max_a x(S_{t+1}, a)^\top w_t - x_t^\top w_t\right) x_t, \tag{2.21}$$

whose divergence is documented in Baird (1995).

One attempt to address the deadly triad is to reweight the off-policy update, e.g., (2.18), with the products of importance sampling ratios

$$\rho_0 \rho_1 \dots \rho_t$$

instead of a single importance sampling ratio ρ_t. Assuming the variance of all such products is bounded, Precup et al. (2001) confirm the convergence of the corresponding update rule. Such an assumption about the bounded variance is, however, restrictive and may not hold for many behavior and target policies especially in infinite horizon MDPs we consider in this thesis.

Residual gradient algorithms (Baird, 1995) address the deadly triad by performing gradient descent on TD errors directly. Consider, e.g., the off-policy prediction problem with linear function approximation in the i.i.d setting (Definition 2.3). One could define the *Mean Squared Bellman Error* (MSBE) for w as

$$\text{MSBE}(w) \doteq \|r_\pi + \gamma P_\pi X w - X w\|_{d_\mu}^2.$$

Computing the gradient of MSBE(w) yields

$$\begin{aligned}
&\nabla_w \text{MSBE}(w) \\
=&2(r_\pi + \gamma P_\pi X w - X w)^\top D_\mu(\gamma P_\pi X - X) \\
=&2\sum_s d_\mu(s) \left(\sum_{s'} P_\pi(s, s')(r_\pi(s) + \gamma x(s')^\top w - x(s)^\top w)\right) \\
&\times \left(\sum_{s''} P_\pi(s, s'')(\gamma x(s'') - x(s))^\top\right).
\end{aligned}$$

If we could get an unbiased estimate of $\nabla_w \text{MSBE}(w)$, standard stochastic gradient descent algorithms can be applied to minimize $\text{MSBE}(w)$, and the convergence of the resulting algorithm follows directly from standard optimization theories. Such an unbiased gradient estimate, however, requires two independently sampled successor states s' and s'' from a single state s, which is not available in the RL setting we consider. This is the notorious *double sampling* issue.

It is important to note that when it comes to the deadly triad, we limit our discussion to only *computationally efficient* algorithms, i.e., algorithms that are incremental and have linear per-step computational and memory complexity w.r.t. the feature dimension K. If we relax this requirement, the deadly triad can be addressed via least square TD (LSTD) methods (Bradtke and Barto, 1996; Boyan, 1999; Lagoudakis and Parr, 2003; Peters and Schaal, 2008; Yu, 2010). If off-policy linear TD (2.18) converged, the weight w would converge to w_* satisfying

$$Aw_* + b = 0,$$

where

$$\begin{aligned} A \doteq & X^\top D_\mu (\gamma P_\pi - I) X, \\ b \doteq & X^\top D_\mu r_\pi. \end{aligned}$$

Thus one straightforward way to compute w_* is to estimate both A and b directly and then use matrix inversion to compute w_*, which is the central idea of LSTD methods. LSTD methods, however, in general exhibit $\mathcal{O}(K^2)$ per-step computational and memory complexity, which is prohibitive in large-scale applications. When additional sparsity assumption is imposed on the features, Geramifard et al. (2006) reduce the per-step computational complexity of LSTD methods to $\mathcal{O}(K + \bar{K}^2)$, where $\bar{K}$ is the maximum nonzero features for any state. The memory complexity, however, remains $\mathcal{O}(K^2)$.

Nevertheless, several computationally efficient algorithms have been developed to address the deadly triad in some settings, which we detail in the following sections.

2.8 Gradient Temporal Difference Learning

In this section, we discuss the first family of algorithms that address the deadly triad. The idea presented in this section is repeatedly used in Chapters 4, 6, 8, and 12 for deriving different learning objectives and computing their gradients.

Linear TD (2.9) can be regarded as performing gradient descent on the following objective

$$\left(R_{t+1} + \gamma x_{t+1}^\top w - x_t^\top w\right)^2$$

w.r.t. w, pretending $x_{t+1}^\top w$ is independent of w. Such a method is usually referred to as a *semi-gradient* method since the gradient of $x_{t+1}^\top w$ is ignored. If this gradient of $x_{t+1}^\top w$ is not ignored, one would end up with a true stochastic gradient method, i.e., residual gradients. As discussed before, residual gradient remains stable even if under off-policy training. Residual gradient, however, suffers from the notorious double sampling issue, yielding implementation challenges in many practical settings. It thus would be desirable if one could have a true stochastic gradient descent method while avoiding the double sampling issue. Fortunately, Gradient Temporal Difference (GTD, Sutton et al. (2008, 2009); Maei (2011)) learning methods achieve such a goal.

We study GTD2 (Sutton et al., 2009) as a representative of GTD methods. The on-policy linear TD (2.9) converges to the minimizer of the on-policy MSPBE

$$\|\Pi_{d_\pi} \mathcal{T}_\pi(Xw) - Xw\|_{d_\pi}^2.$$

Consider the i.i.d. setting in Definition 2.3. Since the data is sampled from the stationary distribution d_μ, one natural objective for learning the weight w, in analogue to the MSPBE in the on-policy setting, is the off-policy MSPBE:

$$\text{MSPBE}(w) \doteq \left\|\Pi_{d_\mu} \mathcal{T}_\pi(Xw) - Xw\right\|_{d_\mu}^2 \tag{2.22}$$

Similar to residual gradient methods, which compute the gradient of $\text{MSBE}(w)$ directly, GTD2 computes the gradient of $\text{MSPBE}(w)$ directly. We first rewrite the off-policy $\text{MSPBE}(w)$ as

$$\begin{aligned}
&\text{MSPBE}(w) \qquad (2.23)\\
=&\left\|\Pi_{d_\mu}(\mathcal{T}_\pi(Xw) - Xw)\right\|_{d_\mu}^2\\
=&(\mathcal{T}_\pi(Xw) - Xw))^\top \Pi_{d_\mu}^\top D_\mu \Pi_{d_\mu}(\mathcal{T}_\pi(Xw) - Xw))\\
=&(\mathcal{T}_\pi(Xw) - Xw))^\top D_\mu X(X^\top D_\mu X)^{-1} X^\top D_\mu(\mathcal{T}_\pi(Xw) - Xw))\\
&\qquad (\text{Using } \Pi_{d_\mu} \doteq X(X^\top D_\mu X)^{-1} X^\top D_\mu)\\
=&\left\|X^\top D_\mu(\mathcal{T}_\pi(Xw) - Xw))\right\|_{(X^\top D_\mu X)^{-1}}^2\\
=&\left\|X^\top D_\mu(r_\pi + \gamma P_\pi Xw - Xw)\right\|_{(X^\top D_\mu X)^{-1}}^2\\
=&\|A_{\pi,\mu} w + b_{\pi,\mu}\|_{C_\mu^{-1}}^2,
\end{aligned}$$

where

$$\begin{aligned} A_{\pi,\mu} &\doteq X^\top D_\mu(\gamma P_\pi - I)X, \\ b_{\pi,\mu} &\doteq X^\top D_\mu r_\pi, \\ C_\mu &\doteq X^\top D_\mu X. \end{aligned}$$

According to Fenchel's duality, for any positive definite matrix $M \in \mathbb{R}^{K\times K}$ and any vector $y \in \mathbb{R}^K$, we have

$$y^\top M^{-1} y = \max_{\nu\in\mathbb{R}^K} 2y^\top \nu - \nu^\top M \nu. \tag{2.24}$$

We can then continue expanding MSPBE(w) as

$$\text{MSPBE}(w) = \max_\nu 2(A_{\pi,\mu}w + b_{\pi,\mu})^\top \nu - \nu^\top C_\mu \nu.$$

Let

$$L(w,\nu) \doteq 2(A_{\pi,\mu}w + b_{\pi,\mu})^\top \nu - \nu^\top C_\mu \nu,$$

we have

$$\min_w \text{MSPBE}(w) = \min_w \max_\nu L(w,\nu).$$

Since $L(w,\nu)$ is concave in ν and convex in w, the original problem of minimizing w for MSPBE(w) now becomes a convex-concave saddle-point (CCSP) problem. One can use primal-dual methods to solve this CCSP problem, i.e., update w and ν following $-\nabla_w L_{\pi,\mu}(w,\nu)$ and $\nabla_\nu L_{\pi,\mu}(w,\nu)$:

$$\begin{aligned} &\nabla_w L_{\pi,\mu}(w,\nu) \\ =&2\nu^\top A_{\pi,\mu} \\ =&2\nu^\top (X^\top D_\mu(\gamma P_\pi - I)X) \\ =&2\sum_s d_\mu(s)(x(s)^\top \nu)\sum_{s'} P_\pi(s,s')(\gamma x(s') - x(s))^\top, \\ &\nabla_\nu L_{\pi,\nu}(w,\nu) \\ =&2((A_{\pi,\mu}w + b_{\pi,\mu})^\top - \nu^\top C_\mu) \\ =&2((r_\pi + \gamma P_\pi X w - Xw)^\top D_\mu X - \nu^\top X^\top D_\mu X) \\ =&2\sum_s d_\mu(s)x(s)^\top \sum_{a,s'} \pi(a|s)p(s'|s,a)\left(r(s,a) + \gamma x(s')^\top w - x(s)^\top w - x(s)^\top \nu\right). \end{aligned}$$

GTD2 updates w and ν iteratively as

$$\nu_{k+1} \doteq \nu_k + \alpha_k \rho_k (R_k + \gamma x_k'^\top w_k - x_k^\top w_k - x_k^\top \nu_k) x_k,$$
$$w_{k+1} \doteq w_k + \alpha_k \rho_k (x_k - \gamma x_k') x_k^\top \nu_k,$$

where ρ_k, x_k, and x_k' are shorthand for $\frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}$, $x(S_k)$, and $x(S_k')$. Under mild conditions, Sutton et al. (2009) prove that $\{w_k\}$ in GTD2 converges to the unique minimizer of the off-policy MSPBE(w) a.s. GTD2, therefore, solves the deadly triad in the discounted setting for the prediction problem.

The derivation of GTD2 presented here using the Fenchel's duality is due to Liu et al. (2015); Macua et al. (2015) and is different from the original derivation in Sutton et al. (2009). Besides GTD2, TDC (Sutton et al., 2009) can also be used to minimize the off-policy MSPBE(w). Moreover, there also exists GTD(0) (Sutton et al., 2008) that optimizes an objective that is slightly different from the off-policy MSPBE(w). GTD(0), GTD2, and TDC are all in the family of GTD algorithms and solve the deadly triad in the discounted setting for the prediction problem. In this thesis, for easing presentation, we use GTD to indicate the GTD2 algorithm unless otherwise specified. Though we consider the i.i.d. setting in Definition 2.3 here, GTD can also be used in the Markovian setting in Definition 2.2, see, e.g., Wang et al. (2017a). GTD methods have also been extended to estimate the action-value function q_π, see, e.g., Maei and Sutton (2010).

Maei et al. (2010) extend the idea of GTD methods to the control setting in Greedy-GQ. Consider the i.i.d. setting in Definition 2.4, Greedy-GQ considers the following MSPBE objective

$$\left\| \Pi_{d_\mu} \mathcal{T}_{\pi_w}(Xw) - Xw \right\|_{d_\mu}^2,$$

where X is the state-action feature matrix, $\mathcal{T}_{\pi_w}$ is the action value Bellman operator, and π_w is some policy parameterized by w, e.g., π_w can be a greedy policy w.r.t. to the action value estimate Xw. Arguably, optimizing this MSPBE is more involved than optimize (2.22) since this MSPBE is non-convex and can even be non-differentiable due to the dependence of π_w on the weight w. Fortunately, Greedy-GQ managed to optimize this MSPBE and is proven to converge to some stationary points of this MSPBE objective, making it a solution for the deadly triad in the discounted setting for the control problem.

2.9 Emphatic Temporal Difference Learning

In this section, we discuss the second family of algorithms that address the deadly triad. This section introduces a new quantity, *emphasis*, which is repeatedly used in Chapters 4, 5, 11 and 12. This new quantity plays a key role in the design of many novel convergent off-policy algorithms in this thesis and motivates the introduction of the retrospective knowledge.

As discussed in Section 2.7, the root cause of the deadly triad is that the matrix

$$X^\top D_\mu(\gamma P_\pi - I)X$$

is not guaranteed to be negative definite. One possible solution is to replace D_μ with some other distribution (e.g., by reweighting the update at each time step) to regain negative definiteness. This is the central idea of Emphatic Temporal Difference learning methods (ETD, Sutton et al. (2016)). In its simplest form, considering the Markovian setting in Definition 2.2, ETD introduces the *followon trace* F_t for reweighting the updates, which is defined recursively as

$$\begin{aligned} F_t &\doteq i(S_t) + \gamma\rho_{t-1}F_{t-1} \quad (t \geq 0) \\ F_{-1} &\doteq 0, \end{aligned} \tag{2.25}$$

where $i : \mathcal{S} \to (0, +\infty)$ is the *interest* function specifying users' preference of different states ($i(s) \equiv 1$ is usually used). ETD then updates the weight w iteratively as

$$w_{t+1} \doteq w_t + \alpha_t F_t \rho_t (R_{t+1} + \gamma x_{t+1}^\top w_t - x_t^\top w_t)x_t. \tag{2.26}$$

It can be shown (Sutton et al., 2016; Zhang et al., 2019, 2020d) that under mild conditions, the limit

$$m_{\pi,\mu}(s) \doteq \lim_{t\to\infty} \mathbb{E}_\mu[F_t | S_t = s] \tag{2.27}$$

exists and

$$m_{\pi,\mu} = D_\mu^{-1}(I - \gamma P_\pi^\top)^{-1} D_\mu i.$$

In this thesis, we refer to $m_{\pi,\mu}$ as *emphasis*. Then the limiting A matrix of ETD (2.26)

can be computed as

$$
\begin{aligned}
A &\doteq \lim_{t\to\infty} \mathbb{E}_\mu[F_t \rho_t x_t (\gamma x_{t+1} - x_t)^\top] \\
&= \lim_{t\to\infty} \sum_s d_\mu(s) \mathbb{E}_\mu[F_t \rho_t x_t (\gamma x_{t+1} - x_t)^\top | S_t = s] \\
&= \lim_{t\to\infty} \sum_s d_\mu(s) \mathbb{E}_\mu[F_t | S_t = s] \mathbb{E}_\mu[\rho_t x_t (\gamma x_{t+1} - x_t)^\top | S_t = s] \\
&= \sum_s d_\mu(s) m_{\pi,\mu}(s) \lim_{t\to\infty} \mathbb{E}_\mu[\rho_t x_t (\gamma x_{t+1} - x_t)^\top | S_t = s] \\
&= \sum_s d_\mu(s) m_{\pi,\mu}(s) x(s) \left(\sum_{s'} P_\pi(s, s') \gamma x(s')^\top - x(s)^\top \right) \\
&= X^\top D_{f_{\pi,\mu}} (\gamma P_\pi - I) X,
\end{aligned}
\tag{2.28}
$$

where $D_{f_{\pi,\mu}}$ is a diagonal matrix such that

$$
D_{f_{\pi,\mu}}(s, s) \doteq f_{\pi,\mu}(s) \doteq d_\mu(s) m_{\pi,\mu}(s).
$$

In other words, we have defined $f_{\pi,\mu}$ as

$$
f_{\pi,\mu} \doteq (I - \gamma P_\pi^\top)^{-1} D_\mu i \tag{2.29}
$$

Sutton et al. (2016) prove that such an A matrix is negative definite. Consequently, the convergence of ETD (2.26) is expected. This negative definiteness is the motivation for introducing the weighting scheme F_t. The rigorous convergence proof of ETD (2.26) is given by Yu (2015), making ETD another solution for the deadly triad in the discounted setting for the prediction problem.

2.10 Density Ratio Learning

In this section, we discuss the third family of algorithms that address the deadly triad issue. This section introduces the concept of *density ratio*, which is be repeatedly used in Chapter 6. Different from value function, density ratio provides a new perspective for off-policy policy evaluation and is a key quantity to be learned by several proposed algorithms.

When the goal is to estimate the discounted total rewards J_π, *density ratio* can

be used. By definition,

$$\begin{aligned} J_\pi &= \mathbb{E}_{s\sim p_0(\cdot)}[v_\pi(s)] \\ &= \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r(S_t, A_t) \mid p_0, p, \pi\right] \\ &= \sum_{s,a}\sum_{t=0}^{\infty}\Pr(S_t = s, A_t = a|p_0, p, \pi)\gamma^t r(s, a). \end{aligned}$$

If we define

$$d_{\pi,\gamma}(s) \doteq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(S_t = s|p_0, p, \pi) \quad (\gamma < 1)$$

and also overload it as

$$d_{\pi,\gamma}(s, a) \doteq d_{\pi,\gamma}(s)\pi(a|s),$$

we can then express the discounted total rewards as

$$J_\pi = \frac{1}{1-\gamma}\sum_{s,a} d_{\pi,\gamma}(s, a)r(s, a).$$

This $d_{\pi,\gamma}$ is usually referred to as the *normalized discounted state action pair occupancy* and it is easy to see

$$\sum_{s,a} d_{\pi,\gamma}(s, a) = 1.$$

Consider the i.i.d. setting in Definition 2.4, we can alternatively express J_π as

$$J_\pi = \frac{1}{1-\gamma}\sum_{s,a} d_\mu(s, a)\frac{d_{\pi,\gamma}(s, a)}{d_\mu(s, a)}r(s, a).$$

If the density ratio

$$\tau_\gamma(s, a) \doteq \frac{d_{\pi,\gamma}(s, a)}{d_\mu(s, a)}$$

is known, we can simply use

$$\frac{1}{1-\gamma}\tau_\gamma(S_k, A_k)R_k$$

as an unbiased estimator for J_π.

One standard approach in the machine learning community for learning the density ratio τ_γ (see, e.g., Sugiyama et al. (2012)) is to make use of the fact that τ_γ is the unique minimizer of the following optimization problem:

$$\min_{z \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}} L(z),$$

where

$$L(z) \doteq \frac{1}{2}\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}[z(s,a)^2] - \mathbb{E}_{(s,a)\sim d_{\pi,\gamma}(\cdot)}[z(s,a)].$$

Consequently, the problem of learning the density ratio becomes the problem of minimizing $L(z)$. Typically, SGD is used for such a minimization problem, which requires to obtain unbiased samples of $\nabla_z L(z)$. Getting such unbiased samples, however, requires to sample from both the denominator distribution (i.e., d_μ) and the nominator distribution (i.e., $d_{\pi,\gamma}$). In standard machine learning settings, access to samples from both distributions are usually assumed. In our off-policy learning setting, however, we have only samples from d_μ and obtaining samples from $d_{\pi,\gamma}$ is usually impractical.

To address this problem, Nachum et al. (2019a) use a change of variable trick. Since

$$\begin{aligned}
&d_{\pi,\gamma}(s,a)\\
=&(1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(S_t = s, A_t = a \mid p_0, p, \pi)\\
=&(1-\gamma)d_{p_0\pi}(s,a) + (1-\gamma)\sum_{t=1}^{\infty}\gamma^t \Pr(S_t = s, A_t = a \mid p_0, p, \pi)\\
=&(1-\gamma)d_{p_0\pi}(s,a) + (1-\gamma)\sum_{t=0}^{\infty}\gamma^{t+1} \Pr(S_{t+1} = s, A_{t+1} = a \mid p_0, p, \pi)\\
=&(1-\gamma)d_{p_0\pi}(s,a)\\
&+ (1-\gamma)\sum_{t=0}^{\infty}\gamma^{t+1}\sum_{\bar{s},\bar{a}} \Pr(S_t = \bar{s}, A_t = \bar{a} \mid p_0, p, \pi)P_\pi((\bar{s},\bar{a}),(s,a))\\
=&(1-\gamma)d_{p_0\pi}(s,a)\\
&+ \gamma\sum_{\bar{s},\bar{a}} P_\pi((\bar{s},\bar{a}),(s,a))(1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(S_t = \bar{s}, A_t = \bar{a} \mid p_0, p, \pi)\\
=&(1-\gamma)d_{p_0\pi}(s,a) + \gamma\sum_{\bar{s},\bar{a}} P_\pi((\bar{s},\bar{a}),(s,a))d_{\pi,\gamma}(\bar{s},\bar{a}),
\end{aligned}$$

we then have, in a matrix form, that

$$d_{\pi,\gamma} = (1-\gamma)d_{p_0\pi} + \gamma P_\pi^\top d_{\pi,\gamma},$$

implying

$$d_{\pi,\gamma} = (1-\gamma)(I - \gamma P_\pi^\top)^{-1} d_{p_0\pi}.$$

Let

$$\nu \doteq (I - \gamma P_\pi)^{-1} z. \tag{2.30}$$

It is then easy to see that

$$\begin{aligned}
\mathbb{E}_{(s,a)\sim d_{\pi,\gamma}(\cdot)}[z(s,a)] =& z^\top d_{\pi,\gamma} \\
=& (1-\gamma) z^\top (I - \gamma P_\pi^\top)^{-1} d_{p_0\pi} \\
=& (1-\gamma) \nu^\top d_{p_0\pi} \\
=& (1-\gamma) \mathbb{E}_{(s,a)\sim d_{p_0\pi}(\cdot)}[\nu(s,a)].
\end{aligned}$$

It is also easy to see that

$$z(s,a) = ((I - \gamma P_\pi)\nu)(s,a) = \nu(s,a) - \gamma \sum_{s',a'} P_\pi((s,a),(s',a'))\nu(s',a').$$

Consequently,

$$L(z) = \frac{1}{2}\mathbb{E}_{(s,a,s',a')\sim d_{\mu p\pi}(\cdot)}[(\nu(s,a) - \gamma\nu(s',a'))^2] + (1-\gamma)\mathbb{E}_{(s,a)\sim d_{p_0\pi}(\cdot)}[\nu(s,a)].$$

Since $(I - \gamma P_\pi)^{-1}$ has full rank, we have

$$\min_{z\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}} L(z) = \min_{\nu\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}} L'(\nu),$$

where

$$L'(\nu) \doteq \frac{1}{2}\mathbb{E}_{(s,a,s',a')\sim d_{\mu p\pi}(\cdot)}[(\nu(s,a) - \gamma\nu(s',a'))^2] + (1-\gamma)\mathbb{E}_{(s,a)\sim d_{p_0\pi}(\cdot)}[\nu(s,a)].$$

In other words, to find the minimizing z of $L(z)$, we can proceed by finding the minimizing ν of $L'(\nu)$ and then compute the minimizing z (i.e., the density ratio) with (2.30).

Though we have access to samples from $d_{\mu p\pi}$ and $d_{p_0\pi}$, optimizing $L'(\nu)$ via SGD is still impractical: obtaining an unbiased sample of $\nabla_\nu L'(\nu)$ runs into the double sampling issue again, just like residual algorithms. To address this issue, Nachum et al. (2019a) further expand $L'(\nu)$ as

$$\begin{aligned}
L'(\nu) =& \mathbb{E}_{(s,a,s',a')\sim d_{\mu p\pi}(\cdot)}[\max_{\tau\in\mathbb{R}}(\nu(s,a) - \gamma\nu(s',a'))\tau - \frac{1}{2}\tau^2] \\
&+ (1-\gamma)\mathbb{E}_{(s,a)\sim d_{p_0\pi}(\cdot)}[\nu(s,a)] \\
=& \max_{\tau\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}} \mathbb{E}_{(s,a,s',a')\sim d_{\mu p\pi}(\cdot)}[(\nu(s,a) - \gamma\nu(s',a'))\tau(s,a) - \frac{1}{2}\tau(s,a)^2] \\
&+ (1-\gamma)\mathbb{E}_{(s,a)\sim d_{p_0\pi}(\cdot)}[\nu(s,a)],
\end{aligned} \tag{2.31}$$

where the first equality results from Fenchel's duality and the second equality results from the interchangeability principle (Shapiro et al., 2014). Consequently, we have

$$\min_{\nu \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}} L'(\nu) = \min_{\nu \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}} \max_{\tau \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}} L(\nu, \tau), \tag{2.32}$$

where

$$\begin{aligned} L(\nu, \tau) \doteq & \mathbb{E}_{(s,a,s',a') \sim d_{\mu p \pi}(\cdot)}[(\nu(s,a) - \gamma \nu(s',a'))\tau(s,a) - \frac{1}{2}\tau(s,a)^2] \\ & + (1-\gamma)\mathbb{E}_{(s,a)\sim d_{p_0\pi}(\cdot)}[\nu(s,a)]. \end{aligned}$$

Since $L(\nu, \tau)$ is convex in ν and concave in τ, the minimax problem in (2.32) is a standard CCSP problem, and primal-dual methods can take over. Nachum et al. (2019a) show that τ_γ is in the saddle point of $L(\nu, \tau)$ and refer to the algorithm that uses primal-dual methods to solve this CCSP problem as *Dual stationary DIstribution Correction Estimation* (DualDICE).

DualDICE can be combined with function approximation easily. For example, when $\tau(s,a)$ and $\nu(s,a)$ are parameterized by w_τ and w_ν respectively, DualDICE considers the objective

$$\begin{aligned} L'(w_\nu, w_\tau) \doteq & \mathbb{E}_{(s,a,s',a') \sim d_{\mu p \pi}(\cdot)}[(\nu(s,a) - \gamma \nu(s',a'))\tau(s,a) - \frac{1}{2}\tau(s,a)^2] \\ & + (1-\gamma)\mathbb{E}_{(s,a)\sim d_{p_0\pi}(\cdot)}[\nu(s,a)] \end{aligned}$$

and updates w_ν and w_τ following $-\nabla_{w_\nu} L'(w_\nu, w_\tau)$ and $\nabla_{w_\tau} L'(w_\nu, w_\tau)$ respectively. Importantly, when τ and ν are linear in w_τ and w_ν respectively, the objective $L'(w_\nu, w_\tau)$ is convex in w_ν and concave in w_τ, i.e., DualDICE with linear function approximation solves a CCSP problem with primal dual algorithms. Its convergence is, therefore, expected (similar to the convergence proof of GTD in Sutton et al. (2009)), making DualDICE a solution for the deadly triad in the discounted setting for the prediction problem.

It is worth mentioning that Nachum et al. (2019a) exploits Fenchel's duality for solving the double sampling issue in DualDICE. Similarly, Fenchel's duality can also be used to solve the double sampling issue in residual gradient algorithms, which, however, will recover GTD.

In the average reward setting, it is easy to show (see, e.g., Puterman (2014)) that

$$\bar{J}_\pi = \sum_{s,a} d_\pi(s,a) r(s,a) = \sum_{s,a} d_\mu(s,a) \frac{d_\pi(s,a)}{d_\mu(s,a)} r(s,a).$$

In other words, if the density ratio $\frac{d_\pi(s,a)}{d_\mu(s,a)}$ is known, estimating the average reward $\bar{J}_\pi$ becomes trivial in the i.i.d. setting in Definition 2.4. In the rest of this thesis, we define

$$\tau_\gamma(s,a) \doteq \begin{cases} \frac{d_{\pi,\gamma}(s,a)}{d_\mu(s,a)}, & (\gamma < 1) \\ \frac{d_\pi(s,a)}{d_\mu(s,a)}, & (\gamma = 1) \end{cases}$$

for unifying notations in discounted and average reward settings. When $\gamma < 1$, recall that we have

$$d_{\pi,\gamma} = (1-\gamma) d_{p_0\pi} + \gamma P_\pi^\top d_{\pi,\gamma}. \tag{2.33}$$

When $\gamma = 1$, by the property of stationary distributions, we have

$$d_{\pi,\gamma} = P_\pi^\top d_{\pi,\gamma},$$

i.e., (2.33) holds for $\gamma = 1$ as well. Plugging $d_{\pi,\gamma} = D_\mu \tau_\gamma$ in (2.33) yields

$$D_\mu \tau_\gamma = (1-\gamma) d_{p_0\pi} + \gamma P_\pi^\top D_\mu \tau_\gamma,$$

which is another useful equation for learning τ_γ.

2.11 Target Network

In this section, we discuss *target networks*, a commonly used *empirical* technique for mitigating the deadly triad. We theoretically study target networks in different settings in Chapters 3, 7, and 9 as a tool for breaking the deadly triad, providing theoretical understanding for the conventional wisdom that target networks stabilize learning. We also extend target networks to bidirectional target networks in Section 10 for residual gradient algorithms.

The core idea of bootstrapping is to construct an update target for the estimate recursively by using the estimate itself. The estimate is updated every step, and so does the update target. Consequently, the update target becomes nonstationary and instability arises. The idea of the target network (Mnih et al., 2015) is to slow down the change of the update target, thus the instability from the nonstationarity of the update target can be reduced. To achieve this, Mnih et al. (2015) keep a copy of the parameters of the function approximator and construct update targets with that copy, instead of the original parameters. The copy is synchronized with the original parameters only periodically. It thus changes much slower. Since the copy is mainly

used to construct update targets, it is usually referred to as *target network*. In this thesis, we refer to the original parameters as *main network*.

We use Deep-Q-Networks (DQN, Mnih et al. (2015)) to demonstrate how a target network works. Let $q_w(s,a)$ be our estimate for $q_*(s,a)$. Here q_w indicates that the function q is parameterized by a weight vector w. In the case of linear function approximation,

$$q_w(s,a) \doteq x(s,a)^\top w.$$

In general, q_w does not have to have a linear form. Consider the Markovian setting in Definition 2.1, DQN updates w iteratively as

$$w_{t+1} \doteq w_t + \alpha \left(R_{t+1} + \gamma \max_a q_{\bar{w}_t}(S_{t+1}, a) - q_{w_t}(S_t, A_t) \right) \nabla_w q_{w_t}(S_t, A_t),$$
$$\bar{w}_{t+1} \doteq \begin{cases} w_t, & t\%I == 0 \\ \bar{w}_t, & \text{otherwise} \end{cases},$$

where I is a hyperparameter, i.e., the target network $\bar{w}$ is synchronized every I steps. Note we have removed many other ingredients of DQN, e.g., experience replay, in the above equation for easing presentation. Besides this periodic target network update, Lillicrap et al. (2016) proposes a Polyak-averaging target network update, which updates $\bar{w}$ as

$$\bar{w}_{t+1} \doteq (1-\beta)\bar{w}_t + \beta w_t, \tag{2.34}$$

where β is a hyperparameter determining the portion of the target network to be updated at each step.

Overall target networks have achieved great empirical success when q_w is deep networks and the conventional wisdom is that target networks stabilize training (Mnih et al., 2015; Lillicrap et al., 2016; Haarnoja et al., 2018). Lee and He (2019a) study target networks theoretically in the on-policy setting with linear function approximation for prediction problems. van Hasselt et al. (2018) empirically study the role of a target network in the deadly triad setting in deep RL. However, a theoretical study of target networks in the context of the deadly triad is still missing.

2.12 Actor Critic Methods

In this section, we discuss *policy-based* methods for the control problem to prepare us for the methods discussed in Chapter 12.

So far we have considered only *value-based* methods for the control problem, i.e., we first estimate the action-value function and then derive a policy from the action-value function, e.g., an ϵ-greedy policy. *Policy-based* methods is another family of methods for solving the control problem, where the policy π is parameterized by parameters θ directly. We then update θ directly to improve certain performance metrics of π. We use $\pi_\theta(a|s)$ to denote the parameterized policy. In the rest of the thesis, when it does not confuse, we drop the subscript θ in π_θ for easing presentation.

REINFORCE (Williams, 1992) is perhaps the earliest policy-based control method in RL. In the discounted setting, REINFORCE updates θ in the direction of $\nabla_\theta J_\pi$, which is computed by the policy gradient theorem (Sutton et al., 1999a) as

$$\nabla_\theta J_\pi \propto \mathbb{E}_{(s,a)\sim d_{\pi,\gamma}(\cdot)}[q_\pi(s,a)\nabla_\theta \log \pi_\theta(a|s)].$$

Consider the Markovian on-policy setting where a trajectory $(S_0, A_0, R_1, S_2, \ldots, S_T)$ is obtained by executing the policy π_θ and T denotes a time step that the rewards are all zero afterwards. At time step t, $\nabla_\theta \log \pi_{\theta_t}(A_t|S_t)$ is immediately available and G_t is an unbiased estimator of $q_\pi(S_t, A_t)$. REINFORCE then updates θ for $t = 0, 1, \ldots, T-1$ as

$$\theta \leftarrow \theta + \alpha_t \gamma^k G_t \nabla_\theta \log \pi_\theta(A_t|S_t).$$

REINFORCE, however, has at least two main disadvantages. First, the return G_t usually has a large variance, making the overall update unstable. Second, such an update cannot be performed until G_t is available, which might take a long time if T is large. To address those issues, besides using a parameterized function for representing the policy π, Sutton et al. (1999a) use another parameterized function to estimate the action-value function q_π as well. Here the policy π is referred to as the *actor*, the action-value function is referred to as the *critic*, and the whole algorithm is referred to as an actor-critic method. Let $q_w(s,a)$ be the function parameterized by w to estimate q_π, the canonical actor-critic algorithm updates θ and w iteratively as

$$\begin{aligned} w_{t+1} &\doteq w_t + \alpha_t \left(R_{t+1} + \gamma q_{w_t}(S_{t+1}, A_{t+1}) - q_{w_t}(S_t, A_t)\right) \nabla_w q_{w_t}(S_t, A_t), \\ \theta_{t+1} &\doteq \theta_t + \beta_t \gamma^t q_{w_t}(S_t, A_t) \nabla_\theta \log \pi_{\theta_t}(A_t|S_t), \end{aligned} \tag{2.35}$$

where $A_t \sim \pi_{\theta_t}(\cdot|S_t)$ and β_t is another sequence of learning rates such that the update to w is much faster than the update to θ. Consequently, the actor is quasi-stationary from the critic's view. When $q_w(s,a)$ is linear in w, under mild conditions and with additional adaptive learning rates, the convergence of (2.35) is given by Konda and

Tsitsiklis (1999). The critic iterates $\{w_t\}$ track the action-value function of the actor π in the sense that

$$\lim_{t\to\infty} \left\| \Pi_{d_{\pi_{\theta_t}}} \mathcal{T}_{\pi_{\theta_t}}(Xw_t) - Xw_t \right\| = 0 \quad a.s.,$$

and the actor iterates $\{\theta_t\}$ visit a neighborhood of the stationary points of J_π infinitely often.

2.13 Overloaded Notations and Common Assumptions

In this thesis, we need to learn both state value and action value functions. To ease presentation, we have overloaded several notations. In this section, we summarize the overloaded notations used in the rest of the thesis for clarity and list several commonly used assumptions.

Definition 2.5. *(Notations regarding state value functions)*

- $x : \mathcal{S} \to \mathbb{R}^K$, *state feature function*
- $X \in \mathbb{R}^{|\mathcal{S}|\times K}$, *state feature matrix, whose s-th row is* $x(s)^\top$
- $P_\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$, *state transition matrix under a policy* π, $P_\pi(s,s') \doteq \sum_a \pi(a|s)p(s'|s,a)$
- $d_\pi \in \mathbb{R}^{|\mathcal{S}|}$, *stationary state distribution under a policy* π
- $d_{\pi,\gamma} \in \mathbb{R}^{|\mathcal{S}|}$, *normalized discounted state occupancy measure*
- $d_\mu \in \mathbb{R}^{|\mathcal{S}|}$, *stationary state distribution under a policy* μ
- $D_\pi \doteq diag(d_\pi) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}, D_\mu \doteq diag(d_\mu) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$
- $\Pi_{d,X} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$, *projection onto the column space of a matrix* X *under the norm induced by the vector* d,

  $$\Pi_{d,X} v \doteq X \arg\min_w \|Xw - v\|_d^2 = X(X^\top diag(d) X)^{-1} X^\top diag(d) v.$$

  When it does not confuse, we suppress the subscript X.
- $r_\pi \in \mathbb{R}^{|\mathcal{S}|}$, *reward vector under a policy* π, $r_\pi(s) \doteq \sum_a \pi(a|s) r(s,a)$
- $\mathcal{T}_\pi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$, *state value Bellman operator,* $\mathcal{T}_\pi v \doteq r_\pi + \gamma P_\pi v$

- $i : \mathcal{S} \to (0, +\infty)$, *state interest function*
- $$\begin{aligned} A_{\pi,\mu} \doteq & X^\top D_\mu (\gamma P_\pi - I) X, \\ b_{\pi,\mu} \doteq & X^\top D_\mu r_\pi, \\ C_\mu \doteq & X^\top D_\mu X, \\ \bar{A}_{\pi,\mu} \doteq & X^\top (D_\mu - d_\mu d_\mu^\top)(P_\pi - I) X, \\ \bar{b}_{\pi,\mu} \doteq & X^\top (D_\mu - d_\mu d_\mu^\top) r_\pi \end{aligned}$$

  When it does not confuse, we suppress the subscripts π, μ
- $x_t \doteq x(S_t)$
- $i_t \doteq i(S_t)$

Definition 2.6. *(Notations regarding action value functions)*

- $x : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^K$, *state-action feature function*
- $X \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times K}$, *state-action feature matrix, whose* (s,a)*-th row is* $x(s,a)^\top$
- $P_\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$, *state transition matrix under a policy* π, $P_\pi((s,a),(s',a')) \doteq p(s'|s,a)\pi(a'|s')$
- $d_\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, *stationary state-action distribution under a policy* π
- $d_{\pi,\gamma} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, *normalized discounted state-action pair occupancy measure*
- $d_\mu \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, *stationary state-action distribution under a policy* μ*; an arbitrary state-action distribution with full support in the i.i.d. setting in Definition 2.4*
- $D_\pi \doteq diag(d_\pi) \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}, D_\mu \doteq diag(d_\mu) \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$
- $\Pi_{d,X} : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, *projection onto the column space of a matrix* X *under the norm induced by the vector* d,

  $$\Pi_{d,X} q \doteq X \arg\min_w \|Xw - q\|_d^2 = X(X^\top diag(d) X)^{-1} X^\top diag(d) q.$$

  When it does not confuse, we suppress the subscript X.
- $r \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, *reward vector*

- $\mathcal{T}_\pi : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, *action value Bellman operator,* $\mathcal{T}_\pi q \doteq r + \gamma P_\pi q$
- $i : \mathcal{S} \times \mathcal{A} \to (0, +\infty)$, *state-action interest function*
-

$$\begin{aligned} A_{\pi,\mu} \doteq & X^\top D_\mu (\gamma P_\pi - I) X, \\ b_{\pi,\mu} \doteq & X^\top D_\mu r, \\ C_\mu \doteq & X^\top D_\mu X, \\ \bar{A}_{\pi,\mu} \doteq & X^\top (D_\mu - d_\mu d_\mu^\top)(P_\pi - I) X, \\ \bar{b}_{\pi,\mu} \doteq & X^\top (D_\mu - d_\mu d_\mu^\top) r \end{aligned}$$

  When it does not confuse, we suppress the subscripts π, μ

- $x_t \doteq x(S_t, A_t)$
- $i_t \doteq i(S_t, A_t)$

Remark 1. *When we inevitably need to consider both state and action value functions simultaneously, we add an additional˜ for the notations of action value functions. For example,* $X \in \mathbb{R}^{|\mathcal{S}|\times K}$ *is the state feature matrix and* $\tilde{X} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times K}$ *is the state action feature matrix.*

In the following, we collect several commonly used assumptions in the rest of the thesis for clarity.

Assumption 2.3. *X has linearly independent columns.*

Assumption 2.4. $\{\alpha_t\}$ *is a deterministic positive nonincreasing sequence satisfying* $\sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty$.

Assumption 2.5. $\{\beta_t\}$ *is a deterministic positive nonincreasing sequence satisfying* $\sum_t \beta_t = \infty, \sum_t \beta_t^2 < \infty$.

Assumption 2.6. *There exists some* $d > 0$ *such that* $\sum_t (\beta_t/\alpha_t)^d < \infty$.

The sequences $\{\alpha_t\}$ and $\{\beta_t\}$ in Assumptions 2.4 - 2.6 are typically used as learning rates. Those assumptions can be easily fulfilled. For example, one can consider

$$\alpha_t \doteq \frac{1}{(t+1)^\alpha},$$
$$\beta_t \doteq \frac{1}{(t+1)^\beta},$$

where $\alpha \in (0.5, 1], \beta \in (0.5, 1]$ are some constants satisfying $\alpha < \beta$. Assumptions 2.4 - 2.6 are mainly used to obtain almost sure convergence for proposed algorithms and are common in RL literature (see, e.g., Tsitsiklis and Roy (1996a); Sutton et al. (2008); Yu (2015, 2017)). If constant learning rates are considered, which are more common in empirical study, obtaining almost sure convergence is typically impossible. However, weaker convergence can usually be established, see, e.g., Yu (2015); Zou et al. (2019); Wu et al. (2020). In this thesis, we perform theoretical analysis with decaying learning rates for simplicity but use constant learning rates for experiments to facilitate our empirical investigation. The empirical results with constant learning rates are indicative. If an RL algorithm with a sufficiently small constant learning rate diverges, it is also expected to diverge with decaying learning rates as long as the decaying learning rates make enough updates (i.e., the decaying learning rates sum up to infinity). In this case, using a decaying learning rate can only delay the divergence but cannot make it convergent. This intuition is widely used in the RL community (see, e.g., Chapter 11.2 of Sutton and Barto (2018)). We, however, leave a formal proof of this intuition, as well as empirical investigation with decaying learning rates, for future work.

Regarding the Markovian settings in Definitions 2.1 and 2.2 and the i.i.d. setting in Definition 2.3, we make the following assumptions for state value functions and action value functions respectively.

Assumption 2.7. *The Markov chain in $\mathcal{S}$ induced by $\mu_t \forall t$ or μ is ergodic.*

Assumption 2.8. *The Markov chain in $\mathcal{S}$ induced by $\mu_t \forall t$ or μ is ergodic and $\mu_t(a|s) > 0 \forall (s,a)$ or $\mu(a|s) > 0 \forall (s,a)$.*

The ergodicity in Assumptions 2.7 and 2.8 can sometimes be relaxed to irreducibility. Since we consider a finite state space $\mathcal{S}$, irreducibility immediately implies that the chain is positive recurrent. We are then safe to claim the existence and uniqueness of a stationary distribution, which is sufficient for many results in this work. We choose to assume ergodicity for easing presentation. In this thesis, we consider finite state action

spaces for simplicity. This is indeed restrictive as many real-world problems have continuous and thus infinite state spaces. However, many concepts in this thesis have been extended to Markov chains with general state spaces. For example, Section 3.4 of Meyn and Tweedie (2012) details how the transition probability can be defined on a general state space. Part III of Meyn and Tweedie (2012) details different notions of ergodicity for a Markov chain with a general state space. Working on a general state space is typically considerably harder than working on a finite state space. We believe the results presented in this thesis with a finite state space can be a stepping stone for the more ambitious investigation with a general state space but leave this extension for future work. So far we have discussed ergodicity only for state spaces, which is sufficient when we are concerned only with state-value function. When it comes to state-action value function, we typically need to work on both state and action spaces. In this case, a common practice is to work on a new augmented Markov chain where the new state space is $\mathcal{S} \times \mathcal{A}$. If both $\mathcal{S}$ and $\mathcal{A}$ are finite, then the ergodicity of the original Markov chain with $\mathcal{S}$ as the state space translates easily into the ergodicity of the augmented Markov chain with $\mathcal{S} \times \mathcal{A}$ as the state space. If either $\mathcal{S}$ or $\mathcal{A}$ is infinite, we would need to consider the augmented Markov chain as a general state space Markov chain, which we leave for future work.

Part I

Off-Policy Prediction for Discounted Total Rewards

In this part, we focus on the prediction problem in the discounted setting. In particular, our goal is to estimate either the scalar performance metric J_π or the value functions v_π and q_π, in the context of the deadly triad. For estimating the value functions v_π and q_π, this part discusses three new methods and compares them with GTD and ETD. For estimating the scalar performance metric J_π, we propose in this thesis a new method based on estimating the density ratio. This new method is a side product of a new algorithm originally designed for the average reward setting. We, therefore, defer the full exposition of this algorithm and its comparison with DualDICE to Section 6.

Chapter 3

Prediction with Target Networks

In this chapter, we discuss how target networks can be used as a tool to address the deadly triad issue theoretically.

3.1 Beyond Deep Reinforcement Learning

As discussed in Section 2.11, target networks have enjoyed great success in deep RL as a technique to empirically stabilize the training of deep networks and the use of target networks is mostly limited to deep RL in previous works. Surprisingly, we find target networks are also capable of stabilizing the training of RL algorithms even with linear function approximation.

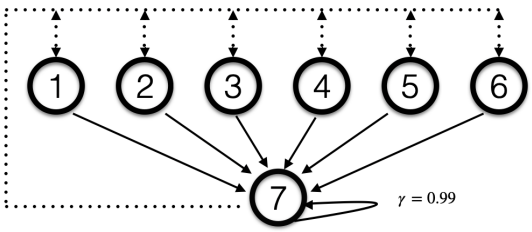

Figure 3.1: Baird's counterexample from Chapter 11.2 of Sutton and Barto (2018). There are two actions available at each state, `dashed` and `solid`. The `solid` action always leads to state 7. The `dashed` action leads to one of states 1 - 6, with equal probability. The discount factor is $\gamma = 0.99$. The reward is always 0. The initial state is sampled uniformly from all the seven states.

We benchmark off-policy linear TD (2.18) and its target network variant in Baird's counterexample (Figure 3.1). We consider linear function approximation for approximating the value function v_π. The feature function is the same as that of Sutton and

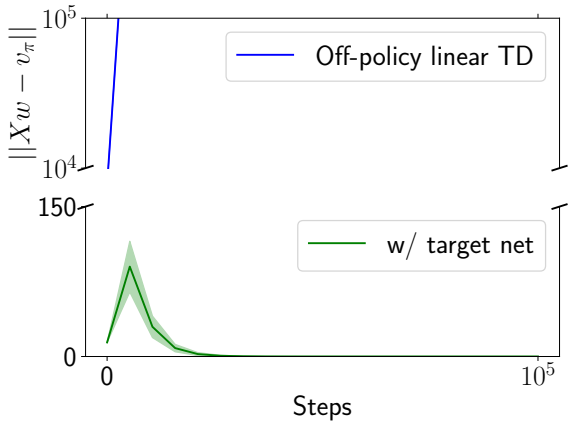

Figure 3.2: Off-policy linear TD and its target network variant in Baird's counterexample

Barto (2018), i.e.,

$$X \doteq \begin{bmatrix} 2I & 0 & \mathbf{1} \\ 0^\top & 1 & 2 \end{bmatrix} \in \mathbb{R}^{7\times 8},$$

where the last row of X corresponds to the feature vector of the seventh state. The initial weight vector w_0 is initialized as $[1, 1, 1, 1, 1, 1, 10, 1]^\top$ as suggested by Sutton and Barto (2018). The behavior policy μ always selects the dashed action w.p. $\frac{6}{7}$ and the solid action w.p. $\frac{1}{7}$. The target policy π always selects the solid action w.p. 1. For off-policy linear TD (2.18), we use a constant learning rate $\alpha_t \equiv 0.01$ as used by Sutton and Barto (2018). As shown by Figure 3.2, off-policy linear TD diverges quickly within the first few steps. It is worth mentioning that as long as the constant learning rate is positive or the summation of a sequence of decreasing learning rates is infinite, off-policy linear TD always diverges in Baird's counterexample eventually. Reducing the learning rates only slows down the divergence. By using a target network, we instead update w recursively as

$$\begin{aligned} w_{t+1} &\leftarrow w_t + \alpha_t \rho_t (R_{t+1} + x_{t+1}^\top \theta_t - x_t^\top w_t) x_t, \\ \theta_{t+1} &\leftarrow \theta_t + \beta_t (w_t - \theta_t), \end{aligned} \tag{3.1}$$

where we set $\theta_0 = w_0, \alpha_t \equiv \beta_t = 0.01$ in our experiments. As shown by Figure 3.2, off-policy linear TD with a target network converges well in Baird's counterexample. This success of a target network in RL with linear function approximation suggests that target networks might not be merely an ad-hoc empirical trick for deep RL. In this chapter, we make contributions towards understanding how and why target networks work theoretically, in the context of the deadly triad.

3.2 Analysis of A Target Network Update

We first propose and analyze a novel target network update rule:

$$\theta_{t+1} \doteq \Gamma_{B_1}\big(\theta_t + \beta_t(\Gamma_{B_2}(w_t) - \theta_t)\big). \tag{3.2}$$

In (3.2), w denotes the main network and θ denotes the target network. $\Gamma_{B_1} : \mathbb{R}^K \to \mathbb{R}^K$ is a projection to the ball $B_1 \doteq \big\{x \in \mathbb{R}^K \mid \|x\| \leq R_{B_1}\big\}$, i.e.,

$$\Gamma_{B_1}(x) \doteq \begin{cases} x, & \|x\| \leq R_{B_1} \\ \frac{R_{B_1}}{\|x\|}x, & \|x\| > R_{B_1} \end{cases}. \tag{3.3}$$

Γ_{B_2} is a projection onto the ball B_2 with a radius R_{B_2}. While (3.2) specifies only how θ is updated, we assume for now w is updated such that w can track θ in the sense that

Assumption 3.1. *There exists $w^* : \mathbb{R}^K \to \mathbb{R}^K$ such that*

$$\lim_{t\to\infty} \|w_t - w^*(\theta_t)\| = 0 \quad a.s..$$

After making some additional assumptions on w^*, we arrive at our general convergent results regarding our new target network updates.

Assumption 3.2. $\sup_\theta \|w^*(\theta)\| < R_{B_2} < R_{B_1} < \infty$.

Assumption 3.3. *w^* is a contraction mapping w.r.t. $\|\cdot\|$.*

Theorem 3.1. *(Convergence of Target Networks) Under Assumptions 2.5, 3.1 - 3.3, the iterate $\{\theta_t\}$ generated by (3.2) satisfies*

$$\lim_{t\to\infty} w_t = \lim_{t\to\infty} \theta_t = \theta^* \quad a.s.,$$

where θ^* is the unique fixed point of $w^*(\cdot)$.

The proof of Theorem 3.1 is provided in Section B.1. Assumptions 3.1 - 3.3 are assumed only for now. Once the concrete update rules for the main network w are specified in the algorithms in later sections, we will prove that those assumptions indeed hold. Assumption 3.1 is expected to hold because we will later require that the target network to be updated much slower than the main network. Consequently, the update of the main network will become a standard least-square regression, whose solution w^* usually exists. Assumption 3.3 is expected to hold because we will later

apply ridge regularization to the least-square regression. Consequently, its solution w^* will not change too fast w.r.t. the change of the regression target.

The target network update (3.2) is the same as that in (3.1) except for the two projections, where the first projection Γ_{B_1} is standard in optimization literature (see, e.g., Nemirovski et al. (2009)) to ensure the iterates are bounded. The second projection Γ_{B_2}, however, appears novel and plays a crucial role in our analysis. *First*, if we have only Γ_{B_1}, the iterate $\{\theta_t\}$ would converge to the invariant set of the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta(t) = w^*(\theta(t)) - \theta(t) + \zeta(t), \tag{3.4}$$

where $\zeta(t)$ is a reflection term that moves $\theta(t)$ back to B_1 when $\theta(t)$ becomes too large. To see this, consider the updates in (3.2) without Γ_{B_2}, i.e.,

$$\theta_{t+1} \doteq \Gamma_{B_1}\big(\theta_t + \beta_t(w_t - \theta_t)\big).$$

If $\theta_t + \beta_t(w_t - \theta_t)$ *was always* inside the ball B_1, then the projection Γ_{B_1} is just an identity mapping and we have $\zeta(t) = 0$, i.e., the corresponding ODE *would* be

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta(t) = w^*(\theta(t)) - \theta(t).$$

However, there is no guarantee that $\theta_t + \beta_t(w_t - \theta_t)$ always lies in the ball B_1. When it is outside the ball B_1, the projection Γ_{B_1} moves it back to the ball B_1 and gets θ_{t+1} via the projection operation. Consequently, there must be an additional term in the ODE corresponding to the projection operator when it is not an identity mapping. We denote this additional term as $\zeta(t)$. Formally speaking, we have

$$\zeta(t) \in -\mathcal{N}_{B_1}(\theta(t)),$$

where $\mathcal{N}_{B_1}(\theta(t))$ denotes the normal cone of B_1 at $\theta(t)$. We refer the reader to Section 4.1 of Yu (2015) and Section 5 of Kushner and Yin (2003) for more details about this reflection term. Due to this reflection term, it is possible that $\theta(t)$ visits the boundary of B_1 infinitely often. It thus becomes unclear what the invariant set of (3.4) is even if w^* is contractive. By introducing the second projection Γ_{B_2} and ensuring $R_{B_1} > R_{B_2}$, we are able to remove the reflection term and show that the iterate $\{\theta_t\}$ tracks the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta(t) = w^*(\theta(t)) - \theta(t),$$

whose invariant set is a singleton $\{\theta^*\}$ when Assumption 3.3 holds. See the proof based on the ODE approach (Kushner and Yin, 2003; Borkar, 2009) for more details. *Second*, to ensure the main network tracks the target network in the sense of Assumption 3.1, it is crucial that the target network changes sufficiently slowly in the following sense:

Lemma 3.2. $\|\theta_{t+1} - \theta_t\| = \mathcal{O}(\beta_t)$.

Lemma 3.2 would not be feasible without the second projection Γ_{B_2} and we defer its proof to Section B.2

In this thesis, we provide several applications of Theorem 3.1 in both discounted and average reward settings, for both prediction and control. We consider a two-timescale framework, where the target network is updated more slowly than the main network. In other words, let $\{\alpha_t\}$ be the learning rates for updating the main network w, we assume Assumption 2.6 holds.

3.3 Expected SARSA for Prediction with A Target Network

We are now ready to present our first successful application of target networks in breaking the deadly triad. In particular, we analyze a variant of off-policy linear expected SARSA for prediction (cf. (2.14)) as an example to demonstrate how a target network addresses the deadly triad theoretically. The analysis presented in this section applies to other prediction algorithms as well (cf. (2.12), (2.13)). We choose off-policy linear expected SARSA for prediction to prepare us for the analysis of control algorithms in later sections.

Consider the Markovian setting in Definition 2.2. Using a target network for bootstrapping in (2.14) and adding linear function approximation yield the following update to the main network w:

$$w_{t+1} \doteq w_t + \alpha_t \left(R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) x(S_{t+1}, a)^\top \theta_t - x_t^\top w_t \right) x_t.$$

Since θ_t is quasi-stationary for w_t in the sense of Lemma 3.2 and Assumption 2.6, this update becomes similar to least squares regression. Motivated by the success of ridge regularization in least squares (Tikhonov et al., 2013) and the widespread use of weight decay in deep RL (see, e.g., Lillicrap et al. (2016)), which is similar to ridge regularization, we add ridge regularization to this least squares, yielding Algorithm 1.

Theorem 3.3. *Under Assumptions 2.3, 2.4, 2.5, 2.6, and 2.8, for any* $\xi \in (0, 1)$, *let*

$$C_0 \doteq \frac{2(1-\xi)\sqrt{\eta}}{\gamma \|P_\pi\|_{d_\mu} \max_{s,a} \sqrt{d_\mu(s,a)}}, C_1 \doteq \frac{\|r\|}{2\xi\sqrt{\eta}} + 1,$$

Algorithm 1: Off-policy linear expected SARSA for prediction with a target network

Initialize $\theta_0 \in B_1$
$S_0 \sim p_0(\cdot)$
$t \leftarrow 0$
while *True* **do**
  $A_t \sim \mu(\cdot|S_t)$
  Execute A_t, get R_{t+1} and S_{t+1}
  $\delta_t \leftarrow R_{t+1} + \gamma \sum_{a'} \pi(a'|S_{t+1}) x(S_{t+1}, a')^\top \theta_t - x_t^\top w_t$
  $w_{t+1} \leftarrow w_t + \alpha_t \delta_t x_t - \alpha_t \eta w_t$
  $\theta_{t+1} \leftarrow \Gamma_{B_1}\left(\theta_t + \beta_t(\Gamma_{B_2}(w_t) - \theta_t)\right)$
  $t \leftarrow t + 1$
end

then for all $\|X\| < C_0, C_1 < R_{B_1}, R_{B_1} - \xi < R_{B_2} < R_{B_1}$ *the iterate* $\{w_t\}$ *generated by Algorithm 1 satisfies*

$$\lim_{t\to\infty} w_t = w_\eta^* \quad a.s.,$$

where w_η^* *is the unique solution to*

$$(A - \eta I)w + b = 0,$$

and

$$\left\|X w_\eta^* - q_\pi\right\| \leq \left(\frac{\sigma_{\max}(X)^2}{\sigma_{\min}(X)^4 \sigma_{\min}(D_\mu)^{2.5}} \|q_\pi\| \eta + \left\|\Pi_{d_\mu} q_\pi - q_\pi\right\|\right) / \xi,$$

where $\sigma_{\max}(\cdot)$ *and* $\sigma_{\min}(\cdot)$ *denote the largest and minimum singular values of a matrix respectively.*

We recall that A and b in Theorem 3.3 are defined in Section 2.13. We defer the proof to Section B.3. Theorem 3.3 requires that the balls for projection are sufficiently large, which is completely feasible in practice. Theorem 3.3 also requires that the feature norm $\|X\|$ is not too large. Similar assumptions on feature norms also appear in Zou et al. (2019); Du et al. (2019); Chen et al. (2019); Carvalho et al. (2020); Wang and Zou (2020); Wu et al. (2020) and can be easily achieved by scaling down the features beforehand.

The solutions to $Aw + b = 0$, if they exist, are TD fixed points for off-policy prediction in the discounted setting (cf. (2.22)), which are also solutions found by GTD. Theorem 3.3 shows that Algorithm 1 finds a regularized TD fixed point w_η^*, which is also the solution of LSTD methods. LSTD maintains estimates for A and

b (referred to as $\hat{A}$ and $\hat{b}$) in an online fashion, which requires $\mathcal{O}(K^2)$ computational and memory complexity per step. As $\hat{A}$ is not guaranteed to be invertible, LSTD usually uses $(\hat{A} - \eta I)^{-1}\hat{b}$ as the solution and η plays a key role in its performance (see, e.g, Chapter 9.8 of Sutton and Barto (2018)). By contrast, Algorithm 1 finds the LSTD solution (i.e., w_η^*) with only $\mathcal{O}(K)$ computational and memory complexity per step. Moreover, Theorem 3.3 provides a performance bound for w_η^*. Let $w_0^* \doteq -A^{-1}b$ denote the TD fixed point, assuming A is nonsingular. Kolter (2011) shows with a counterexample that the approximation error of the TD fixed point (i.e., $\|Xw_0^* - q_\pi\|$) can be arbitrarily large if μ is far from π, as long as there is representation error (i.e., $\left\|\Pi_{d_\mu} q_\pi - q_\pi\right\| > 0$). By contrast, Theorem 3.3 guarantees that $\left\|Xw_\eta^* - q_\pi\right\|$ is bounded from the above, which is one possible advantage of regularized TD fixed points.

In Algorithm 1, both the target network and the ridge regularization are at play. The reader may wonder what if only one of them is in effect. We defer the discussion about this question from a theoretical perspective to Section 9.5 when our analysis of target networks for all settings are ready, though in the next section we shed light on this question empirically.

3.4 Empirical Results

In this section, we empirically investigate the asymptotic and nonasymptotic behavior of using target networks and ridge regularization for prediction.

For the asymptotic behavior, we focus on how η influences the performance of the fixed point w_η^*. To this end, we consider Kolter's example (Kolter, 2011). Kolter's example is a simple two-state Markov Reward Process with $P_\pi \doteq \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$. The reward is set in a way such that the state-value function is $v_\pi = \begin{bmatrix} 1 \\ 1.05 \end{bmatrix}$. The feature matrix is $X \doteq \begin{bmatrix} 1 \\ 1.05 + \epsilon \end{bmatrix}$. Kolter (2011) shows that for any $\epsilon > 0, C_0 > 0$, there exists a $D_\mu = diag\left(\begin{bmatrix} d_\mu(s_1) \\ d_\mu(s_2) \end{bmatrix}\right)$ such that

$$\left\|\Pi_{d_\mu} v_\pi - v_\pi\right\| \leq \epsilon \quad \text{and} \quad \|Xw_0^* - v_\pi\| \geq C_0,$$

where w_0^* is the TD fixed point. This suggests that as long as there is representation error (i.e., $\epsilon > 0$), the performance of the TD fixed point can be arbitrarily poor. We vary the sampling probability of one state ($d_\mu(s_1)$) and compute the corresponding

regularized TD fixed point w_η^* analytically. Figure 3.3 shows that with $\eta = 0$, the performance of w_η^* becomes arbitrarily poor when $d_\mu(s_1)$ approaches around 0.71. With $\eta = 0.01$, the spike exists as well. If we further increase η to 0.02 and 0.03, the performance for w_η^* becomes well bounded. This confirms the potential advantage of the regularized TD fixed points.

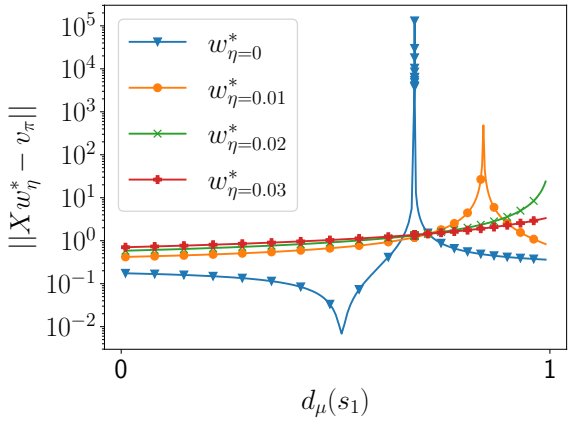

Figure 3.3: Effect of regularization in Kolter's example.

For the nonasymptotic behavior, we use Baird's counterexample to investigate how the ridge regularization affects the training. The curves in Figure 3.4 marked as "standard" correspond to the canonical off-policy linear TD with ridge regularization (cf. setting $\theta_t \equiv w_t$ in Algorithm 1); the curves marked as "ours" correspond to the canonical off-policy linear TD with both ridge regularization and a target network (cf. Algorithm 1 without projection). All curves are averaged over 30 independent runs with shaded regions indicating one standard derivation. Figure 3.4 shows that even with $\eta = 0$, i.e., no ridge regularization, our algorithms with target network still converge in the tested domains. By contrast, without a target network, even when mild regularization is imposed, standard off-policy algorithms still diverge. This empirically confirms the importance of the target network.

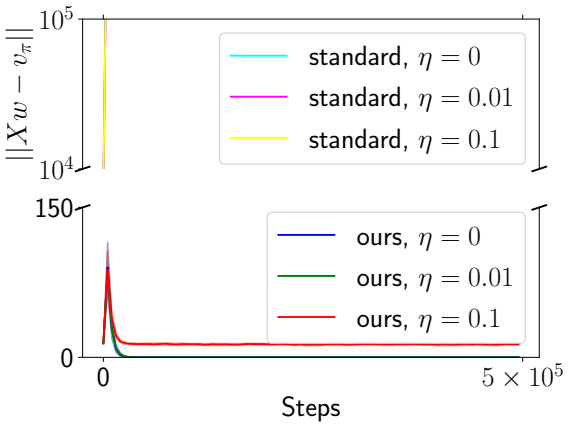

Figure 3.4: Effect of regularization in Baird's counterexample for prediction

Chapter 4

Prediction with Learned Emphasis

In this chapter, we discuss one approach for addressing the large variance of ETD, one classical method to address the deadly triad for prediction. Our new approach is based on learning the emphasis with a secondary function approximator. Inspired by this new approach, we also develop the reverse RL framework for representing retrospective knowledge.

4.1 The Curse of Variance

Though ETD is proven to be convergent, the followon trace F_t usually has a large, possibly infinite, variance, making it hard to use directly. Sutton et al. (2016) provide a concrete example where the variance of the followon trace is infinite. Sutton and Barto (2018) also report that in Baird's counterexample, a commonly used benchmark for testing off-policy RL algorithms, "*it is nigh impossible to get consistent results in computational experiments*" (Chapter 11.9 of Sutton and Barto 2018) for ETD.

There are several attempts to address this variance. Hallak et al. (2016) propose to replace F_t with $F_{t,\beta}$, which is computed recursively as

$$F_{t,\beta} \doteq i(S_t) + \beta\rho_{t-1}F_{t-1,\beta}, \tag{4.1}$$

where $\beta \in (0, 1)$ is an additional hyperparameter. The resulting ETD$(0, \beta)$ then updates $\{w_t\}$ iteratively as

$$w_{t+1} \doteq w_t + \alpha_t F_{t,\beta}\rho_t(R_{t+1} + \gamma x_{t+1}^\top w_t - x_t^\top w_t)x_t. \tag{4.2}$$

Theorem 1 of Hallak et al. (2016) states that there exists a problem-dependent constant β_{upper} such that $\beta \leq \beta_{\text{upper}}$ implies that the variance of $F_{t,\beta}$ is bounded. Further, Proposition 1 of Hallak et al. (2016) states that there exists a problem-dependent

constant β_{lower} such that $\beta \geq \beta_{\text{lower}}$ implies that the expected update corresponding to (4.2) is contractive, which plays a key role in bounding the performance of the fixed point of (4.2), assuming (4.2) converges. Unfortunately, there is no guarantee that $\beta_{\text{lower}} \leq \beta_{\text{upper}}$ always holds, i.e, the desired β that is both sufficiently small and sufficiently large does not always exist. Jiang et al. (2021) propose to clip the importance sampling ratio ρ_t when computing the followon trace F_t to reduce variance. However, nothing can be said about the convergence of the resulting algorithm due to the bias introduced by clipping. Despite these attempts, it remains an open problem to reduce the variance of emphatic TD methods introduced by the followon trace in a theoretically grounded way. In this chapter, we present a new algorithm that partially addresses this variance.

4.2 Gradient Emphasis Learning

In ETD (2.26), we use the followon trace F_t for reweighting the update, which suffers from a large variance. It is, however, the emphasis $m_{\pi,\mu}$, that contributes directly to the negative definiteness of the corresponding A matrix in (2.28). It is, therefore, desirable to update $\{w_t\}$ directly with $m_{\pi,\mu}$ as

$$w_{t+1} \doteq w_t + \alpha_t m_{\pi,\mu}(S_t)\rho_t(R_{t+1} + \gamma x_{t+1}^\top w_t - x_t^\top w_t)x_t. \tag{4.3}$$

Since the emphasis $m_{\pi,\mu}$ is unknown, we propose to use a secondary function approximator to learn an estimate, based on the following important observation. For a vector $y \in \mathbb{R}^{|\mathcal{S}|}$, we define an operator $\hat{\mathcal{T}}$ as

$$\hat{\mathcal{T}}_{\pi,\mu} y \doteq i + \gamma D_\mu^{-1} P_\pi^\top D_\mu y. \tag{4.4}$$

Proposition 4.1. *$\hat{\mathcal{T}}_{\pi,\mu}$ is a contraction mapping w.r.t. some weighted maximum norm and $m_{\pi,\mu}$ is its unique fixed point.*

The proof provided in Section B.4 involves arguments from Bertsekas and Tsitsiklis (2015), where the choice of the weighted maximum norm depends on $\gamma D_\mu^{-1} P_\pi^\top D_\mu$.

Given Proposition 4.1 and the recursive definition of F_t in (2.25), it is tempting to compose a semi-gradient update rule for learning $m_{\pi,\mu}$ analogously to TD but in a backward direction. To be more specific, let $X\nu$ be our estimate for $m_{\pi,\mu}$, where $\nu \in \mathbb{R}^K$ is the learnable weight vector. Consider the Markovian setting in Definition 2.2, one can update ν iteratively as

$$\nu_{t+1} \doteq \nu_t + \alpha_t(i_{t+1} + \gamma\rho_t x_t^\top \nu_t - x_{t+1}^\top \nu_t)x_{t+1}. \tag{4.5}$$

This semi-gradient update (4.5), however, can possibly diverge for the same reason as the divergence of off-policy linear TD. To see this, we can compute the corresponding A matrix of (4.5) as

$$\begin{aligned}
A \doteq & \mathbb{E}_\mu \left[x_{t+1} (\gamma \rho_t x_t - x_{t+1})^\top \right] \\
= & \sum_{s,a,s'} d_\mu(s) \mu(a|s) p(s'|s,a) x(s') \left(\gamma \frac{\pi(a|s)}{\mu(a|s)} x(s) - x(s') \right)^\top \\
= & \sum_{s,a,s'} d_\mu(s) \pi(a|s) p(s'|s,a) \gamma x(s') x(s)^\top - \sum_{s'} d_\mu(s') x(s') x(s')^\top \\
= & \gamma (P_\pi X)^\top D_\mu X - X^\top D_\mu X \\
= & X^\top (\gamma P_\pi^\top - I) D_\mu X.
\end{aligned}$$

This A matrix is simply the transpose of the A matrix of (2.18). Since both A matrices are not guaranteed to be negative definite, neither (4.5) nor (2.18) is guaranteed to be convergent.

Motivated by the success of GTD methods, we instead aim to find an ν such that

$$X\nu = \Pi_{d_\mu} \hat{\mathcal{T}}_{\pi,\mu} X\nu$$

via minimizing the following objective:

$$L(\nu) \doteq \left\| \Pi_{d_\mu} \hat{\mathcal{T}}_{\pi,\mu}(X\nu) - X\nu \right\|_{d_\mu}^2. \tag{4.6}$$

This objective is akin to the MSPBE objective in (2.22) but is induced by the new operator $\hat{\mathcal{T}}_{\pi,\mu}$ instead of the Bellman operator $\mathcal{T}_\pi$.

Similar to (2.23), we can expand $L(\nu)$ as

$$\begin{aligned}
L(\nu) = & \left\| X^\top D_\mu \left(\hat{\mathcal{T}}_{\pi,\mu}(X\nu) - X\nu \right) \right\|_{C^{-1}}^2 \\
= & \max_\kappa 2\kappa^\top X^\top D_\mu \left(\hat{\mathcal{T}}_{\pi,\mu}(X\nu) - X\nu \right) - \kappa^\top C \kappa.
\end{aligned}$$

Then the optimization problem we seek to solve becomes

$$\min_\nu \max_\kappa L(\nu, \kappa),$$

where

$$\begin{aligned}
L(\nu,\kappa) \doteq & 2\kappa^\top X^\top D_\mu \left(\hat{\mathcal{T}}_{\pi,\mu}(X\nu) - X\nu \right) - \kappa^\top C \kappa \\
= & 2\kappa^\top X^\top D_\mu \left(i + \gamma D_\mu^{-1} P_\pi^\top D_\mu X\nu - X\nu \right) - \kappa^\top C \kappa.
\end{aligned}$$

Since $L(\nu, \kappa)$ is convex in ν and concave in κ, we can similarly use primal-dual methods to solve this CCSP problem, i.e., we update ν and κ following $-\nabla_\nu L(\nu, \kappa)$ and $\nabla_\kappa L(\nu, \kappa)$:

$$\begin{aligned}
\nabla_\nu L(\nu, \kappa) =& 2\kappa^\top X^\top D_\mu \left(\gamma D_\mu^{-1} P_\pi^\top D_\mu X\nu - X\nu\right) \\
=& 2\kappa^\top X^\top (\gamma P_\pi^\top - I) D_\mu X \\
=& 2\mathbb{E}_\mu \left[\kappa^\top x_{t+1} (\gamma \rho_t x_t - x_{t+1})^\top\right], \\
\nabla_\kappa L(\nu, \kappa) =& 2\left(i + \gamma D_\mu^{-1} P_\pi^\top D_\mu X\nu - X\nu\right)^\top D_\mu X - 2\kappa^\top X^\top D_\mu X \\
=& 2i^\top D_\mu X + 2\nu^\top X^\top D_\mu (\gamma P_\pi - I) X - 2\kappa^\top X^\top D_\mu X \\
=& 2\mathbb{E}_\mu \left[\left(i_{t+1} + \gamma \rho_t x_t^\top \nu - x_{t+1}^\top \nu\right) x_{t+1} - \left(x_{t+1}^\top \kappa\right) x_{t+1}\right]^\top.
\end{aligned}$$

We call the resulting algorithm Gradient Emphasis Learning (GEM). Algorithm 2 is an instance of GEM in the i.i.d. setting in Definition 2.3. GEM can of course be used in the Markovian setting in Definitions 2.1 and 2.2, which we discuss later in Chapter 12. The following theorem confirms the convergence of GEM.

Algorithm 2: Gradient Emphasis Learning

$k \leftarrow 0$
while *True* **do**
  Sample $S_k \sim d_\mu(\cdot), A_k \sim \mu(\cdot|S_k), R_k \doteq r(S_k, A_k), S'_k \sim p(\cdot|S_k, A_k)$
  $x_k \leftarrow x(S_k), x'_k \leftarrow x(S'_k), \rho_k \leftarrow \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}, i'_k \leftarrow i(S'_k)$
  $\delta_k \leftarrow i'_k + \gamma \rho_k x_k^\top \nu_k - x_k'^\top \nu_k$
  $\kappa_{k+1} \leftarrow \kappa_k + \alpha_k \left(\delta_k - x_k'^\top \kappa_k\right) x'_k$
  $\nu_{k+1} \leftarrow \nu_k + \alpha_k (x'_k - \gamma \rho_k x_k) x_k'^\top \kappa_k$
  $k \leftarrow k + 1$
end

Theorem 4.2. *Let Assumptions 2.3 and 2.4 hold. Assume $X^\top(\gamma P_\pi^\top - I) D_\mu X$ is nonsingular. Then the iterates $\{\nu_k\}$ generated by Algorithm 2 satisfy*

$$\lim_{k\to\infty} \nu_k = \nu_* \quad a.s.,$$

where ν_* is the unique solution to $L(\nu) = 0$.

The convergence proof of Theorem 4.2 is similar to that of GTD2 in Sutton et al. (2009) and is provided in Section B.5. We now study the performance of ν_* in terms of approximating $m_{\pi,\mu}$. Since ν_* is the minimizer of $L(\nu)$, which shares a similar spirit to the off-policy MSPBE objective (2.22), ν_* also suffers from a similar problem of

the minimizer of the off-policy MSPBE objective, as documented in Kolter (2011). Namely, we are able to bound $\|X\nu_* - m_{\pi,\mu}\|_{d_\mu}$ only when the behavior policy is not too different from the target policy in the following sense:

Assumption 4.1. *The matrix*

$$\begin{bmatrix} C & X^\top P_\pi^\top D_\mu X \\ X^\top D_\mu P_\pi X & C \end{bmatrix}$$

is positive semidefinite.

This assumption is from Kolter (2011) and prescribes that μ is not too far from π.

Proposition 4.3. *Let Assumptions 2.3 and 4.1 hold. Assume $X^\top(\gamma P_\pi^\top - I)D_\mu X$ is nonsingular. Then*

$$\|X\nu_* - m_{\pi,\mu}\|_{d_\mu} = \mathcal{O}\left(\left\|\Pi_{d_\mu} m_{\pi,\mu} - m_{\pi,\mu}\right\|_{d_\mu}\right).$$

The proof of Proposition 4.3 is provided in Section B.6.

We are now able to replace the emphasis $m_{\pi,\mu}(S_t)$ in (4.3) with $x(S_t)^\top \nu_*$. Since the estimate ν_* is learned with GEM without eligibility trace, we refer to the resulting algorithm for prediction as GEM-ETD(0) (Algorithm 3). The following theorem confirms the convergence of GEM-ETD(0) provided that the estimate ν_* is good enough.

Algorithm 3: GEM-ETD(0)

$S_0 \sim p_0(\cdot)$
$t \leftarrow 0$
while *True* **do**
  $A_t \sim \mu(\cdot|S_t)$
  Execute A_t, get R_{t+1} and S_{t+1}
  $\rho_t \leftarrow \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}, x_t \leftarrow x(S_t), x_{t+1} \leftarrow x(S_{t+1})$
  $\delta_t \leftarrow R_{t+1} + \gamma x_{t+1}^\top w_t - x_t^\top w_t$
  $w_{t+1} \leftarrow w_t + \alpha_t (x_t^\top \nu_*) \rho_t \delta_t x_t$
  $t \leftarrow t + 1$
end

Theorem 4.4. *Let Assumptions 2.3, 2.4 and 2.7 hold. Then there exists a constant $\epsilon > 0$ such that*

$$\max_s \left| x(s)^\top \nu_* - m_{\pi,\mu}(s) \right| < \epsilon$$

implies that the iterates $\{w_t\}$ *generated by Algorithm 3 satisfy*

$$\lim_{t\to\infty} w_t = w_\infty,$$

where

$$w_\infty \doteq -\left(X^\top D_\mu D_{\nu_*}(\gamma P_\pi - I)X\right)^{-1} X^\top D_\mu D_{\nu_*} r_\pi,$$
$$D_{\nu_*} \doteq diag(X\nu_*).$$

Further,

$$\|Xw_\infty - v_\pi\| = \mathcal{O}(\epsilon) + \mathcal{O}\left(\left\|\Pi_{f_{\pi,\mu}} v_\pi - v_\pi\right\|_{f_{\pi,\mu}}\right).$$

The proof of Theorem 4.4 is provided in Section B.7. Importantly, when the estimation of the emphasis is good enough, Theorem 4.4 also provides a bound on the performance of its solution w_∞. It is worth mentioning that the convergence of GEM-ETD(0) is based on the good quality of the GEM solution ν_*. As shown by Proposition 4.3, a performance bound of the GEM solution is, however, only available when μ is not too far away from π, which is the major limit of GEM-ETD(0).

4.3 Empirical Results

In this section, we empirically investigate how well GEM approximates the emphasis and how well GEM-ETD(0) approximates the value function.

We still consider Baird's counterexample in Figure 3.1. But this time we test four different sets of features: original features, one-hot features, zero-hot features, and aliased features. Original features are the features used by Sutton and Barto (2018), which are documented in Section 3.1. This set of features is, however, uncommon as in practice the number of states is usually much larger than the number of features. One-hot features use one-hot encoding, where each feature lies in $\mathbb{R}^7$, which indeed degenerates to a tabular setting. Zero-hot features are the complements of one-hot features, e.g., the feature of the state 1 is $[0, 1, 1, 1, 1, 1, 1]^\top \in \mathbb{R}^7$. The quantities of interest, e.g., m_π and v_π, can be expressed accurately under all three sets of features. In the fourth set of features, we consider state aliasing. In Baird's counterexample, the states 1-6 are equivalent. We therefore alias the state 7 to the state 6. Namely, we still consider the original features, but now the feature of the state 7 is modified to be identical as the feature of the state 6. The last two dimensions of features then become identical for all states, and therefore we removed them, resulting in features lying in $\mathbb{R}^6$. Now the quantities of interest may not lie in the feature space.

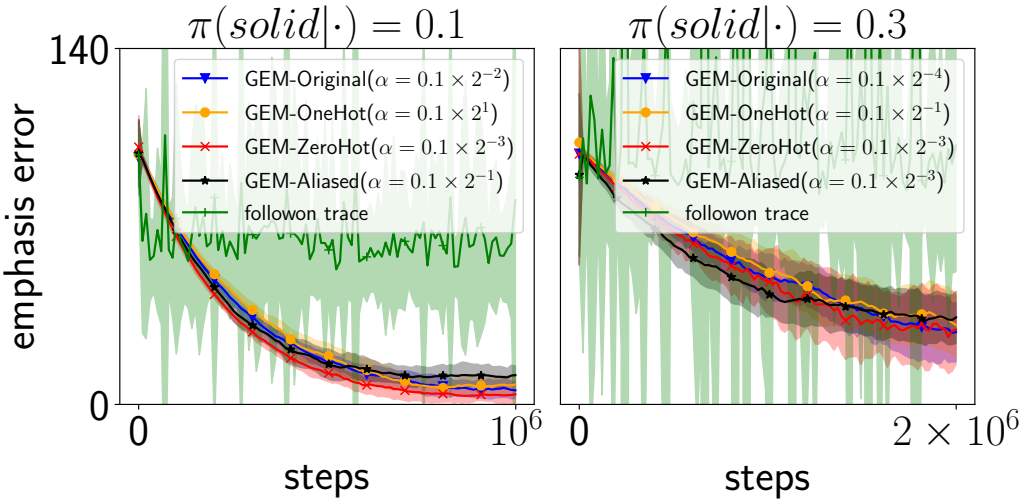

Figure 4.1: Averaged emphasis approximation error in last 1000 steps for the followon trace and GEM with different features. Learning rates used are bracketed.

In this chapter, we propose to approximate the emphasis with GEM while ETD essentially approximates the emphasis with the followon trace F_t directly. We first compare those two approaches. At time step t, the emphasis approximation error is computed as $|F_t - m_{\pi,\mu}(S_t)|$ and $|\nu_t^\top x(S_t) - m_{\pi,\mu}(S_t)|$ for the followon trace and GEM respectively, where $m_{\pi,\mu}$ is computed analytically, $F_{-1} = 0$, and ν_0 is drawn from a unit normal distribution. For GEM, we consider a fixed learning rate α and tune it from $\{0.1 \times 2^1, \ldots, 0.1 \times 2^{-6}\}$. We consider two target policies: $\pi(\texttt{solid}|\cdot) = 0.1$ and $\pi(\texttt{solid}|\cdot) = 0.3$.

As shown in Figure 4.1, the GEM approximation enjoys lower variance than the followon trace approximation and has a lower approximation error under all four sets of features.

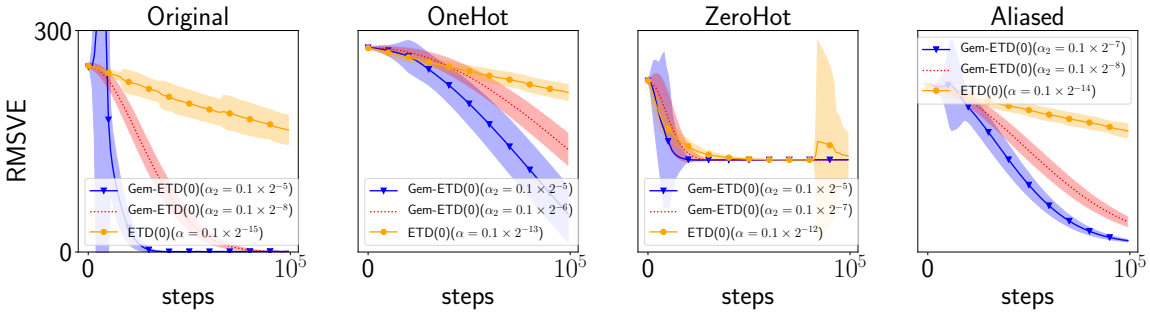

Figure 4.2: Averaged RMSVE in recent 1000 steps for GEM-ETD(0) and ETD(0) with four different sets of features.

We now compare ETD and GEM-ETD(0). To make it fair comparison, we run

GEM and GEM-ETD(0) simultaneously. In other words, in GEM-ETD(0), instead of using ν_*, which is not available during training, we use ν_t, the weight vector of GEM at the current step.

We consider a target policy $\pi(\texttt{solid}|\cdot) = 0.05$. We report the Root Mean Squared Value Error (*RMSVE*) at each time step during training in Figure 4.2. RMSVE is computed as $\|Xw_t - v_\pi\|_{d_\mu}$, where v_π is computed analytically. We use constant learning rates. For ETD, we tune the learning rate α from $\{0.1 \times 2^0, \ldots, 0.1 \times 2^{-19}\}$. For GEM-ETD(0), we set $\alpha_1 = 0.025$ (the learning rate for updating $\{\nu_t\}$) and tune α_2 (the learning rate for updating $\{w_t\}$) in the same range as α. For both algorithms, we report the results with learning rates that minimized the area under the curve (AUC) in the solid lines in Figure 4.2. In our policy evaluation experiments, GEM-ETD(0) has a clear win over ETD under all four sets of features. Note the AUC-minimizing learning rate for ETD is usually several orders smaller than that of GEM-ETD(0), which explains why ETD curves tend to have smaller variance than GEM-ETD(0) curves. When we decrease the learning rate of GEM-ETD(0) (as indicated by the red dashed lines in Figure 4.2), the variance of GEM-ETD(0) can be reduced, and the AUC is still smaller than that of ETD.

GEM-ETD(0) is indeed a way to trade off bias and variance. If the state features are heavily aliased, the GEM emphasis estimation may be heavily biased, as will GEM-ETD(0). We do not claim that GEM-ETD(0) is always better than ETD. For example, when we set the target policy to $\pi(\texttt{solid}|\cdot) = 1$, there was no observable progress for both GEM-ETD(0) and ETD with reasonable computation resources.[1] When it comes to the bias-variance trade-off, the optimal choice is usually task-dependent and our empirical results suggest that GEM-ETD(0) is a promising approach for this trade-off.

4.4 Beyond Emphasis: Reverse Reinforcement Learning

In the on-policy setting, the followon trace F_t can be expanded as

$$F_t = i_t + \gamma i_{t-1} + \gamma^2 i_{t-2} + \ldots.$$

[1]This target policy is problematic for GEM-ETD(0) mainly because the magnitude of δ_t in Algorithm 2 varies dramatically across different states, which makes the supervised learning of κ hard.

Recall that the return G_t is defined as

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots.$$

The return accumulates rewards in the future while the followon trace accumulates interests in the past. The expectation of the return is the value function, which is a fundamental quantity in RL. The conditional expectation of the followon trace is the emphasis, which is less explored compared with the value function. Since there is no fundamental difference between the interest and the reward, one natural question arises: can we accumulate rewards in the past? In this section, we describe the reverse RL framework that focuses on past rewards and show that reverse RL is useful for representing *retrospective knowledge*.

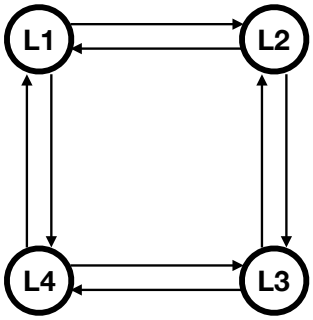

Figure 4.3: A microdrone doing random walk among four different locations. `L4` is a charging station where the microdrone's battery is fully recharged.

Much knowledge can be formulated as answers to predictive questions (Sutton, 2009), for example, "to know that Joe is in the coffee room is to predict that you will see him if you went there" (Sutton, 2009). Such knowledge is referred to as *predictive knowledge* (Sutton, 2009; Sutton et al., 2011). General Value Functions (GVFs, Sutton et al. 2011) are commonly used to represent predictive knowledge. GVFs are essentially the same as canonical value functions (Puterman, 2014; Sutton and Barto, 2018). However, the policy, the reward function, and the discount function associated with GVFs are usually carefully designed such that the numerical value of a GVF at certain state matches the numerical answer to certain predictive question. In this way, GVFs can represent predictive knowledge.

Consider the concrete example in Figure 4.3, where a microdrone is doing a random walk. The microdrone is initialized somewhere with 100% battery. `L4` is a power station where its battery is recharged to 100%. Each clockwise movement consumes 2% of the battery, and each counterclockwise movement consumes 1% (for simplicity,

we assume negative battery levels, e.g., -10%, are legal). Furthermore, each movement fails with probability 1%, in which case the microdrone remains in the same location and no energy is consumed. An example of a predictive question in this system is:

Question 1. *Starting from* `L1`*, how much energy will be consumed in expectation before the next charge?*

To answer this question, we can model the system as an MDP. The policy is uniformly random and the reward for each movement is the additive inverse of the corresponding battery consumption. Whenever the microdrone reaches state `L4`, the episode terminates. Under this setup, the answer to Question 1 is the expected cumulative reward when starting from `L1`, i.e., the state value of `L1`. Hence, GVFs can represent the predictive knowledge in Question 1. As a GVF is essentially a value function, it can be trained with any data stream from agent-environment interaction via RL, demonstrating the generality of the GVF approach. Importantly, the most appealing feature of GVFs is their compatibility with off-policy learning, making this representation of predictive knowledge scalable and efficient. For example, in the Horde architecture (Sutton et al., 2011), many GVFs are learned in parallel with GTD methods. In the microdrone example, we can learn the answer to Question 1 under many different conditions (e.g., when the charging station is located at `L2` or when the microdrone moves clockwise with probability 80%) simultaneously with off-policy learning by considering different reward functions, discount functions, and polices.

GVFs, however, cannot answer many other useful questions, e.g., if at some time t, we find the microdrone at `L1`, how much battery do we expect it to have? As such questions emphasize the influence of possible past events on the present, we refer to their answers as *retrospective knowledge*. Such retrospective knowledge is useful, for example, in anomaly detection. Suppose the microdrone runs for several weeks by itself while we are traveling. When we return at time t, we find the microdrone is at `L1`. We can then examine the battery level and see if it is similar to the expected battery at `L1`. If there is a large difference, it is likely that there is something wrong with the microdrone. There are, of course, many methods to perform such anomaly detection. For example, we could store the full running log of the microdrone during our travel and examine it when we are back. The memory requirement to store the full log, however, increases according to the length of our travel. By contrast, if we have retrospective knowledge, i.e., the expected battery level at each location, we can program the microdrone to log its battery level at each step (overwriting the record from the previous step). We can then examine the battery level when we are

back and see if it matches our expectation. The current battery level can be easily computed via the previous battery level and the energy consumed at the last step, using only constant computation per step. The storage of the battery level requires only constant memory as we do not need to store the full history, which would not be feasible for a microdrone. Thus retrospective knowledge provides a memory-efficient way to perform anomaly detection. Of course, this approach may have lower accuracy than storing the full running log. This is indeed a trade-off between accuracy and memory, and we expect applications of this approach in memory-constrained scenarios such as embedded systems.

To know the expected battery level at `L1` at time t is essentially to answer the following question:

Question 2. *How much energy do we expect the microdrone to have consumed since the last time it had 100% battery given that it is at `L1` at time t?*

Unfortunately, GVFs cannot represent retrospective knowledge (e.g., the answer to Question 2) easily. GVFs provide a mechanism to ignore all future events after reaching certain states via setting the discount function at those states to be 0. This mechanism is useful for representing predictive knowledge. For example, in Question 1, we do not care about events *after* the next charge. For retrospective knowledge, we, however, need a mechanism to ignore all previous events before reaching certain states. For example, in Question 2, we do not care about events *before* the last time the microdrone had 100% battery. Unfortunately, GVFs do not have such a mechanism.

In this section, we propose *Reverse GVFs* to represent retrospective knowledge. Using the same MDP formulation of the microdrone system, let the random variable $\tilde{G}_t$ denote the energy the microdrone has consumed at time t since the last time it had 100% battery. To answer Question 2, we are interested in the conditional expectation of $\tilde{G}_t$ given that $S_t = \texttt{L1}$. We refer to functions describing such conditional expectations as Reverse GVFs, which we propose to learn via *Reverse Reinforcement Learning*. The key idea of Reverse RL is still bootstrapping, but in the reverse direction. It is easy to see that $\tilde{G}_t$ depends on $\tilde{G}_{t-1}$ and the energy consumption from $t-1$ to t. In general, the quantity of interest at time t depends on that at time $t-1$ in Reverse RL.

Inspired by the return G_t, we define the reverse return $\tilde{G}_t$, which accumulates previous rewards:

$$\tilde{G}_t \doteq R_t + \gamma(S_{t-1})\tilde{G}_{t-1}, \quad \tilde{G}_0 \doteq 0.$$

Here we have considered a state-dependent discount factor $\gamma : \mathcal{S} \to [0, 1]$ following Sutton et al. (2011). In the reverse return $\tilde{G}_t$, the discount function γ has different semantics than in the return G_t. Namely, in G_t, the discount function down-weights future rewards, while in $\tilde{G}_t$, the discount function down-weights past rewards. In an extreme case, setting $\gamma(S_{t-1}) = 0$ allows us to ignore all the rewards before time t when computing the reverse return $\tilde{G}_t$, which is exactly the mechanism we need to represent retrospective knowledge.

Let us consider the microdrone example again (Figure 4.3) and try to answer Question 2. Assume the microdrone was initialized at `L3` at $t = 0$ and visited `L4` and `L1` afterwards. Then it is easy to see that $\tilde{G}_2$ is exactly the energy the microdrone has consumed since its last charge. In general, if we find the microdrone at `L1` at time t, the expectation of the energy that the microdrone has consumed since its last charge is exactly $\mathbb{E}[\tilde{G}_t | S_t = \texttt{L1}, \pi, p, r]$. Note the answer to Question 2 is not homogeneous in t. For example, suppose the microdrone is initialized at `L4` at $t = 0$. If we find it at `L1` at $t = 1$, it is trivial to see the microdrone has consumed 2% battery. By contrast, if we find it at `L1` at $t = 100$, computing the energy consumption since the last time it had 100% battery is nontrivial. It is inconvenient that the answer depends the time step t but fortunately, we can show the following:

Assumption 4.2. *The chain induced by π is ergodic and $(I - P_\pi^\top \Gamma)^{-1}$ exists, where $\Gamma \doteq diag(\gamma)$.*

Theorem 4.5. *Under Assumption 4.2, the limit $\lim_{t\to\infty} \mathbb{E}[\tilde{G}_t | S_t = s, \pi, p, r]$ exists, which we refer to as $\tilde{v}_\pi(s)$. Furthermore, we define the reverse Bellman operator $\hat{\mathcal{T}}_\pi$ as*

$$\hat{\mathcal{T}}_\pi y \doteq D_\pi^{-1} \tilde{P}_\pi^\top \tilde{D}_\pi r + D_\pi^{-1} P_\pi^\top \Gamma D_\pi y,$$

where $D_\pi \doteq diag(d_\pi) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ with d_π being the stationary distribution of the chain induced by π, $\tilde{P}_\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}|}$ is the transition matrix, i.e., $\tilde{P}_\pi((s, a), s') \doteq p(s'|s, a)$, and $\tilde{D}_\pi \doteq diag(\tilde{d}_\pi) \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ with $\tilde{d}_\pi(s, a) \doteq d_\pi(s)\pi(a|s)$. Then $\hat{\mathcal{T}}_\pi$ is a contraction mapping w.r.t. some weighted maximum norm, and $\tilde{v}_\pi$ is its unique fixed point. We have $\tilde{v}_\pi = D_\pi^{-1}(I - P_\pi^\top \Gamma)^{-1} \tilde{P}_\pi^\top \tilde{D}_\pi r$.

The proof of Theorem 4.5 is provided in Section B.8. Assumption 4.2 can be fulfilled in the real world as long as the problem we consider has a recurring structure. Theorem 4.5 states that the numerical value of $\tilde{v}_\pi(\texttt{L1})$ approximately answers Question 2.

When Question 2 is asked for a large enough t, the error in the answer $\tilde{v}_\pi(\texttt{L1})$ is arbitrarily small. We call $\tilde{v}_\pi(s)$ a *Reverse General Value Function*, which approximately encodes the retrospective knowledge, i.e., the answer to the retrospective question induced by π, r, γ, t and s.

Canonical RL algorithms for learning the value function v_π can be easily adapted to learn the reverse value function $\tilde{v}_\pi$. For example, let us consider an on-policy setting and use Xw as our estimate for $\tilde{v}_\pi$. Reverse TD updates w iteratively as

$$w_{t+1} \doteq w_t + \alpha_t(R_t + \gamma(S_{t-1})x_{t-1}^\top w_t - x_t^\top w_t)x_t. \tag{4.7}$$

Proposition 4.6. *Let Assumptions 2.3 and 4.2 hold. Then the iterates* $\{w_t\}$ *generated by* (4.7) *satisfy*

$$\lim_{t\to\infty} w_t = -\tilde{A}^{-1}\tilde{b} \quad a.s.,$$

where

$$\begin{aligned}\tilde{A} &\doteq X^\top(P_\pi^\top \Gamma - I)D_\pi X,\\ \tilde{b} &\doteq X^\top \tilde{P}_\pi^\top \tilde{D}_\pi r.\end{aligned}$$

The proof of Proposition 4.6 is provided in Section B.9. Similar to TD(λ) in Sutton (1988), we can also extend Reverse TD to Reverse TD(λ), which updates w iteratively as

$$w_{t+1} \doteq w_t + \alpha_t\Big(R_t + \gamma(S_{t-1})\big((1-\lambda)x_{t-1}^\top w_t + \lambda \bar{G}_{t-1}\big) - x_t^\top w_t\Big)x_t.$$

With $\lambda = 1$, Reverse TD(λ) reduces to supervised learning. There are also off-policy reverse RL algorithms and distributional reverse RL algorithms, which, however, deviate from the main topic of this thesis and we refer the reader to Zhang et al. (2020e) for more details.

Besides Reverse RL, there are other approaches we could consider for answering Question 2. For example, we could formalize it as a simple regression task, where the input is the location and the target is the power consumption since the last time the microdrone had 100% battery. This regression formulation is a special case of Reverse RL, similar to how Monte Carlo is a special case of TD(λ). Alternatively, answering Question 2 is trivial if we have formulated the system as a partially observable MDP. We could use either the location or the battery level as the state and the other as the observation. In either case, however, deriving the conditional observation probabilities is nontrivial. We could also model the system as a reversed chain directly as Morimura

et al. (2010) in light of reverse bootstrapping. This, however, creates difficulties in off-policy learning. First, assume the initial distribution p_0 is the same as the stationary distribution d_π. We can then compute the posterior action distribution given the next state and the posterior state distribution given the action and the next state using Bayes' rule as

$$\begin{aligned}\Pr(a|s') =& \frac{\sum_s d_\pi(s)\pi(a|s)p(s'|s,a)}{d_\pi(s')},\\ \Pr(s|s',a) =& \frac{d_\pi(s)\pi(a|s)p(s'|s,a)}{d_\pi(s')}.\end{aligned}$$

We can then define a new MDP with the same state space $\mathcal{S}$ and the same action space $\mathcal{A}$. But the new policy is the posterior distribution $\Pr(a|s')$ and the new transition kernel is the posterior distribution $\Pr(s|s',a)$. Intuitively, this new MDP flows in the reverse direction of the original MDP. Samples from the original MDP can also be interpreted as samples from the new MDP. Assuming we have a trajectory $\{S_0, A_0, S_1, A_1, \ldots, S_k\}$ from the original MDP following π, we can interpret the trajectory $\{S_k, A_{k-1}, \ldots, A_0, S_0\}$ as a trajectory from the new MDP, allowing us to work on the new MDP directly. For example, applying TD in the new MDP is equivalent to applying the Reverse TD in the original MDP. As TD should converge in this new MDP as it would on any MDP, we can expect the convergence of Reverse TD in the original MDP. However, in the new MDP, we no longer have access to the policy, i.e., we cannot compute $\Pr(a|s')$ explicitly as it requires both d_π and p, to which we do not have access. This is acceptable in the on-policy setting but renders the off-policy setting infeasible, as we do not know the target policy at all. We, therefore, argue that working on the reversed chain directly is only feasible for on-policy learning.

Overall, we argue that reverse RL is a scalable and efficient framework for representing retrospective knowledge. We refer the reader to Zhang et al. (2020e) for more empirical results.

Chapter 5

Prediction with Truncated Followon Traces

In Chapter 4, we introduce a secondary function approximator for learning the emphasis to address the large variance of the followon trace. The quality of the emphasis approximation, however, heavily depends on the quality of the features thus convergence is not always guaranteed. In this chapter, we propose a new method for reducing the variance of the followon trace without introducing a secondary function approximator.

5.1 Less Is More

The original followon trace F_t in (2.25) can be expanded as

$$\begin{aligned} F_t &= i_t + \gamma\rho_{t-1}F_{t-1} \\ &= i_t + \gamma\rho_{t-1}i_{t-1} + \gamma^2\rho_{t-1}\rho_{t-2}F_{t-2} \\ &= i_t + \gamma\rho_{t-1}i_{t-1} + \gamma^2\rho_{t-1}\rho_{t-2}i_{t-2} + \gamma^3\rho_{t-1}\rho_{t-2}\rho_{t-3}F_{t-3} \\ &= \dots \\ &= \sum_{j=0}^{t} \gamma^j \rho_{t-j:t-1} i_{t-j}, \end{aligned} \tag{5.1}$$

where

$$\rho_{j:k} \doteq \begin{cases} \rho_j\rho_{j+1}\cdots\rho_k & j \le k \\ 1 & j > k \end{cases}$$

is shorthand for the product of importance sampling ratios. Clearly, F_t depends on all the history from time steps 0 to t. The followon trace F_t has a large variance because the product of the importance sampling ratios $\rho_{t-j:t-1}$ has a large variance,

especially for earlier steps (i.e., when j is large). However, the motivation of the introduction of the followon trace F_t is to ensure the corresponding A matrix in ETD to be negative definite, which depends on only the expectation of F_t, not the variance of F_t (cf. (2.28)). It can be easily seen that the expectation of the product of the importance sampling ratios is well bounded. Consequently, the term $\gamma^j \mathbb{E}\left[\rho_{t-j:t-1} i_{t-j}\right]$ is negligible when computing $\mathbb{E}\left[F_t\right]$ if j is large. *Earlier steps contribute little to the expectation but are the major source of the large variance.* One straightforward idea to reduce the variance of the followon trace is then to truncate its computation, perhaps up to a window of size n.

The idea of truncated followon traces, introduced in Yu (2012, 2015, 2017), is, for a fixed length n, to compute the followon trace F_t as if F_{t-n-1} was 0. More specifically, let $F_{t,n}$ be the truncated followon traces of length n; we have

$$F_{t,n} \doteq \begin{cases} \sum_{j=0}^{n} \gamma^j \rho_{t-j:t-1} i_{t-j} & t \geq n \\ F_t & t < n \end{cases}. \tag{5.2}$$

For example, if $n = 2$, we then compute $F_{t,2}$ for any t as

$$F_{t,2} = i_t + \gamma \rho_{t-1} i_{t-1} + \gamma^2 \rho_{t-1} \rho_{t-2} i_{t-2}.$$

It is worth mentioning that Yu (2012, 2015, 2017) introduces the truncated traces as an intermediate mathematical tool in proofs to understand the asymptotic behavior of some LSTD methods (e.g., off-policy LSTD(λ) in Yu 2012, emphatic LSTD(λ) in Yu 2015) and gradient TD methods (e.g., GTD(λ) in Sutton et al. 2009) for prediction. In this thesis, we instead use truncated followon traces *algorithmically* as a tool for variance reduction for both prediction (in this section) and control (in Section 11). Moreover, Yu (2015) shows only asymptotic convergence of the original ETD, while we provide both asymptotic and nonasymptotic convergence analysis for our proposed algorithms.

5.2 Truncated Emphatic Temporal Difference Learning

In this section, we propose to replace F_t with $F_{t,n}$ in ETD(0). Apparently, for a fixed n, the variance of $F_{t,n}$ is guaranteed to be bounded. By contrast, Sutton et al. (2016) show that the variance of F_t can be infinite. The resulting algorithm, truncated emphatic TD, is given in Algorithm 4, where we adopt the convention that $i_t = \rho_t = 0$ for any $t < 0$.

Algorithm 4: Truncated emphatic TD

$S_0 \sim p_0(\cdot)$
$t \leftarrow 0$
while *True* **do**
  Sample $A_t \sim \mu(\cdot|S_t)$
  Execute A_t, get R_{t+1}, S_{t+1}
  $\rho_t \leftarrow \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$
  $F_{t,n} \leftarrow 0$
  for $k = 0, \dots, n$ **do**
    $F_{t,n} \leftarrow i_{t-n+k} + \gamma \rho_{t-n+k-1} F_{t,n}$
  end
  $w_{t+1} \leftarrow w_t + \alpha_t F_{t,n} \rho_t (R_{t+1} + \gamma x_{t+1}^\top w_t - x_t^\top w_t) x_t$
  $t \leftarrow t + 1$
end

To compute $F_{t,n}$, one needs to store $2n$ extra scalars: $\rho_{t-1}, \dots, \rho_{t-n}, i_{t-1}, \dots, i_{t-n}$. Such memory overhead is inevitable even for naive on-policy multi-step TD methods (Chapter 7.1 of Sutton and Barto 2018). The computation of $F_{t,n}$ can indeed be done incrementally at the cost of maintaining one more extra scalar:

$$\Delta_t \doteq \frac{\rho_t i_t}{\rho_{t-n-1} i_{t-n-1}} \Delta_{t-1},$$
$$F_{t,n} \doteq i_t + \gamma \rho_{t-1} F_{t-1,n} - \Delta_t.$$

Overall, we argue that compared with ETD in Sutton et al. (2016), the additional memory and computational cost of truncated emphatic TD is negligible. We now proceed to analyzing truncated emphatic TD.

When analyzing the original ETD, we have to consider the chain $\{(F_t, S_t, A_t)\}$ evolving in the space $\mathbb{R} \times \mathcal{S} \times \mathcal{A}$ (see, e.g., Yu 2015). The space $\mathbb{R}$ is not even countable, making it hard to analyze the chain $\{(F_t, S_t, A_t)\}$ even with Assumption 2.8. With the truncated followon trace $F_{t,n}$, we only need to consider the chain $\{(S_{t-n}, A_{t-n}, \dots, S_t, A_t)\}$ which evolves in a *finite* space $(\mathcal{S} \times \mathcal{A})^n$. The ergodicity of this chain follows immediately from Assumption 2.8. Once the ergodicity is established, we can analyze the limiting update matrix under the corresponding stationary distribution.

The additional hyperparameter n in (5.2) defines a hard truncation. By contrast, the additional hyperparameter β in (4.1) defines a soft truncation. As discussed in Section 4.1, a desired β does not always exist since we require β to be both sufficiently large and sufficiently small. By contrast, we will show soon that a desired n always

exists because we only require n to be sufficiently large. Further, to analyze ETD$(0,\beta)$ with the soft truncation, we still need to work on the chain $\{(F_{t,\beta}, S_t, A_t)\}$, whose behavior is hard to analyze. Consequently, though the asymptotic convergence of ETD$(0,\beta)$ in prediction may be established similarly to Yu (2015) for certain β, so far no finite sample analysis is available for ETD$(0,\beta)$ in prediction. Nevertheless, we believe the soft truncation and the hard truncation are two different directions for variance reduction. The soft truncation is analogous to computing the return G_t with a discount factor different from γ (see, e.g., Romoff et al. 2019); the hard truncation is analogous to computing the return G_t with a fixed horizon (see, e.g., Asis et al. 2020). It is straightforward to combine the two techniques together. For example, we can consider $F_{t,\beta,n}$ defined as

$$F_{t,\beta,n} \doteq \begin{cases} \sum_{j=0}^{n} \beta^j \rho_{t-j:t-1} i_{t-j} & t \geq n \\ F_{t,\beta} & t < n \end{cases}.$$

This combination, however, deviates from the main purpose of this section and is saved for future work.

We now study the truncated trace $F_{t,n}$. Similar to (2.27), we study the limit of the conditional expectation of the truncated followon trace and define

$$m_{\pi,\mu,n}(s) \doteq \lim_{t\to\infty} \mathbb{E}\left[F_{t,n} | S_t = s\right].$$

We refer to m_n as *truncated emphasis* for a finite n.

Lemma 5.1. *Let Assumption 2.8 hold. Then*

$$m_{\pi,\mu,n} = \sum_{j=0}^{n} \gamma^j D_\mu^{-1} (P_\pi^\top)^j D_\mu i,$$
$$\lim_{n\to\infty} m_{\pi,\mu,n} = D_\mu^{-1}(I - \gamma P_\pi^\top)^{-1} D_\mu i.$$

The proof of Lemma 5.1 is provided in Section B.10. By definition, the weighting vector $f_{\pi,\mu}$ in (2.29) involved in the A matrix of the ETD update (2.26) satisfies $f_{\pi,\mu} = D_\mu m_{\pi,\mu}$. Similarly, we define

$$f_{\pi,\mu,n} \doteq D_\mu m_{\pi,\mu,n}. \tag{5.3}$$

Lemma 5.2. *Let Assumption 2.8 hold. Then*

$$\|m_{\pi,\mu,n} - m_{\pi,\mu}\|_1 \leq \gamma^{n+1} \frac{d_{\mu,max}}{d_{\mu,min}} \|m_{\pi,\mu}\|_1,$$
$$\|f_{\pi,\mu,n} - f_{\pi,\mu}\|_\infty \leq \gamma^{n+1} \frac{d_{\mu,max}^2}{d_{\mu,min}} \|m_{\pi,\mu}\|_1,$$

where $d_{\mu,max} \doteq \max_s d_\mu(s)$ *and* $d_{\mu,min} \doteq \min_s d_\mu(s)$.

The proof of Lemma 5.2 is provided in Section B.11. The A matrix of the ETD update (2.26) is $X^\top D_{f_{\pi,\mu}}(\gamma P_\pi - I)X$. Similarly, it can be shown that the A matrix of truncated emphatic TD (Algorithm 4) is

$$X^\top D_{f_{\pi,\mu,n}}(\gamma P_\pi - I)X,$$

where $D_{f_{\pi,\mu,n}} \doteq diag(f_{\pi,\mu,n})$. Lemma 5.2 asserts that $f_{\pi,\mu,n}$ approaches $f_{\pi,\mu}$ geometrically fast. Consequently, we can expect $X^\top D_{f_{\pi,\mu,n}}(\gamma P_\pi - I)X$ to be n.d. if n is not too small.

Lemma 5.3. *Let Assumptions 2.3 and 2.8 hold. If*

$$\gamma^{n+1} < \frac{\lambda_{min} d_{\mu,min}}{d_{\mu,max}^2 \|\gamma P_\pi - I\| \|m_{\pi,\mu}\|_1}, \tag{5.4}$$

then $X^\top D_{f_{\pi,\mu,n}}(\gamma P_\pi - I)X$ is n.d., where λ_{min} is the minimum eigenvalue of

$$\frac{1}{2}\left(D_{f_{\pi,\mu}}(I - \gamma P_\pi) + (I - \gamma P_\pi^\top) D_{f_{\pi,\mu}}\right).$$

Sutton et al. (2016) prove that $\lambda_{min} > 0$. The proof of Lemma 5.3 is provided in Section B.12. Since the LHS of (5.4) diminishes geometrically as n increases, we argue that in practice we do not need a very large n. Recall that the motivation of using the followon trace F_t is to ensure the limiting update matrix to be n.d. Lemma 5.3 shows that to ensure this negative definiteness, we do not need to use all history to compute F_t. *Earlier steps contribute little to this negative definiteness due to discounting but introduce large variance due to the products of importance sampling ratios.* As suggested by (5.4), the desired value of n depends on the magnitude of the emphasis $m_{\pi,\mu}$, which is determined together by the behavior policy μ, the target policy π, the structure of the MDP, and the magnitude of the interest i. In general, when the magnitude of the emphasis increases, the desired truncation length also increases. In practice, we propose to treat the truncation length n as an additional hyperparameter, as estimating the desired n without access to the transition kernel p can be very challenging, which we leave for future work.

We can now show the asymptotic convergence of truncated emphatic TD using the standard ODE-based approach.

Theorem 5.4. *Let the assumptions and conditions of Lemma 5.3 hold. Let Assumption 2.4 hold. Then the iterates $\{w_t\}$ generated by truncated empathic TD (Algorithm 4) satisfy*

$$\lim_{t\to\infty} w_t = w_{*,n} \quad a.s., \text{ where}$$
$$w_{*,n} \doteq -A_n^{-1} b_n,\ A_n \doteq X^\top D_{f_{\pi,\mu,n}}(\gamma P_\pi - I)X,\ b_n \doteq X^\top D_{f_{\pi,\mu,n}} r_\pi.$$

The proof of Theorem 5.4 is provided in Section B.13, which, after the negative definiteness of A_n is established with Lemma 5.3, follows the same routine as the convergence proof of on-policy TD in Proposition 6.4 of Bertsekas and Tsitsiklis (1996).

We now give a finite sample analysis of projected truncated emphatic TD (Algorithm 5). Algorithm 5 is different from Algorithm 4 in that it adopts an additional projection Π_R when updating the weight w_t. Here Π_R denotes the projection onto the ball of a radius R centered at the origin w.r.t. ℓ_2 norm. Introducing such a projection is common practice in finite sample analysis of TD methods (Bhandari et al., 2018; Zou et al., 2019). This projection is mainly used to control the errors introduced by Markovian samples. If i.i.d. samples are used instead, such projection can indeed be eliminated (Bhandari et al., 2018; Dalal et al., 2018).

Algorithm 5: Projected Truncated Emphatic TD

$S_0 \sim p_0(\cdot)$
$t \leftarrow 0$
while *True* **do**
  Sample $A_t \sim \mu(\cdot|S_t)$
  Execute A_t, get R_{t+1}, S_{t+1}
  $\rho_t \leftarrow \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$
  $F_{t,n} \leftarrow 0$
  for $k = 0, \dots, n$ **do**
    $F_{t,n} \leftarrow i_{t-n+k} + \gamma \rho_{t-n+k-1} F_{t,n}$
  end
  $w_{t+1} \leftarrow \Pi_R \left(w_t + \alpha_t F_{t,n} \rho_t (R_{t+1} + \gamma x_{t+1}^\top w_t - x_t^\top w_t) x_t \right)$
  $t \leftarrow t + 1$
end

Theorem 5.5. *Let the assumptions and conditions of Lemma 5.3 hold. Let* $R \geq \|w_{*,n}\|$. *With proper learning rates* $\{\alpha_t\}$, *for sufficiently large* t,

$$\mathbb{E}\left[\|w_t - w_{*,n}\|^2\right] = \mathcal{O}\left(\frac{\ln^3 t}{t}\right).$$

The proof of Theorem 5.5 is omitted to avoid verbatim repetition since it is just a special case of a more general result in the control setting (Theorem 11.5). The conditions on learning rates and the constants hidden by $\mathcal{O}(\cdot)$ are also similar to those of Theorem 11.5. We now analyze the performance of $w_{*,n}$.

Lemma 5.6. *Let* $\kappa \doteq \min_s \frac{d_\mu(s)i(s)}{f_{\pi,\mu}(s)}$. *Let Assumptions 2.3 and 2.8 hold. If*

$$\gamma^{n+1} < \frac{\kappa d_{\mu,min} \min_s i(s) d_\mu(s)}{d_{\mu,max}^2 \|I - \gamma P_\pi^\top\|_\infty \|m_{\pi,\mu}\|_1}, \tag{5.5}$$

then $\Pi_{f_{\pi,\mu,n}} \mathcal{T}_\pi$ *is a* $\sqrt{\gamma}$*-contraction in* $\|\cdot\|_{f_{\pi,\mu,n}}$ *and*

$$\|X w_{*,n} - v_\pi\|_{f_{\pi,\mu,n}} \leq \frac{1}{\sqrt{1-\gamma}} \left\|\Pi_{f_{\pi,\mu,n}} v_\pi - v_\pi\right\|_{f_{\pi,\mu,n}}. \tag{5.6}$$

The proof of Lemma 5.6 is similar to Hallak et al. (2016) and is provided in Section B.14. Again, the LHS of (5.5) diminishes geometrically. So in practice, n might not need to be too large.

Lemma 5.6 characterizes the performance of the fixed point of truncated ETD methods in prediction settings. In particular, the $\left\|\Pi_{f_{\pi,\mu,n}} v_\pi - v_\pi\right\|_{f_{\pi,\mu,n}}$ term in (5.6) is the representation error resulting from the limit of the capacity of the linear function approximator. With different truncation length, we use different norm (i.e., $\|\cdot\|_{f_{\pi,\mu,n}}$) to measure the representation error. The multiplicative factor $\frac{1}{\sqrt{1-\gamma}}$, however, does not depend on n. In other words, as long as n is sufficiently large in the sense of (5.5), the exact value of n, including $n = \infty$ (i.e., no truncation), does not seem to affect the performance of the fixed point much. The intuition is straightforward. Comparing (5.2) and (5.1), it is easy to see that by using the truncation, we discard the term $\sum_{j=n+1}^{t} \gamma^j \rho_{t-j:t-1} i_{t-j}$ corresponding to earlier transitions from steps 0 to $t-n-1$. This term has a large, possibly infinite, variance because of the product of importance sampling ratios. The expectation of this term is, however, negligible because the expectation of the importance sampling ratios are well bounded (see the proof of Lemma 5.2) and the multiplicative factor γ^j is negligible. It is the expectation, not the variance, of the trace that determines the performance of the corresponding fixed point. Consequently, the truncation proposed in this work does not seem to yield a compromise in the performance of the fixed point. Different truncation lengths (including $n = \infty$, i.e., no truncation) lead to different fixed points. We, however, argue that those fixed points are in general equally good since their performances can all be bounded by the representation error and the truncation length has little effect on $\|\cdot\|_{f_{\pi,\mu,n}}$ when n is large (see Lemma 5.2). By contrast, the performance of the fixed points of GTD methods minimizing d_μ-induced MSPBE can be arbitrarily worse, no matter how small the representation error is (Kolter, 2011).

	$n=\infty$	$n=0$	$n=2$	$n=4$	$n=8$	$\beta=0.8$
$\pi(dashed=0\|s)$	-	-	-	-	-	-
$\pi(dashed=0.02\|s)$	-	-	-	10^4	10^{14}	-
$\pi(dashed=0.04\|s)$	10^7	-	10^1	10^1	10^9	10^9
$\pi(dashed=0.06\|s)$	-	-	10^2	10^0	10^4	10^4
$\pi(dashed=0.08\|s)$	-	-	10^{-1}	10^0	10^7	10^7
$\pi(dashed=0.1\|s)$	-	-	10^{-11}	10^0	10^2	10^4

Table 5.1: Average variance of curves in Figure 5.1. Each curve in Figure 5.1 consists of 100 data points. The average variance of those data points is reported in this table. Here we consider only successful configurations whose averaged prediction error at the end of training is smaller than 5. The average variance of other curves are not included and denoted as "-".

5.3 Empirical Results

In this section, we empirically compare truncated emphatic TD (Algorithm 4), ETD, and ETD(0, β).

We use Baird's counterexample as the benchmark (Figure 3.1) and consider a behavior policy $\mu(\texttt{solid}|s) = \frac{1}{7}$ and $\mu(\texttt{dashed}|s) = \frac{6}{7}$, which is the same as the behavior policy used in Sutton and Barto (2018). We consider different target policies from $\pi(\texttt{dashed}|s) = 0$ to $\pi(\texttt{dashed}|s) = 0.1$. We benchmark Algorithm 4 with different selection of n. When $n = \infty$, Algorithm 4 reduces to the original ETD. When $n = 0$, Algorithm 4 reduces to the naive off-policy TD. We use a fixed learning rate α, which is tuned from $\Lambda_\alpha \doteq \{0.1 \times 2^0, 0.1 \times 2^{-1}, \ldots, 0.1 \times 2^{-19}\}$ for each n, with 30 independent runs. We report learning curves with the learning rate minimizing the value prediction error at the end of training. Additionally, we also benchmark ETD$(0, \beta)$, where we replace the $F_{t,n}$ in Algorithm 4 with the trace $F_{t,\beta}$ computed via (4.1). We tune β in $\{0.1, 0.2, 0.4, 0.8\}$. For each β, we tune the learning rate α in Λ_α as before. The interest is 1 for all states (i.e., $i(s) \equiv 1 \, \forall s$). We report the learning curves with the best β. All curves are averaged over 30 independent runs with shaded regions indicating standard errors, unless otherwise specified.

As shown by Figures 5.1 with $n = 0$, the naive off-policy TD makes no progress in this prediction setting. The curve is almost flat because the best learning rate is 0.1×2^{-19}; using any larger learning rate simply accelerates divergence. As shown by the curves with $n = \infty$, naive ETD does make some progresses when $\pi(\texttt{dashed}|s) > 0$, though the final prediction errors at the end of training are usually large. By contrast, using $n = 4$ leads to quick convergence in all the tasks with $\pi(\texttt{dashed}|s) > 0$.

Reducing n from 4 to 2 also works when $\pi(\texttt{dashed}|s) \geq 0.04$ and increasing n from 4 to 8 significantly increases the variance. Obviously increasing n leads to a larger variance, so in practice we want to find the smallest n. Moreover, though ETD$(0, \beta)$ converges when $\pi(\texttt{dashed}|s) \geq 0.04$, it usually exhibits larger variance than our truncated ETD with $n = 2$ or $n = 4$ (Table 5.1). We conjecture that this is because the trace (4.1) still relies on all the history. Consider, e.g., $\pi(\texttt{dashed}|s) = 0.02$: the maximum importance ratio is $\rho_{max} = 0.98 \times 7 = 6.86$. If $\beta\rho_{max} > 1$, there is still a chance that the trace in (4.1) goes to infinity since it depends on all the history. However, requiring $\beta\rho_{max} < 1$ would require using a small β, which itself could also lead to instability. By contrast, with truncation, $F_{t,n}$ is always guaranteed to be bounded. The results suggest that our hard truncation also has empirical advantages over the soft truncation in Hallak et al. (2016), besides the theoretical advantages of enabling finite sample analysis for both prediction and control settings. It can be analytically computed that for all $\pi(\texttt{dashed}|s) \in \{0, 0.02, 0.04, 0.06, 0.08, 0.1\}$, the desired n as suggested by Lemma 5.3 is around 700. The n we use in the experiments is much smaller than the suggested one. This is because Lemma 5.3 has to be conservative enough to cope well with all possible MDPs. In this chapter, we focus on establishing the existence of such an n and giving an initial but possibly loose bound. We leave the improvement of Lemma 5.3 for future work. For computational experiments, we recommend to treat n as an additional hyperparameter.

When $\pi(\texttt{dashed}|s) = 0$, which is used in the original Baird's counterexample, no selection of n or β is able to make any progress. The failure of ETD with this target policy is also observed by Sutton and Barto (2018). This target policy is particularly challenging because its off-policyness is the largest in all the tested target policies, making it hard to observe progress in computational experiments. Though truncation is not guaranteed to always reduce the variance to desired levels while maintaining convergence, our experiments in the prediction setting do suggest it is a promising approach. We leave a more in-depth investigation of this target policy for future work.

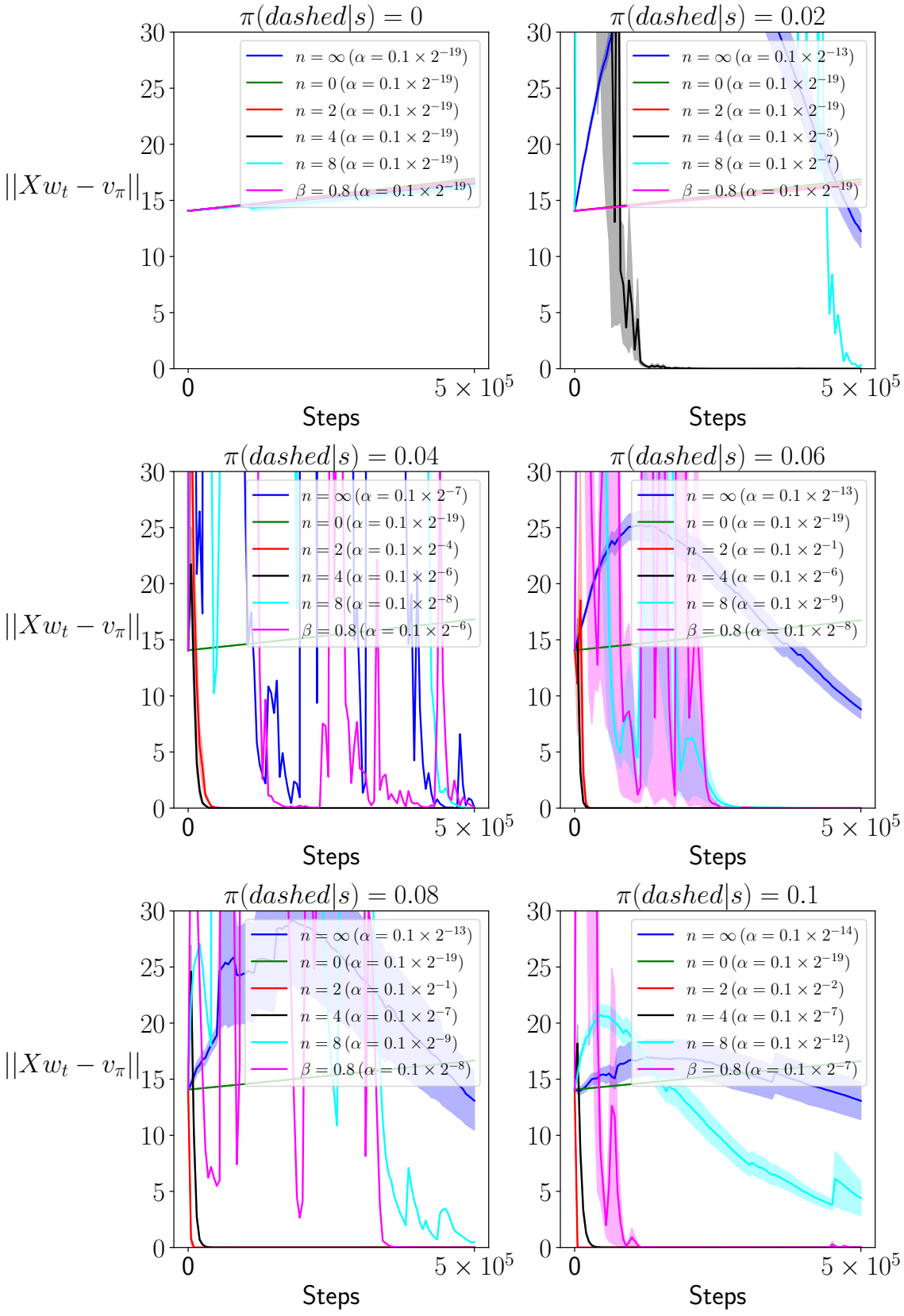

Figure 5.1: Truncated emphatic TD and ETD(0, β) in the prediction setting.

Part II

Off-Policy Prediction for Average Reward

In this part, we focus on the prediction problem in the average reward setting. In particular, our goal is to estimate either the scalar performance metric $\bar{J}_\pi$ or the differential value functions $\bar{v}_\pi$ and $\bar{q}_\pi$, in the context of the deadly triad. For estimating the average reward $\bar{J}_\pi$, we describe a new method based on learning the density ratio. Though this new method is designed for the average reward setting, it also works in the discounted setting as a special case. For estimating the differential value functions, we describe two new methods. The first one using target networks is a natural extension of the methods in Section 3. The second one is an extension of GTD methods to the average reward setting.

Chapter 6

Prediction with Density Ratios

In this chapter, we propose a new method for estimating the average reward $\bar{J}_\pi$ based on learning the density ratio. Though it is designed primarily for the average reward setting, it also works for the discounted setting as a special case.

6.1 Less Is More

As discussed in Section 2.10, when our goal for prediction is to estimate the scalar performance metric J_π or $\bar{J}_\pi$, the density ratio

$$\tau_\gamma(s,a) = \frac{d_{\pi,\gamma}(s,a)}{d_\mu(s,a)}.$$

is a useful quantity. For estimating J_π, we consider $\gamma < 1$; for estimating $\bar{J}_\pi$, we consider $\gamma = 1$. In either case, τ_γ is a solution to the linear system

$$D_\mu \tau = (1-\gamma) d_{p_0\pi} + \gamma P_\pi^\top D_\mu \tau, \tag{6.1}$$

where $\tau \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ is the free variable. Solving (6.1) is, therefore, a natural approach to finding τ_γ. For the discounted setting with $\gamma < 1$, it is easy to see that τ_γ is the *unique* solution to (6.1). For the average reward setting with $\gamma = 1$, (6.1) degenerates to

$$D_\mu \tau = P_\pi^\top D_\mu \tau,$$

which, unfortunately, has infinitely many solutions and τ_γ is merely one of them. It is thus desirable to add additional constraints to ensure that τ_γ is the unique solution to (6.1) even with $\gamma = 1$. There are two natural constraints. First, τ_γ is nonnegative. Correspondingly, we have the constraint

$$\tau(s,a) \geq 0, \forall s, a. \tag{6.2}$$

Second, τ_γ is properly normalized. Correspondingly, we have the constraint

$$d_\mu^\top \tau = 1. \tag{6.3}$$

It is easy to see that for any $\gamma \in [0, 1]$, τ_γ is the only τ that satisfies (6.1), (6.2), and (6.3) simultaneously. Existing works such as Zhang et al. (2020a); Mousavi et al. (2020), therefore, consider both (6.2) and (6.3) when solving (6.1). To fulfill (6.2), positive nonlinearity such as $\exp(\cdot), (\cdot)^2$ is applied in function approximation (Zhang et al., 2020a; Mousavi et al., 2020). To fulfill (6.3), self-normalization is applied in function approximation (Mousavi et al., 2020). Those techniques, however, make the problem of solving (6.1) nonconvex. Consequently, even with tabular representation, there is no convergence guarantee for the algorithms proposed by Zhang et al. (2020a); Mousavi et al. (2020), much less convergence to τ_γ. In this chapter, we instead make the following important observations:

Lemma 6.1. *For any $\gamma \in [0, 1]$, τ_γ is the unique τ that satisfies (6.1) and (6.3) simultaneously.*

The proof of Lemma 6.1 is provided in Section B.15. In other words, to ensure the uniqueness of the solution to (6.1), adding (6.3) is already sufficient. We argue that considering both constraints unnecessarily introduces additional obstacles in optimization. In this thesis, we, therefore, design our algorithm based on only (6.1) and (6.3).

6.2 Gradient Stationary Distribution Correction Estimation

Consider the i.i.d. setting in Definition 2.4. To solve (6.1) subject to (6.3), we consider the following optimization problem:

$$\min_{\tau \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}} L(\tau) \doteq \frac{1}{2}\left\|(1-\gamma)d_{p_0\pi} + \gamma P_\pi^\top D_\mu \tau - D_\mu \tau\right\|_{D_\mu^{-1}}^2 + \frac{\lambda}{2}(d_\mu^\top \tau - 1)^2, \tag{6.4}$$

where $\lambda > 0$ is a constant. Readers may find the first term of this objective similar to the MSBE. However, while in MSBE the norm is induced by D_μ, we consider a norm induced by D_μ^{-1}. This norm is carefully designed and provides expectations that we can sample from, which will be clear once $L(\tau)$ is expanded (see (6.5) below). Remarkably, we have:

Theorem 6.2. *τ is optimal for (6.4) i.f.f. $\tau = \tau_\gamma$.*

Theorem 6.2 follows immediately from Lemma 6.1. We now expand $L(\tau)$. With

$$\delta \doteq (1-\gamma)d_{p_0\pi} + \gamma P_\pi^\top D_\mu \tau - D_\mu \tau,$$

we have

$$\begin{aligned} L(\tau) &\doteq \frac{1}{2}\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[\left(\frac{\delta(s,a)}{d_\mu(s,a)}\right)^2\right] + \frac{\lambda}{2}(d_\mu^\top \tau - 1)^2 \\ &= \max_{f\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}} \mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[\frac{\delta(s,a)}{d_\mu(s,a)}f(s,a) - \frac{1}{2}f(s,a)^2\right] \\ &\quad + \lambda \max_{\eta\in\mathbb{R}}\left(\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[\eta\tau(s,a) - \eta\right] - \frac{\eta^2}{2}\right), \end{aligned} \tag{6.5}$$

where the equality comes also from the Fenchel's conjugate and the interchangeability principle as in Zhang et al. (2020a) and DualDICE (cf. (2.31)). We, therefore, consider the following problem

$$\min_{\tau\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}} \max_{f\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|},\eta\in\mathbb{R}} L(\tau,\eta,f), \tag{6.6}$$

where

$$\begin{aligned} &L(\tau,\eta,f) \\ \doteq &\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[\frac{\delta(s,a)}{d_\mu(s,a)}f(s,a) - \frac{1}{2}f(s,a)^2\right] + \lambda\left(\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[\eta\tau(s,a)-\eta\right] - \frac{\eta^2}{2}\right). \end{aligned}$$

We expand the first term as

$$\begin{aligned} &\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[\frac{\delta(s,a)}{d_\mu(s,a)}f(s,a) - \frac{1}{2}f(s,a)^2\right] \\ =&\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[\frac{(1-\gamma)d_{p_0\pi}(s,a) + \gamma\left(P_\pi^\top D_\mu\tau\right)(s,a) - (D_\mu\tau)(s,a)}{d_\mu(s,a)}f(s,a) - \frac{1}{2}f(s,a)^2\right] \\ =&(1-\gamma)\mathbb{E}_{(s,a)\sim d_{p_0\pi}(\cdot)}\left[f(s,a)\right] + \gamma\sum_{s,a}\left(P_\pi^\top D_\mu\tau\right)(s,a)f(s,a) \\ &- \mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[\tau(s,a)f(s,a)\right] - \frac{1}{2}\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[f(s,a)^2\right] \\ =&(1-\gamma)\mathbb{E}_{(s,a)\sim d_{p_0\pi}(\cdot)}\left[f(s,a)\right] + \gamma\mathbb{E}_{(s,a,s',a')\sim d_{\mu p\pi}(\cdot)}\left[\tau(s,a)f(s',a')\right] \\ &- \mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[\tau(s,a)f(s,a)\right] - \frac{1}{2}\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[f(s,a)^2\right]. \end{aligned}$$

Consequently, we have

$$
\begin{aligned}
& L(\tau, \eta, f) \\
=& (1-\gamma)\mathbb{E}_{(s,a)\sim d_{p_0\pi}(\cdot)}\left[f(s,a)\right] + \gamma\mathbb{E}_{(s,a,s',a')\sim d_{\mu p\pi}(\cdot)}\left[\tau(s,a)f(s',a')\right] \\
& - \mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[\tau(s,a)f(s,a)\right] - \frac{1}{2}\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}\left[f(s,a)^2\right] \\
& + \lambda\left(\mathbb{E}_{(s,a)\sim d_\mu(\cdot)}[\eta\tau(s,a) - \eta] - \frac{\eta^2}{2}\right).
\end{aligned}
$$

The optimization problem (6.6) is an *unconstrained* optimization problem and L is convex (linear) in τ and concave in f, η. Assuming τ, f is parameterized by w, κ respectively and including ridge regularization for w for reasons that will soon be clear, we consider the following problem

$$
\min_{w} \max_{\eta,\kappa} L(\tau_w, \eta, f_\kappa) + \frac{\xi}{2}||w||^2, \tag{6.7}
$$

where $\xi \geq 0$ is a constant. When a linear architecture is considered for τ_w and f_κ, the problem (6.7) is CCSP. Namely, it is convex in w and concave in κ, η. Recall that X is the state-action feature matrix. Assuming

$$
\tau_w \doteq Xw, f_\kappa \doteq X\kappa,
$$

we perform gradient descent on w and gradient ascent on η, κ in the i.i.d. setting in Definition 2.4. As we use techniques similar to Gradient TD methods to prove the convergence of our new algorithm, we term it Gradient stationary DIstribution Correction Estimation (GradientDICE, Algorithm 6). We now proceed to analyzing

Algorithm 6: GradientDICE

$k \leftarrow 0$
while *True* **do**
  Sample $S_k^0 \sim p_0(\cdot), A_k^0 \sim \pi(\cdot|S_k^0)$
  Sample $(S_k, A_k) \sim d_\mu(\cdot), R_k \doteq r(S_k, A_k), S_k' \sim p(\cdot|S_k, A_k), A_k' \sim \pi(\cdot|S_k')$
  $x_{0,k} \leftarrow x(S_k^0, A_k^0), x_k \leftarrow x(S_k, A_k), x_k' \leftarrow x(S_k', A_k')$
  $\delta_k \leftarrow (1-\gamma)x_{0,k} + \gamma x_k^\top w_k x_k' - x_k^\top w_k x_k$
  $\kappa_{k+1} \leftarrow \kappa_k + \alpha_k(\delta_k - x_k^\top \kappa_k x_k)$
  $\eta_{k+1} \leftarrow \eta_k + \alpha_k\lambda(x_k^\top w_k - 1 - \eta_k)$
  $w_{k+1} \leftarrow w_k - \alpha_k\left(\gamma x_k'^\top \kappa_k x_k - x_k^\top \kappa_k x_k + \lambda\eta_k x_k + \xi w_k\right)$
  $k \leftarrow k+1$
end

the convergence of GradientDICE with the following assumption.

Assumption 6.1. *The matrix $A = X^\top D_\mu(\gamma P_\pi - I)X$ is nonsingular or $\xi > 0$.*

When $\gamma < 1$, it is common to assume A is nonsingular (Maei, 2011), the ridge regularization (i.e., $\xi > 0$) is then optional. When $\gamma = 1$, A can easily be singular (e.g., in a tabular setting). We, therefore, impose the extra ridge regularization.

Theorem 6.3. *Let Assumptions 2.3, 2.4, and 6.1 hold. Then the iterates $\{w_k\}, \{\kappa_k\}, \{\eta_k\}$ generated by Algorithm 6 satisfy*

$$\lim_{k\to\infty} \begin{bmatrix} \kappa_k \\ w_k \\ \eta_k \end{bmatrix} = -G^{-1}g \quad a.s.,$$

where

$$G \doteq \begin{bmatrix} -C & A^\top & 0 \\ -A & -\xi I & -\lambda X^\top d_\mu \\ 0 & \lambda d_\mu^\top X & -\lambda \end{bmatrix}, g \doteq \begin{bmatrix} (1-\gamma)X^\top d_{p_0\pi} \\ 0 \\ -\lambda \end{bmatrix}.$$

See Section B.16 for the proof of Theorem 6.3, which is inspired by Sutton et al. (2009). One key step in the proof is to show that the real parts of all eigenvalues of G are strictly negative. The G in Sutton et al. (2009) satisfies this condition easily. However, for our G to satisfy this condition when $\gamma = 1$, we must have $\xi > 0$, which motivates the use of ridge regularization.

With simple block matrix inversion expanding G^{-1}, we have

$$\lim_{k\to\infty} w_k = w_{\infty,\xi},$$

where

$$\begin{aligned} w_{\infty,\xi} &\doteq -(1-\gamma)\Xi A C^{-1} X^\top d_{p_0\pi} + \lambda\beta^{-1} z \left(1 + (1-\gamma) z^\top A C^{-1} X^\top d_{p_0\pi}\right) \\ \Xi &\doteq (\xi I + A C^{-1} A^\top)^{-1}, \\ z &\doteq \Xi X^\top d_\mu, \quad \beta \doteq 1 + \lambda d_\mu^\top X \Xi X^\top d_\mu. \end{aligned}$$

The maximization step in (6.7) is quadratic (with linear function approximation) and thus can be solved analytically. Simple algebraic manipulation shows that this quadratic problem has a unique optimizer for all $\gamma \in [0, 1]$. Plugging the analytical solution for the maximization step in (6.7), the KKT conditions then state that the optimizer $w_{*,\xi}$ for the minimization step must satisfy $A_{*,\xi} w_{*,\xi} = b_*$, where

$$\begin{aligned} A_{*,\xi} &\doteq A C^{-1} A^\top + \lambda X^\top d_\mu d_\mu^\top X + \xi I, \\ b_* &\doteq -(1-\gamma) A C^{-1} X^\top \mu_0 + \lambda X^\top d_\mu. \end{aligned}$$

Assumption 6.1 ensures $A_{*,\xi}$ is nonsingular. Using the Sherman-Morrison formula (Sherman and Morrison, 1950), it is easy to verify $w_{*,\xi} = w_{\infty,\xi}$. In other words, GradientDICE is able to solve the problem (6.7).

To ensure convergence, we require ridge regularization in (6.7) when $\gamma = 1$. The asymptotic solution $w_{\infty,\xi}$ is therefore biased. We now study the regularization path consistency when $\gamma = 1$, i.e., we study the behavior of $w_{\infty,\xi}$ when ξ approaches 0. We start with a simple setting where $\tau_\gamma \in col(X)$. Here $col(\cdot)$ indicates the column space. As X has linearly independent columns, we use w_* to denote the unique w satisfying $Xw = \tau_\gamma$. When $\gamma = 1$, A can be singular. Hence both $w_{\infty,0}$ and $A_{*,0}^{-1}$ can be ill-defined. We now show under some regularization, we still have the desired consistency. As $AC^{-1}A^\top$ is always positive semidefinite, we consider its eigendecomposition $AC^{-1}A^\top = Q^\top \Lambda Q$, where Q is an orthogonal matrix, $\Lambda \doteq diag([\lambda_1, \cdots, \lambda_r, 0, \cdots, 0])$, r is the rank of $A^\top C^{-1} A$, and $\lambda_i > 0$ are eigenvalues. Let $u \doteq QX^\top d_\mu$. We have

Proposition 6.4. *Assuming $XC^{-1}X^\top$ is positive definite, $||u_{r+1:N_{sa}}|| \neq 0$, then*

$$\lim_{\xi \to 0} w_{\infty,\xi} = w_*,$$

where $u_{i:j}$ denotes the vector consisting of the elements indexed by $i, i+1, \ldots, j$ in the vector u.

The proof of Proposition 6.4 is provided in Section B.17. The assumption $||u_{r+1:N_{sa}}|| \neq 0$ is not restrictive as it is independent of learnable parameters and mainly controlled by features. Requiring $XC^{-1}X^\top$ to be positive definite is more restrictive, but it holds at least for the tabular setting (i.e., $X = I$). The difficulty of the setting $\gamma = 1$ comes mainly from the fact that the objective of the minimization step in the problem (6.7) is no longer strictly convex when $\xi = 0$ (i.e., $A_{*,0}$ can be singular). Thus there may be multiple optima for this minimization step, only one of which is w_*. Extra domain knowledge (e.g., assumptions in the proposition statement) is necessary to ensure the regularization path converges to the desired optimum. We provide a sufficient condition here and leave the investigation of necessary conditions for future work. When $\tau_\gamma \notin col(X)$, it is not clear how to define w_*. We leave the investigation of this scenario for future work.

6.3 Empirical Results

In this section, we present empirical results comparing GradientDICE with baseline algorithms. In particular, we consider DualDICE and GenDICE (Zhang et al., 2020a)

as baselines. DualDICE is convergent only in the discounted setting. GenDICE is designed for both the discounted and average reward settings. GenDICE, however, considers all the three constraints (6.1), (6.2), (6.3) and is thus not convergent in either setting. All the curves in this section are averaged over 30 independent runs and shaded regions indicate one standard derivation.

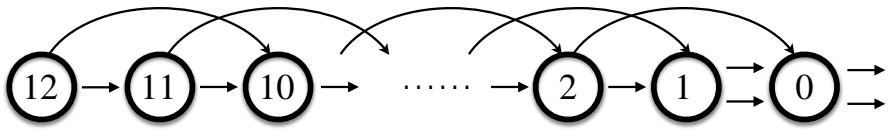

Figure 6.1: Two variants of Boyan's Chain. There are 13 states in total with two actions $\{a_0, a_1\}$ available at each state. The initial distribution p_0 is uniform over $\{s_0, \ldots, s_{12}\}$. At a state $s_i (i \geq 2)$, a_0 leads to s_{i-1} and a_1 leads to s_{i-2}. At s_1, both actions leads to s_0. At s_0, there are two variants. (1) `Episodic Boyan's Chain`: both actions at s_0 lead to s_0 itself, i.e., s_0 is an absorbing state. (2) `Continuing Boyan's Chain`: both actions at s_0 lead to a random state among $\{s_0, \ldots, s_{12}\}$ with equal probability.

We first benchmark the algorithms in terms of learning the density ratios. We consider two variants of Boyan's Chain (Boyan, 1999) as shown in Figure 6.1. In particular, we use `Episodic Boyan's Chain` when $\gamma < 1$ and `Continuing Boyan's Chain` when $\gamma = 1$. We consider a uniform sampling distribution, i.e., $d_\mu(s, a) = \frac{1}{26} \forall (s, a)$, and a target policy π satisfying $\pi(a_0|s) = 0.1 \, \forall s$. We design a sequence of tasks by varying the discount factor γ in $\{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$.

We train all compared algorithms for 3×10^4 steps. We evaluate the Mean Squared Error (MSE) for the predicted τ every 300 steps, computed as

$$\text{MSE}(\tau) \doteq \frac{1}{26} \sum_{s,a} (\tau(s, a) - \tau_\gamma(s, a))^2,$$

where the ground truth τ_γ is computed analytically. We use fixed learning rates α for all algorithms, which is tuned from $\{4^{-6}, 4^{-5}, \ldots, 4^{-1}\}$ to minimize the MSE(τ) at the end of training. For the setting $\gamma = 1$, we additionally tune ξ from $\{0, 10^{-3}, 10^{-2}, 10^{-1}\}$ (for a fair comparison, we also add this ridge regularization for GenDICE and DualDICE). For the penalty coefficient, we set $\lambda = 1$ as recommended by Zhang et al. (2020a). We find λ has little influence on the learning process in this domain.

We report the results in both tabular (Figure 6.2) and linear (Figure 6.3) settings. In the tabular setting, we use lookup tables to store τ, f and η. In the linear setting,

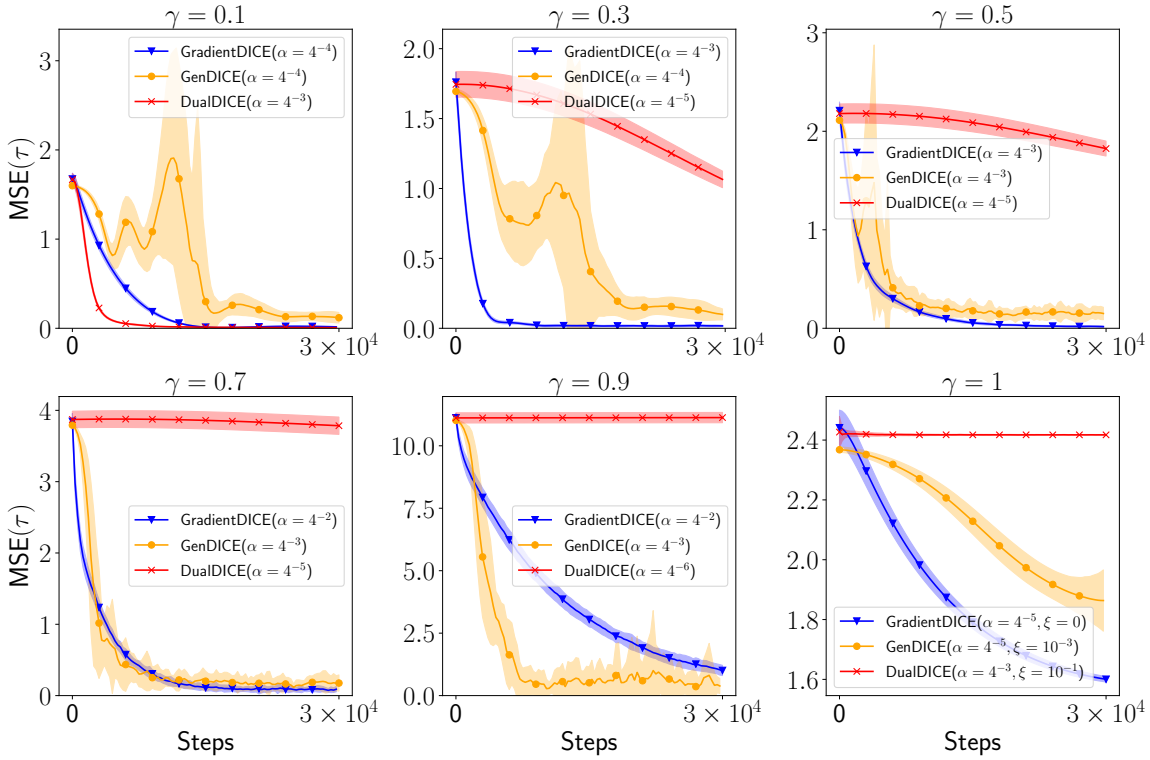

Figure 6.2: Density ratio learning in Boyan's Chain with a tabular representation.

we use the same state features as Boyan (1999):

$$
\begin{aligned}
x(s_{12}) &\doteq [1, 0, 0, 0]^\top \\
x(s_{11}) &\doteq [0.75, 0.25, 0, 0]^\top \\
x(s_{10}) &\doteq [0.5, 0.5, 0, 0]^\top \\
x(s_9) &\doteq [0.25, 0.75, 0, 0]^\top \\
x(s_8) &\doteq [0, 1, 0, 0]^\top \\
x(s_7) &\doteq [0, 0.75, 0.25, 0]^\top \\
x(s_6) &\doteq [0, 0.5, 0.5, 0]^\top \\
x(s_5) &\doteq [0, 0.25, 0.75, 0]^\top \\
x(s_4) &\doteq [0, 0, 1, 0]^\top \\
x(s_3) &\doteq [0, 0, 0.75, 0.25]^\top \\
x(s_2) &\doteq [0, 0, 0.5, 0.5]^\top \\
x(s_1) &\doteq [0, 0, 0.25, 0.75]^\top \\
x(s_0) &\doteq [0, 0, 0, 1]^\top
\end{aligned}
$$

We use two independent sets of weights for the two actions. As GenDICE requires

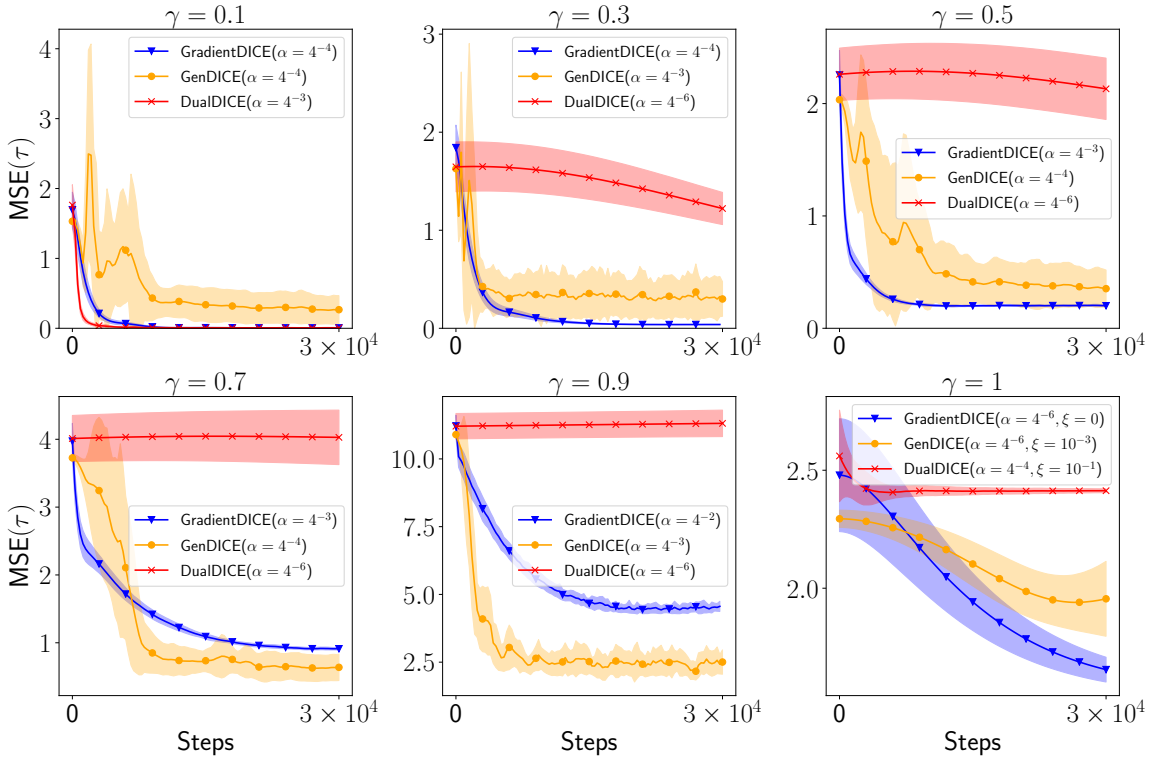

Figure 6.3: Density ratio learning in Boyan's Chain with a linear architecture.

$\tau(s, a) > 0 \, \forall (s, a)$, we use the nonlinearity $(\cdot)^2$ for its τ prediction as suggested by Zhang et al. (2020a). We do not apply any nonlinearity for GradientDICE and DualDICE. Our results show that GradientDICE reaches a lower prediction error at the end of training than GenDICE in 5 (4) out 6 tasks in the tabular (linear) setting. Moreover, the learning curves of GradientDICE are more stable than those of GenDICE in all the 6 tasks in both tabular and linear settings. Although DualDICE performs the best for the task $\gamma = 0.1$, it becomes unstable as γ increases, which is also observed in Zhang et al. (2020a).

We then benchmark DualDICE, GenDICE, and GradientDICE in an off-policy prediction problem. We consider `Reacher-v2` from OpenAI Gym (Brockman et al., 2016). We consider policies in the form of $\pi_d(s, a) + \mathcal{N}(0, \sigma^2)$, where π_d is a deterministic policy trained via TD3 (Fujimoto et al., 2018) for 10^6 steps and $\mathcal{N}$ is Gaussian noise. For the behavior policy, we set $\sigma = 0.1$ and run the policy for $N = 10^5$ steps to collect transitions, which form the dataset used across all the experiments. For the target policy, we set $\sigma = 0.05$.

We use neural networks to parameterize τ and f, each of which is represented by a two-hidden-layer network with 64 hidden units and ReLU (Nair and Hinton, 2010)

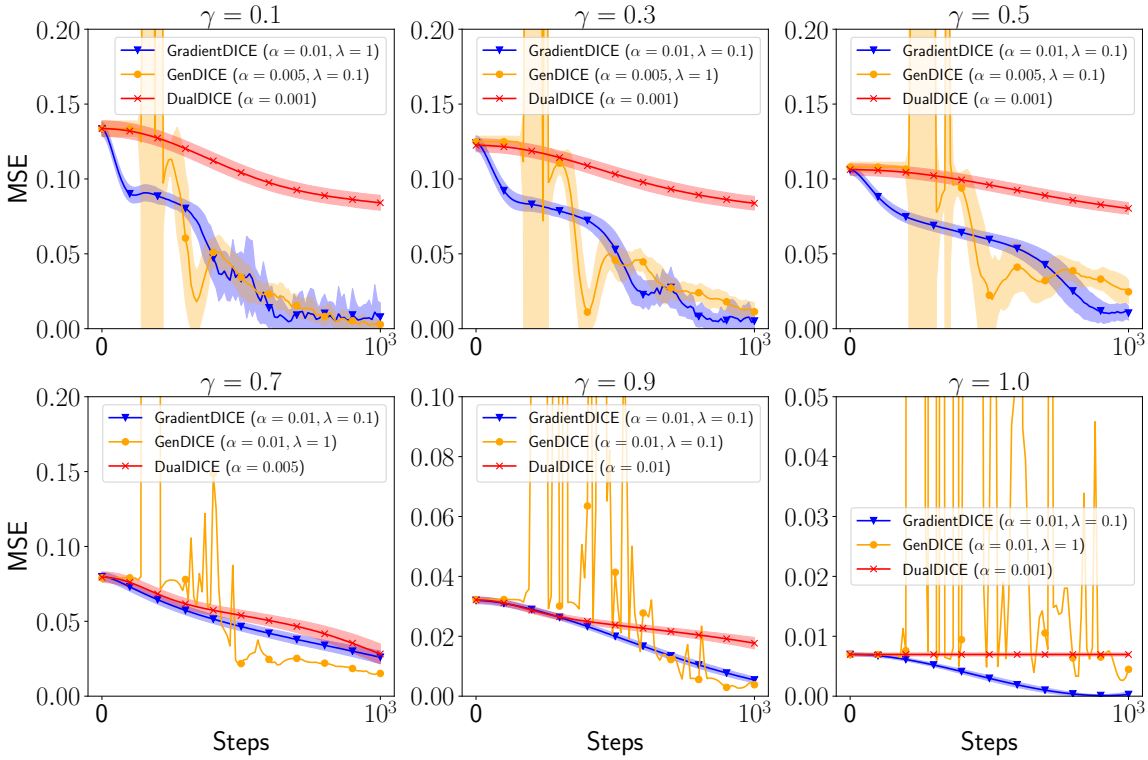

Figure 6.4: Off-policy prediction in `Reacher-v2` with neural network function approximators.

activation function. For GenDICE, we add $(\cdot)^2$ nonlinearity for the τ prediction by the network. For GradientDICE and DualDICE, we do not have such nonlinearity in their τ prediction. Given the learned τ, the performance $J_{\pi,\gamma}$ is approximated by

$$\hat{J}_{\pi,\gamma} \doteq \frac{1}{N} \sum_{i=1}^{N} \tau(s_i, a_i) r_i.$$

Here $J_{\pi,\gamma}$ denotes the normalized discounted total rewards $(1-\gamma)J_\pi$ when $\gamma < 1$ and the average reward $\bar{J}_\pi$ when $\gamma = 1$. We train each algorithm for 10^3 steps and examine

$$\text{MSE} \doteq \frac{1}{2} \left(\hat{J}_{\pi,\gamma} - J_{\pi,\gamma} \right)^2$$

every 10 steps, where the ground truth $J_{\pi,\gamma}$ is computed from Monte Carlo methods via executing the target policy π multiple times. We use SGD to train the neural networks with batch size 128. The learning rate α and the penalty coefficient λ are tuned from $\{0.01, 0.005, 0.001\}$ and $\{0.1, 1\}$ with grid search to minimize MSE at the end of training.

The results are reported in Figure 6.4. Although prediction errors of GradientDICE and GenDICE tend to be similar at the end of training, the learning curves of GradientDICE are more stable than those of GenDICE, which matches the results in the tabular and linear settings. Although DualDICE tends to be more stable than both GradientDICE and GenDICE, it learns slower and does not work for the setting $\gamma = 1$, which also matches the results in Zhang et al. (2020a). To summarize, GradientDICE combines the advantages of both DualDICE (stability in discounted setting) and GenDICE (compatibility with the average reward setting).

Chapter 7

Prediction with Target Networks

Though approximating the average reward $\bar{J}_\pi$ is already sufficient in some scenarios, in many other cases, learning the differential value functions $\bar{v}_\pi$ and $\bar{q}_\pi$ is also necessary, for example, when we want to consider the more ambitious control problems. In this chapter, we extend the target-network-based approach in Section 3 from the discounted setting to the average reward setting for learning the differential value functions.

7.1 Divergence of Differential Temporal Difference Methods

In this section, we motivate the introduction of target networks by showing that the straightforward semi-gradient method can possibly diverge. Consider, for example, the problem of estimating $\bar{q}_\pi$ and $\bar{J}_\pi$ in the i.i.d. setting in Definition 2.4. One natural approach is to equip (2.15) with linear function approximation. Let Xw be our estimate for $\bar{q}_\pi$ and $\hat{r}$ be our estimate for $\bar{J}_\pi$. We update w and $\hat{r}$ iteratively as

$$\begin{aligned} w_{k+1} &\doteq w_k + \alpha_k \delta_k(w_k, \hat{r}_k) x_k, \\ \hat{r}_{k+1} &\doteq \hat{r}_k + \alpha_k \delta_k(w_k, \hat{r}_k), \end{aligned} \tag{7.1}$$

where

$$\delta_k(w, \hat{r}) \doteq R_k - \hat{r} + x_k'^\top w - x_k^\top w$$

is the temporal difference error. We refer to the updates (7.1) as differential semi-gradient Q-evaluation, or Diff-SGQ. If Diff-SGQ converged, the expected updates must vanish. In other words, the limiting point $(w, \hat{r})$ must verify

$$\mathbb{E}\left[\delta_k(w, \hat{r}) x_k\right] = 0, \tag{7.2}$$

$$\mathbb{E}\left[\delta_k(w, \hat{r})\right] = 0. \tag{7.3}$$

We refer to $(w, \hat{r})$ that verifies (7.2) and (7.3) as *TD fixed points* in the average reward setting. Writing (7.3) in a vector form yields

$$\hat{r} = d_\mu^\top (r + P_\pi X w - X w). \tag{7.4}$$

Replacing $\hat{r}$ in (7.2) with (7.4) yields

$$X^\top D_\mu (r + P_\pi X w - X w) - X^\top D_\mu 1 d_\mu^\top (r + P_\pi X w - X w) = 0,$$

or equivalently,

$$\bar{A} w + \bar{b} = 0.$$

We recall that $\bar{A}$ and $\bar{b}$ are defined in Section 2.13. Depending on the structure of $\bar{A}$, there could be no TD fixed point, one TD fixed point, or infinitely many TD fixed points. Unfortunately, the following example demonstrates that the naive differential temporal difference method (7.1) does not necessarily converge, even if there is a unique TD fixed point.

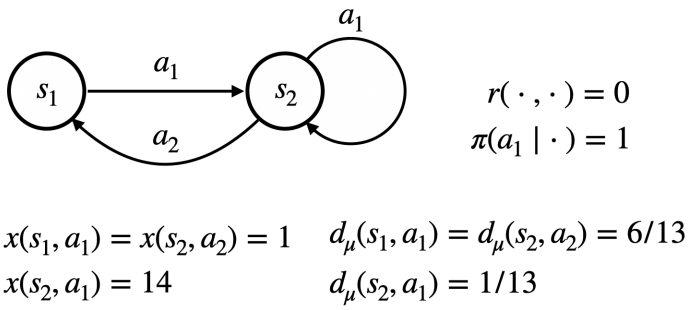

Figure 7.1: An example showing the divergence of naive differential temporal difference methods.

Example 7.1. *(From Zhang et al. (2021c)) Consider a two-state MDP (Figure 7.1). The expected per-step update of (7.1) in this MDP can be written as*

$$\begin{bmatrix} \hat{r}_{k+1} \\ w_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{r}_k \\ w_k \end{bmatrix} + \alpha \begin{bmatrix} -1 & 6 \\ -2 & 6 \end{bmatrix} \begin{bmatrix} \hat{r}_k \\ w_k \end{bmatrix}.$$

Here, we consider α a constant learning rate. The eigenvalues of $A = \begin{bmatrix} -1 & 6 \\ -2 & 6 \end{bmatrix}$ are both positive. Hence, no matter what positive learning rate is used, the expected update diverges. The sample update (7.1) using standard stochastic approximation learning rates, therefore, also diverge. Furthermore, because both eigenvalues are positive, A is an invertible matrix, implying the unique existence of the TD fixed-point.

7.2 Differential Expected SARSA for Prediction with A Target Network

Motivated by the success of target networks in addressing the deadly triad in the discounted setting, we in this section seek to use target networks to address the deadly triad in the average reward setting as well.

In the average-reward setting, we need to approximate $\bar{J}_\pi$ and $\bar{q}_\pi$ with $\hat{r}$ and Xw respectively. Hence, we consider target networks θ^r and θ^w for $\hat{r}$ and w respectively. Plugging θ^r and θ^w into (7.1) for bootstrapping yields differential off-policy linear expected SARSA for prediction with a target network (Algorithm 7), where $\{B_i\}$ are now balls in R^{K+1} (cf. (3.3)). In Algorithm 7, we impose ridge regularization only on w as $\hat{r}$ is a scalar and thus does not have any representation capacity limit.

Algorithm 7: Differential off-policy linear expected SARSA for prediction with a target network

Initialize $\theta_0 \in B_1$
$S_0 \sim p_0(\cdot)$
$t \leftarrow 0$
while *True* **do**
  $A_t \sim \mu(\cdot|S_t)$
  Execute A_t, get R_{t+1} and S_{t+1}
  $\delta_t \leftarrow R_{t+1} - \theta_t^r + \sum_{a'} \pi(a'|S_{t+1}) x(S_{t+1}, a')^\top \theta_t^w - x_t^\top w_t$
  $w_{t+1} \leftarrow w_t + \alpha_t \delta_t x_t - \alpha_t \eta w_t$
  $\hat{r}_{t+1} \leftarrow \hat{r}_t + \alpha_t (R_{t+1} + \sum_{a'} \pi(a'|S_{t+1}) x(S_{t+1}, a')^\top \theta_t^w - x_t^\top \theta_t^w - \hat{r}_t)$
  $\begin{bmatrix} \theta_{t+1}^r \\ \theta_{t+1}^w \end{bmatrix} \leftarrow \Gamma_{B_1} \left(\begin{bmatrix} \theta_t^r \\ \theta_t^w \end{bmatrix} + \beta_t \left(\Gamma_{B_2} \left(\begin{bmatrix} \hat{r}_t \\ w_t \end{bmatrix} \right) - \begin{bmatrix} \theta_t^r \\ \theta_t^w \end{bmatrix} \right) \right)$
  $t \leftarrow t + 1$
end

Theorem 7.1. *Let Assumptions 2.1, 2.3, 2.4, 2.5, 2.6, and 2.8 hold. For any $\xi \in (0, 1)$, there exist constants C_0 and C_1 such that for all*

$$\|X\| < C_0, C_1 < R_{B_1}, R_{B_1} - \xi < R_{B_2} < R_{B_1},$$

the iterates $\{\hat{r}_t\}$ and $\{w_t\}$ generated by Algorithm 7 satisfy

$$\lim_{t\to\infty} \hat{r}_t = \hat{r}_\eta^*,$$
$$\lim_{t\to\infty} w_t = w_\eta^* \quad a.s.,$$

where

$$\hat{r}_\eta^* \doteq d_\mu^\top (r + P_\pi X w_\eta^* - X w_\eta^*)$$

and w_η^* *is the unique solution to*

$$(\bar{A} - \eta I) w + \bar{b} = 0.$$

If features are zero-centered (i.e., $X^\top d_\mu = 0$*), then*

$$\begin{aligned} \left\| X w_\eta^* - \bar{q}_\pi^c \right\| &\leq \left(\frac{\sigma_{\max}(X)^2}{\sigma_{\min}(X)^4 \sigma_{\min}(D_\mu)^{2.5}} \|\bar{q}_\pi^c\| \eta + \left\| \Pi_{d_\mu} \bar{q}_\pi^c - \bar{q}_\pi^c \right\| \right) / \xi, \\ |\bar{r}_\eta^* - \bar{J}_\pi| &\leq \left\| d_\mu^\top (P_\pi - I) \right\| \inf_c \left\| (X w_\eta^* - \bar{q}_\pi^c) \right\|, \end{aligned}$$

where $\bar{q}_\pi^c \doteq \bar{q}_\pi + c1$.

We defer the proof to Section B.18. As the differential Bellman equation (2.6) has infinitely many solutions for q, all of which differ only by some constant offsets, we focus on analyzing the quality of $X w_\eta^*$ w.r.t. $\bar{q}_\pi^c$ in Theorem 7.1. The zero-centered feature assumption can be easily fulfilled in practice by subtracting all features with the estimated mean. In the on-policy case (i.e., $\mu = \pi$), we have $d_\mu^\top (P_\pi - I) = 0$, indicating $\bar{r}_\eta^* = \bar{J}_\pi$, i.e., the regularization on the differential value estimate does not pose any bias on the average reward estimate in the on-policy setting.

It is worth mentioning that target networks can also be used in learning the differential state value function $\bar{v}_\pi$. We in this section focus on learning the differential action value function $\bar{q}_\pi$ mainly to prepare us for the control setting in later sections.

Chapter 8

Prediction with Gradient Temporal Difference Methods

In Chapter 7, we introduce a target-network-based algorithm for learning the differential value functions, which, however, requires to have a sufficiently large regularization. In this chapter, we introduce a new algorithm that relaxes this constraint.

8.1 A New Mean Squared Projected Bellman Error Objective

Motivated by the success of GTD methods in the discounted setting, we, in this chapter, introduce gradient temporal difference methods in the average reward setting. The first step is to define an MSPBE objective. Consider approximating $\bar{q}_\pi$ with Xw. In the on-policy setting, finding an MSPBE objective for w is straightforward. One can, in analogue to (2.10), consider

$$\|\Pi_{d_\pi}(r - \hat{r}1 + P_\pi Xw) - Xw\|_{d_\pi}^2,$$

where w is the free variable and $\hat{r}$ is a *known* estimate of the average reward $\bar{J}_\pi$ (Tsitsiklis and Roy, 1999). This MSPBE objective is feasible because it is straightforward to get a decent $\hat{r}$ in the on-policy setting. For example, the empirical average of the received rewards when executing π is a good estimate for the average reward $\bar{J}_\pi$. It is attempting to similarly define an MSPBE objective for the off-policy setting in analogue to (2.22) as

$$\left\|\Pi_{d_\mu}(r - \hat{r}1 + P_\pi Xw) - Xw\right\|_{d_\mu}^2.$$

This MSPBE is, however, no longer feasible because obtaining a good estimate $\hat{r}$ for the average reward $\bar{J}_\pi$ is nontrivial in the off-policy setting. The empirical average

of the received rewards is no longer a good estimate because they are obtained from following the behavior policy μ instead of the target policy π. One has to resort to other complicated methods such as density ratio learning in Section 6 to get such an estimate.

To address this challenge, we replace $\hat{r}$ with $d_\mu^\top(r + P_\pi Xw - Xw)$ as suggested by (7.4), yielding the following MSPBE objective:

$$\text{MSPBE}(w) \doteq \left\|\Pi_{d_\mu}\left(r - d_\mu^\top(r + P_\pi Xw - Xw)1 + P_\pi Xw\right) - Xw\right\|_{d_\mu}^2. \tag{8.1}$$

Here we have borrowed the idea from Schwartz (1993); Singh (1994); Wan et al. (2021) that the TD error is a good estimate for the average reward.

8.2 Two-Stage Differential Gradient Q-Evaluation

We now proceed to designing a new gradient temporal difference method for the new MSPBE objective. In particular, we consider the following objective

$$L(w) \doteq \left\|\Pi_{d_\mu}\left(r - d_\mu^\top(r + P_\pi Xw - Xw)1 + P_\pi Xw\right) - Xw\right\|_{d_\mu}^2 + \eta\|w\|^2, \tag{8.2}$$

where we have added an additional ridge regularization term with a weight $\eta \geq 0$ to the MSPBE objective (8.1). Denote $\bar{\delta}_w \doteq r + P_\pi Xw - Xw$, we have

$$\begin{aligned}
&L(w)\\
=&\left\|\Pi_{d_\mu}(\bar{\delta}_w - d_\mu^\top\bar{\delta}_w 1)\right\|_{d_\mu}^2 + \eta\|w\|^2\\
=&\left\|X^\top D_\mu(\bar{\delta}_w - d_\mu^\top\bar{\delta}_w 1)\right\|_{(X^\top D_\mu X)^{-1}}^2 + \eta\|w\|^2 \quad \text{(cf. (2.23))}\\
=&\max_\nu 2\nu^\top X^\top D_\mu(\bar{\delta}_w - d_\mu^\top\bar{\delta}_w 1) - \nu^\top C\nu + \eta\|w\|^2 \quad \text{(Using (2.24))} \quad .
\end{aligned}$$

We, therefore, consider the following optimization problem:

$$\min_w \max_\nu L(w, \nu)$$

where

$$L(w, \nu) \doteq 2\nu^\top X^\top D_\mu(\bar{\delta}_w - d_\mu^\top\bar{\delta}_w 1) - \nu^\top C\nu + \eta\|w\|^2.$$

In other words, minimizing $L(w)$ now becomes finding the saddle point of $L(w, \nu)$, which is convex in w and concave in ν. We then use primal-dual methods to find the

saddle point, i.e., we perform gradient ascent for ν following $\nabla_\nu L(w,\nu)$ and gradient descent for w following $-\nabla_w L(w,\nu)$. It can be computed that

$$
\begin{aligned}
&\nabla_\nu L(w,\nu)\\
=&2\left(X^\top D_\mu(\bar{\delta}_w - d_\mu^\top \bar{\delta}_w 1)\right)^\top - 2\nu^\top X^\top D_\mu X\\
=&2\mathbb{E}\left[\left(r(s_1,a_1)+x(s_1',a_1')^\top w - x(s_1,a_1)^\top w\right)x(s_1,a_1)^\top\right] -\\
&2\mathbb{E}\left[x(s_1,a_1)^\top\right]\mathbb{E}\left[r(s_2,a_2)+x(s_2',a_2')^\top w - x(s_2,a_2)^\top w\right] -\\
&2\mathbb{E}\left[\nu^\top x(s_1,a_1)x(s_1,a_1)^\top\right],\\
&\nabla_w L(w,\nu)\\
=&2\mathbb{E}\left[\left(\nu^\top x(s_1,a_1)\right)\left(x(s_1',a_1')-x(s_1,a_1)\right)^\top\right] -\\
&2\mathbb{E}\left[\left(\nu^\top x(s_1,a_1)\right)\left(x(s_2',a_2')-x(s_2,a_2)\right)^\top\right] + \eta w,
\end{aligned}
$$

where $(s_1,a_1,s_1',a_1') \sim d_{\mu p\pi}(\cdot)$ and $(s_2,a_2,s_2',a_2') \sim d_{\mu p\pi}(\cdot)$. Now it becomes clear that in the i.i.d. setting in Definition 2.4, we would require two i.i.d. samples $(S_{k,1},A_{k,1},R_{k,1},S_{k,1}',A_{k,1}')$ and $(S_{k,2},A_{k,2},R_{k,2},S_{k,2}',A_{k,2}')$ from $d_{\mu p\pi}$ at the k-th iteration for a single gradient update. This is not the notorious double sampling issue in minimizing MSBE, where two successor states s_1' and s_2' from a single state action pair (s,a) are required, which is not possible in the function approximation setting. Sampling two i.i.d. tuples from $d_{\mu p\pi}$ is completely feasible (e.g., we can perform one update every two steps). We now arrive at Algorithm 8 for the i.i.d. setting in Definition 2.4. Since in Algorithm 8, the average reward estimate $\hat{r}$ cannot converge until the differential action value estimate w has converged, we term Algorithm 8 two-state differential gradient Q-evaluation, or Diff-GQ2. We now proceed to analyzing the convergence of Diff-GQ2.

Assumption 8.1. *There exists at least one TD fixed point, i.e., the linear equations (7.2) and (7.3) have at least one solution.*

Theorem 8.1. *Let Assumptions 2.1, 2.3, 2.4, and 2.5 hold. Let $\eta > 0$. Then almost surely, the iterates $\{w_k\},\{\hat{r}_k\}$ generated by Diff-GQ2 (Algorithm 8) satisfy*

$$
\begin{aligned}
\lim_{k\to\infty} w_k &= w_\eta^*,\\
\lim_{k\to\infty} \hat{r}_k &= d_\mu^\top(r + P_\pi X w_\eta^* - X w_\eta^*),
\end{aligned}
$$

where

$$
w_\eta^* \doteq -(\eta I + \bar{A}^\top C^{-1}\bar{A})^{-1}\bar{A}^\top C^{-1}\bar{b}
$$

Algorithm 8: Two-state differential gradient Q evaluation

$t \leftarrow 0$
while *True* **do**
  Sample $(S_{k,1}, A_{k,1}, R_{k,1}, S'_{k,1}, A'_{k,1})$
  Sample $(S_{k,2}, A_{k,2}, R_{k,2}, S'_{k,2}, A'_{k,2})$
  $x_{k,1} \leftarrow x(S_{k,1}, A_{k,1}), x'_{k,1} \leftarrow x(S'_{k,1}, A'_{k,1})$
  $x_{k,2} \leftarrow x(S_{k,2}, A_{k,2}), x'_{k,2} \leftarrow x(S'_{k,2}, A'_{k,2})$
  $\delta_{k,1} \leftarrow R_{k,1} + x'^{\top}_{k,1} w_k - x^{\top}_{k,1} w_k$
  $\delta_{k,2} \leftarrow R_{k,2} + x'^{\top}_{k,2} w_k - x^{\top}_{k,2} w_k$
  $\nu_{k+1} \leftarrow \nu_k + \alpha_k(\delta_{k,1} - \delta_{k,2} - x^{\top}_{k,1}\nu_k)x_{k,1},$
  $w_{k+1} \leftarrow w_k + \alpha_k(x_{k,1} - x'_{k,1} + (x_{k,2} - x'_{k,2}))x^{\top}_{k,1}\nu_k - \alpha_k \eta w_k,$
  $\hat{r}_{k+1} \leftarrow \hat{r}_k + \beta_k \left(\frac{\delta_{k,1}+\delta_{k,2}}{2} - \hat{r}_k\right)$
  $t \leftarrow t + 1$
end

is the unique minimizer of $L(w)$ *in* (8.2)*. Define*

$$w_0^* \doteq \lim_{\eta \to 0} w_\eta^*,$$

we then have

$$\left\| w_\eta^* - w_0^* \right\| = \mathcal{O}(\eta).$$

Further, if Assumption 8.1 holds, then $\bar{A} w_0^* + \bar{b} = 0$*, and if* $\bar{A}$ *is invertible, then for* $\eta = 0$*,* $\{w_k\}$ *and* $\{\hat{r}_k\}$ *converge almost surely to the unique solution of* (7.2) *and* (7.3)*.*

The proof of Theorem 8.1 is provided in Section B.19. Theorem 8.1 confirms that Diff-GQ2 converges to an η-regularized TD fixed point and the regularization bias is proportional to η. Bounding the performance of w_0^*, a TD fixed point, is then sufficient for bounding the performance of w_η^*.

To understand the quantity of TD fixed points, we have to first assume its existence. Let Assumption 8.1 hold and let $(w^*, \hat{r}^*)$ be one TD fixed point. We are interested in bounding the difference between the estimated average reward and the true average reward, i.e., $\left|\hat{r}^* - \bar{J}_\pi\right|$, and the minimum distance between the estimated differential value function to the set $\{\bar{q}_\pi + c1 \mid c \in \mathbb{R}\}$. In general, as long as there is representation error, the performance of TD fixed points can be arbitrarily poor even in the discounted setting (Kolter, 2011). We, therefore, in this section study the bounds only when d_μ is close to d_π, in the sense of the following assumption. Let $\xi \in (0, 1)$ be a constant.

Assumption 8.2. *The matrix*

$$\begin{bmatrix} X^\top D_\mu X & X^\top D_\mu P_\pi X \\ X^\top P_\pi^\top D_\mu X & \xi^2 X^\top D_\mu X \end{bmatrix}.$$

is positive semidefinite.

A similar assumption is also used by Kolter (2011) in the analysis of the performance of the MSPBE minimizer in the discounted setting. Kolter (2011) uses $\xi = 1$ while we use $\xi < 1$ to account for the lack of discounting. Furthermore, we consider the bounds for zero-centered features.

Assumption 8.3. $X^\top d_\mu = 0$.

This can easily be done by subtracting each feature vector sampled in our learning algorithm by some estimated mean feature vector, which is the empirical average of all the feature vectors sampled from d_μ. Note without this mean-centered feature assumption, a looser bound can also be obtained. Our intention here is to show that bounds of our algorithms are on par with their counterparts in the discounted setting and thus one does not lose these bounds when one moves from the discounted setting to the average-reward setting.

Proposition 8.2. *Let Assumptions 2.1, 2.3, 8.1 - 8.3 hold. Then*

$$\inf_{c\in\mathbb{R}} \|Xw^* - q_\pi^c\|_{d_\mu} \leq \frac{\|P_\pi\|_{d_\mu} + 1}{1-\xi} \inf_{c\in\mathbb{R}} \left\|\Pi_{d_\mu} q_\pi^c - q_\pi^c\right\|_{d_\mu},$$

$$|\hat{r}^* - \bar{J}_\pi| \leq \frac{\left\|d_\mu^\top (P_\pi - I)\right\|_{D_\mu^{-1}} (\|P_\pi\|_{d_\mu} + 1)}{1-\xi} \inf_{c\in\mathbb{R}} \left\|\Pi_{d_\mu} q_\pi^c - q_\pi^c\right\|_{d_\mu},$$

where $q_\pi^c \doteq q_\pi + c1$.

We defer the proof to Section B.20. As a special case, there exists a unique TD fixed point in the on-policy case (i.e., $d_\mu = d_\pi$) under Assumptions 2.1 and 2.2. Then we have $|\hat{r}^* - \bar{J}_\pi| = 0$ since $d_\pi^\top (P_\pi - I) = 0$. A tighter bound for the estimated differential value function can also be obtained (see, e.g., Tsitsiklis and Roy (1999)).

We are now ready to compare the target-network-based approach Algorithm 7 in Chapter 7 and Diff-GQ2. Algorithm 7 requires a sufficiently large ridge regularization. Diff-GQ2 also requires the presence of a ridge regularization term but it can be arbitrarily small. This indicates that Diff-GQ2 suffers less from the bias introduced by the regularization. However, the performance of the Diff-GQ2 fixed points can be bounded only when d_μ is sufficiently close to d_π. By contrast, the performance of the fixed points of Algorithm 7 can always be bounded regardless of the difference between the behavior policy and the target policy. To summarize, there is no clear winner between Algorithm 1 and Diff-GQ2.

8.3 Yet Another Mean Squared Projected Bellman Error Objective

In this chapter, we propose a new off-policy MSPBE objective for the average reward setting and develop the corresponding Diff-GQ2 algorithm. In the discounted setting, there is only one off-policy MSPBE. In the average reward setting, besides the off-policy MSPBE introduced in this chapter, there also exists yet another off-policy MSPBE objective. The corresponding algorithm optimizing the other MSPBE objective is referred to as one-stage differential gradient Q-evaluation, or Diff-GQ1. We refer the reader to Zhang et al. (2021c) for more details. In the following, we briefly discuss the difference between the MSPBE objective proposed in this chapter (referred to as MSPBE_2) and the other MSPBE objective in Zhang et al. (2021c) (referred to as MSPBE_1), as well as the difference between Diff-GQ2 in this chapter and Diff-GQ1 in Zhang et al. (2021c).

- MSPBE_2 has only one free variable w corresponding to the action value function estimation; MSPBE_1 has two free variables w and $\hat{r}$ corresponding to the action value function estimation and the average reward estimation respectively.
- Diff-GQ2 requires two i.i.d. samples per update; Diff-GQ1 requires only one i.i.d. sample per update.
- For Diff-GQ2 to converge, we require only Assumption 2.3; for Diff-GQ1 to converge, we require a stronger Assumption 2.2.
- Diff-GQ2 is two-stage in the sense that it learns $\bar{q}_\pi$ first then $\bar{J}_\pi$; Diff-GQ1 is one-stage in the sense that it learns both $\bar{q}_\pi$ and $\bar{J}_\pi$ simultaneously.

There are, however, also several similarities.

- Both MSPBE_2 and MSPBE_1 share the same minimizer.
- If the ridge regularization is not in effect (i.e., $\eta = 0$), Diff-GQ2 and Diff-GQ1 converge to the same fixed point.

In the next section, we further provide some empirical results comparing Diff-GQ1 and Diff-GQ2.

8.4 Empirical Results

In this section, we empirically compare four different algorithms for estimating the average reward $\bar{J}_\pi$: Diff-SGQ, Diff-GQ1, Diff-GQ2, and GradientDICE. We use `Continuing Boyan's Chain` in Figure 6.1 as our testbed. Since we are interested in the prediction problem, we add a nonzero reward for each action. Namely, the action a_0 always generates a reward $+2$ and the action a_1 always generates a reward $+1$. We consider target policies of the form $\pi(a_1|s_i) = \pi_0$ for all s_i, where $\pi_0 \in [0, 1]$ is some constant. The sampling distribution we consider has the form $d_\mu(s_i, a_1) = \frac{\mu_0}{13}$ and $d_\mu(s_i, a_1) = \frac{1-\mu_0}{13}$ for all s_i, where $\mu_0 \in [0, 1]$ is some constant. Note that even if $\mu_0 = \pi_0$, the problem is still off-policy. We consider linear function approximation and use the same state features as Boyan (1999). We use a one-hot encoding for actions. Concatenating the state feature and the one-hot action feature yields the state-action feature we use in the experiments.

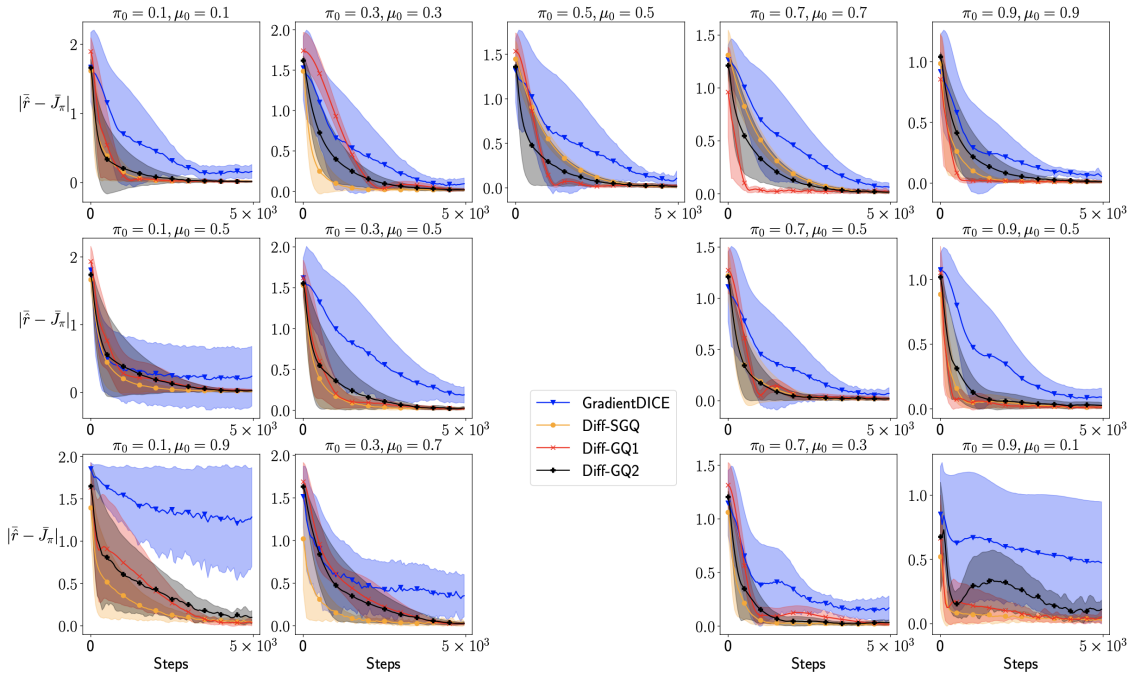

Figure 8.1: Boyan's chain with linear function approximation. We vary π_0 in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. In the first row, we use $\mu_0 = \pi_0$; in the second row, we use $\mu_0 = 0.5$; in the third row, we use $\mu_0 = 1 - \pi_0$. $\bar{\hat{r}}$ is the average $\hat{r}$ of recent 100 steps.

We use constant learning rates α for all compared algorithms, which is tuned in $\{2^{-20}, 2^{-19}, \ldots, 2^{-1}\}$. For Diff-GQ1 and Diff-GQ2, besides tuning α in the same way as Diff-SGQ, we tune η in $\{0, 0.01, 0.1\}$. For GradientDICE, besides tuning

(α, η) in the same way as Diff-GQ2, we tune λ, the weight for a normalizing term, in $\{0, 0.1, 1, 10\}$. we perform a grid search with 30 independent runs for hyperparameter tuning in all our experiments. Each curve corresponds to the best hyperparameters minimizing the error of the average reward prediction at the end of training and is averaged over 30 independent runs with the shaded region indicating one standard deviation.

We run each algorithm for 5×10^3 steps. Diff-GQ2 updates are applied every two steps as one Diff-GQ2 update requires two samples. The results in Figure 8.1 suggest that the three differential-value-based algorithms perform similarly and consistently outperform the density-ratio-based algorithm GradientDICE in the tested domain.

Part III

Value-Based Off-Policy Control

In this part, we focus on control problems with value-based methods. The general idea is to learn the action value function with function approximation and derive a policy from the approximated action value function. We discuss mainly the discounted setting with some coverage of the average reward setting. We extend the use of target networks and truncated followon traces from the prediction setting in the previous parts to the control setting. We also propose a new bi-directional target network for improving residual algorithms.

Chapter 9

Control with Target Networks

In Chapters 3 and 7, we use target networks for addressing the deadly triad in prediction settings. In this chapter, we extend those ideas to the control settings.

9.1 Q-Learning with A Target Network

Target networks are first introduced in DQN as an empirical trick for stabilizing Q-learning with deep networks, similar to other tricks such as dropout, layer norm for deep learning. In this section, we show that target networks are more than an ad-hoc trick for Q-learning.

Algorithm 9: Q-learning with a target network

Initialize $\theta_0 \in B_1$
$S_0 \sim p_0(\cdot)$
$t \leftarrow 0$
$A_t \sim \mu_{\theta_0}(\cdot|S_t)$
while *True* **do**
  Execute A_t, get R_{t+1} and S_{t+1}
  Sample $A_{t+1} \sim \mu_{\theta_t}(\cdot|S_t)$
  $\delta_t \leftarrow R_{t+1} + \gamma \max_{a'} x(S_{t+1}, a')^\top \theta_t - x_t^\top w_t$
  $w_{t+1} \leftarrow w_t + \alpha_t \delta_t x_t - \alpha_t \eta w_t$
  $\theta_{t+1} \leftarrow \Gamma_{B_1}\big(\theta_t + \beta_t(\Gamma_{B_2}(w_t) - \theta_t)\big)$
  $t \leftarrow t + 1$
end

We consider the Markovian setting in Definition 2.1 and analyze linear Q-learning with a target network (Algorithm 9). Here Γ_{B_1} and Γ_{B_2} are projections defined in (3.3). Similar to the prediction setting in Chapter 3, we add ridge regularization when updating the main network w. The behavior policy μ_θ depends on the target

network θ through the action value estimate $X\theta$ and can be any policy satisfying the following two assumptions.

Assumption 9.1. *Let Λ_μ be the closure of $\left\{P_{\mu_\theta} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|} \mid \theta \in \mathbb{R}^K\right\}$. For any $P \in \Lambda_\mu$, the Markov chain evolving in $\mathcal{S}\times\mathcal{A}$ induced by P is ergodic.*

Assumption 9.2. *$\mu_\theta(a|s)$ is Lipschitz continuous in $X_s\theta$, where $X_s \in \mathbb{R}^{|\mathcal{A}|\times K}$ is the feature matrix for the state s, i.e., its a-th row is $x(s,a)^\top$.*

A similar assumption to Assumption 9.1 is also used by Marbach and Tsitsiklis (2001) in their analysis of on-policy actor-critic methods. When the behavior policy μ is fixed (independent of θ), the induced chain is usually assumed to be ergodic when analyzing the behavior of Q-learning (see, e.g., Melo et al. (2008); Chen et al. (2019); Cai et al. (2019)). In Algorithm 9, the behavior policy μ_θ changes every step, so it is natural to assume that those behavior policies, as well as their limits, induce ergodic chains. Assuming the chain induced by a uniformly random policy is ergodic, which we argue is a fairly weak assumption, Assumptions 9.1 and 9.2 can be easily fulfilled. For example, one can use a softmax policy with a temperature $\iota \in (0,\infty)$:

$$\mu_\theta(a|s) = \frac{\exp\bigl(x(s,a)^\top\theta/\iota\bigr)}{\sum_{b\in\mathcal{A}}\exp(x(s,b)^\top\theta/\iota)}. \tag{9.1}$$

Theorem 9.1. *Under Assumptions 2.3, 2.4, 2.5, 2.6, 9.1, and 9.2, for any*

$$\xi \in (0,1), R_{B_1} > R_{B_2} > R_{B_1} - \xi > 0,$$

there exists a constant C_0 such that for all $\|X\| < C_0$, the iterate $\{w_t\}$ generated by Algorithm 9 satisfies

$$\lim_{t\to\infty} w_t = w_\eta^* \quad a.s.,$$

where w_η^* is the unique solution of

$$(A_{\pi_w,\mu_w} - \eta I)w + b_{\mu_w} = 0 \tag{9.2}$$

inside B_1. Here π_w denotes the greedy policy w.r.t. $X_s w$ with random tie breaking.

We defer the proof to Section B.21. Analogously to the prediction setting, if we call the solutions to $A_{\pi_w,\mu_w}w + b_{\mu_w} = 0$ TD fixed points for control in the discounted setting, then Theorem 9.1 asserts that Algorithm 9 finds a regularized TD fixed point.

Algorithm 9 and Theorem 9.1 are significant in two aspects. *First*, in Algorithm 9, the behavior policy is a function of the target network and thus changes every time

step. By contrast, previous work on Q-learning with function approximation (e.g., Melo et al. (2008); Maei et al. (2010); Chen et al. (2019); Cai et al. (2019); Chen et al. (2020a); Lee and He (2019b); Xu and Gu (2020); Carvalho et al. (2020); Wang and Zou (2020)) usually assumes the behavior policy is fixed. Though Fan et al. (2020) also adopt a changing behavior policy, they consider bi-level optimization. At each time step, the nested optimization problem must be solved exactly, which is computationally expensive and sometimes infeasible. To the best of our knowledge, we are the first to analyze Q-learning with function approximation under a changing behavior policy and without nested optimization problems. Compared with the fixed behavior policy setting or the bi-level optimization setting, our two-timescale setting with a changing behavior policy is more closely related to actual practice (e.g., Mnih et al. (2015); Lillicrap et al. (2016)).

Second, Theorem 9.1 does not enforce any similarity between μ_θ and π_w; they can be arbitrarily different. By contrast, previous work (e.g., Melo et al. (2008); Chen et al. (2019); Cai et al. (2019); Xu and Gu (2020); Lee and He (2019b)) usually requires the strong assumption that the fixed behavior policy μ is sufficiently close to the target policy π_w. As the target policy (i.e., the greedy policy) can change every time step due to the changing action-value estimates, this strong assumption rarely holds. While some work removes this strong assumption, it introduces other problems instead. In Greedy-GQ, Maei et al. (2010) avoid this strong assumption by computing sub-gradients of an MSPBE objective

$$\text{MSPBE}(w) \doteq \|A_{\pi_w,\mu} w - b_\mu\|^2_{C_\mu^{-1}} \tag{9.3}$$

directly. If linear Q-learning (2.21) under a fixed behavior policy μ converged, it converged to the minimizer of (9.3). Greedy-GQ, however, converges only to a stationary point of (9.3). By contrast, Algorithm 9 converges to a minimizer of our regularized MSPBE (cf. (9.2)). In Coupled Q-learning, Carvalho et al. (2020) avoid this strong assumption by using a target network as well, which they update as

$$\theta_{t+1} \doteq \theta_t + \alpha_t((x_t x_t^\top) w_t - \theta_t).$$

This target network update deviates much from the commonly used Polyak-averaging style update, while our (3.2) is identical to the Polyak-averaging style update most times if the balls for projection are sufficiently large. Coupled Q-learning updates the main network w as usual, replacing the $x(S_{t+1}, a)^\top w_t$ with $x(S_{t+1}, a)^\top \theta_t$ in (2.21).

With the Coupled Q-learning updates, Carvalho et al. (2020) prove that the main network and the target network converge to $\bar{w}$ and $\bar{\theta}$ respectively, which satisfy

$$X\bar{w} = XX^\top D_\mu \mathcal{T}_{\pi_{\bar{w}}} X\bar{w}, \quad X\bar{\theta} = \Pi_{d_\mu} \mathcal{T}_{\pi_{\bar{w}}} X\bar{w}.$$

It is, however, not clear how $\bar{w}$ and $\bar{\theta}$ relate to TD fixed points. Yang et al. (2019) also use a target network to avoid this strong assumption. Their target network update is the same as (3.2) except that they have only one projection Γ_{B_1}. Consequently, they face the problem of the reflection term $\zeta(t)$ (cf. (3.4)). They also assume the main network $\{w_t\}$ is always bounded, a strong assumption that we do not require. Moreover, they consider a fixed sampling distribution for obtaining i.i.d. samples, while our data collection is done by executing the changing behavior policy μ_θ in the MDP.

Other convergence results of Q-learning with function approximation include Tsitsiklis and Roy (1996b); Szepesvári and Smart (2004), which require special approximation architectures, Wen and Roy (2013); Du et al. (2020), which consider deterministic MDPs, Li et al. (2011); Du et al. (2019), which require a special oracle to guide exploration, Chen et al. (2020a), which require matrix inversion every time step, and Wang et al. (2019); Yang and Wang (2019, 2020); Jin et al. (2020), which consider linear MDPs (i.e., both p and r are assumed to be linear).

Achiam et al. (2019) characterize the divergence of Q-learning with nonlinear function approximation via Taylor expansions and use preconditioning to empirically stabilize training.

9.2 Gradient Q-Learning with A Target Network

One limit of Theorem 9.1 is that the bound on $\|X\|$ (i.e., C_0) depends on $1/R_{B_1}$ (see the proof in Section B.21 for the analytical expression), which means C_0 could potentially be small. Though we can use a small η accordingly to ensure that the regularization effect of η is modest, a small C_0 may not be desirable in some cases. To address this issue, we propose gradient Q-learning with a target network, inspired by Greedy-GQ. We first equip (9.3) with a changing behavior policy μ_w, yielding the following MSPBE objective

$$\|A_{\pi_w,\mu_w} w - b_{\mu_w}\|^2_{C^{-1}_{\mu_w}}.$$

We then use the target network θ in place of w in the non-convex components, yielding

$$L(w,\theta) \doteq \|A_{\pi_\theta,\mu_\theta} w - b_{\mu_\theta}\|^2_{C^{-1}_{\mu_\theta}} + \eta\|w\|^2, \tag{9.4}$$

where we have also introduced a ridge term. At time step t, we update w_t with primal-dual methods similar to GTD, GradientDICE, and Diff-GQ2 and update the target network θ_t as usual. Details are provided in Algorithm 10, where Γ_{B_1} and Γ_{B_2} are projections defined in (3.3).

Algorithm 10: Gradient Q-learning with a target network

Initialize $\theta_0 \in B_1$
$S_0 \sim p_0(\cdot)$
$t \leftarrow 0$
$A_t \sim \mu_{\theta_0}(\cdot|S_t)$
while *True* **do**
  Execute A_t, get R_{t+1} and S_{t+1}
  Sample $A_{t+1} \sim \mu_{\theta_t}(\cdot|S_t)$
  $\delta_t \leftarrow R_{t+1} + \gamma \sum_{a'} \pi_{\theta_t}(a'|S_{t+1}) x(S_{t+1}, a')^\top w_t - x_t^\top w_t$
  $u_{t+1} \leftarrow u_t + \alpha_t(\delta_t - x_t^\top u_t) x_t$
  $w_{t+1} \leftarrow w_t + \alpha_t \left(x_t - \gamma \sum_{a'} \pi_{\theta_t}(a'|S_{t+1}) x(S_{t+1}, a')\right) x_t^\top u_t - \alpha_t \eta w_t$
  $\theta_{t+1} \leftarrow \Gamma_{B_1}\left(\theta_t + \beta_t(\Gamma_{B_2}(w_t) - \theta_t)\right)$
  $t \leftarrow t + 1$
end

In Algorithm 9, the target policy π_θ is a greedy policy, which is not continuous in θ. This discontinuity is not a problem there but requires sub-gradients in the analysis of Algorithm 10, which complicates the presentation. We, therefore, impose Assumption 9.2 on the target policy as well.

Assumption 9.3. *$\pi_\theta(a|s)$ is Lipschitz continuous in $X_s\theta$.*

Though a greedy policy no longer satisfies Assumption 9.3, we can simply use a softmax policy (cf. (9.1)) with small temperature to approximate a greedy policy.

Theorem 9.2. *Under Assumptions 2.3, 2.4, 2.5, 2.6, 9.1, 9.2, and 9.3, there exist positive constants C_0 and C_1 such that for all*

$$\|X\| < C_0, R_{B_1} > R_{B_2} > C_1,$$

the iterate $\{w_t\}$ generated by Algorithm 10 satisfies

$$\lim_{t\to\infty} w_t = w_\eta^* \quad a.s.,$$

where w_η^* is the unique solution of

$$(A_{\pi_w,\mu_w}^\top C_{\mu_w}^{-1} A_{\pi_w,\mu_w} + \eta I)w + A_{\pi_w,\mu_w}^\top C_{\mu_w}^{-1} b_{\mu_w} = 0.$$

We defer the proof to Section B.22. Importantly, the C_0 here does not depend on R_{B_1} and R_{B_2}. Define

$$w_0^* \doteq \lim_{\eta \to 0} w_\eta^*.$$

If we further assume $A_{\pi_{w_0^*}, \mu_{w_0^*}}$ is invertible, we can then show similarly to Theorem 8.1 that

$$A_{\pi_{w_0^*}, \mu_{w_0^*}} w_0^* + b_{\mu_{w_0^*}} = 0,$$

indicating w_0^* is a TD fixed point. The fixed point w_η^* can therefore be regarded as a regularized TD fixed point, though how the regularization is imposed here (cf. (9.4)) is different from that in Algorithm 9 (cf. (9.2)).

9.3 Differential Q-Learning with A Target Network

In this section, we extend the success of target networks for control from the discounted setting to the average reward setting. Similar to Algorithm 9, introducing a target network and ridge regularization in (2.17) yields differential Q-learning with a target network (Algorithm 11). Similar to Algorithm 7, $\{B_i\}$ are now balls in R^{K+1} (cf. (3.3)).

Algorithm 11: Differential Q-learning with a target network

Initialize $\theta_0 \in B_1$
$S_0 \sim p_0(\cdot)$
$t \leftarrow 0$
$A_t \sim \mu_{\theta_0^w}(\cdot|S_t)$
while *True* **do**
  Execute A_t, get R_{t+1} and S_{t+1}
  Sample $A_{t+1} \sim \mu_{\theta_t^w}(\cdot|S_t)$
  $\delta_t \leftarrow R_{t+1} - \theta_t^r + \max_{a'} x(S_{t+1}, a')^\top \theta_t^w - x_t^\top w_t$
  $w_{t+1} \leftarrow w_t + \alpha_t \delta_t x_t - \alpha_t \eta w_t$
  $\delta_t' \leftarrow R_{t+1} + \max_{a'} x(S_{t+1}, a')^\top \theta_t^w - x_t^\top \theta_t^w - \hat{r}_t$
  $\hat{r}_{t+1} \leftarrow \hat{r}_t + \alpha_t \delta_t'$
  $\begin{bmatrix} \theta_{t+1}^r \\ \theta_{t+1}^w \end{bmatrix} \leftarrow \Gamma_{B_1}\left(\begin{bmatrix} \theta_t^r \\ \theta_t^w \end{bmatrix} + \beta_t \left(\Gamma_{B_2}\left(\begin{bmatrix} \hat{r}_t \\ w_t \end{bmatrix}\right) - \begin{bmatrix} \theta_t^r \\ \theta_t^w \end{bmatrix}\right)\right)$
  $t \leftarrow t + 1$
end

Theorem 9.3. *Let Assumptions 2.3, 2.4, 2.5, 2.6, 9.1, and 9.2 hold. Let L_μ denote the Lipschitz constant of μ_{θ^w}. Then for any*

$$\xi \in (0,1), R_{B_1} > R_{B_2} > R_{B_1} - \xi > 0,$$

there exist constants C_0 and C_1 such that for all

$$\|X\| < C_0, L_\mu < C_1,$$

the iterate $\{w_t\}, \{\hat{r}_t\}$ generated by Algorithm 11 satisfies

$$\begin{aligned}
\lim_{t\to\infty} w_t =& w_\eta^*, \\
\lim_{t\to\infty} \hat{r}_t =& d_{\mu_{w_\eta^*}}^\top \left(r + P_{\pi_{w_\eta^*}} X w_\eta^* - X w_\eta^*\right) \quad a.s.,
\end{aligned}$$

where w_η^* is the unique solution of

$$(\bar{A}_{\pi_w,\mu_w} - \eta I)w + \bar{b}_{\mu_w} = 0$$

inside B_1. Here π_w is a greedy policy w.r.t. $X_s w$.

We defer the proof to Section B.23. Theorem 9.3 requires μ_θ to be sufficiently smooth, which is a standard assumption even in the on-policy setting (e.g., Melo et al. (2008); Zou et al. (2019)). It is easy to see that if (2.17) with linear function approximation converged, it converged to a solution of

$$\bar{A}_{\pi_w,\mu_w} w + \bar{b}_{\mu_w} = 0,$$

which we call a TD fixed point for control in the average reward setting. Theorem 9.3, which shows that Algorithm 11 finds a regularized TD fixed point, is to the best of our knowledge the first theoretical study for linear Q-learning in the average-reward setting.

9.4 Empirical Results

In this section, we empirically study Algorithm 9 as a representative of the target-network-based algorithms introduced in this chapter.

We still use Baird's counterexample (Figure 3.1) as our testbed. We construct the state-action feature in the same way as the Errata of Baird (1995), i.e.,

$$X \doteq \begin{bmatrix} \begin{bmatrix} 2I & \mathbf{0} & \mathbf{1} \\ \mathbf{0}^\top & 1 & 2 \end{bmatrix} & \mathbf{0}^\top \mathbf{0} \\ \mathbf{0}^\top \mathbf{0} & I \end{bmatrix} \in \mathbb{R}^{14\times 15}.$$

The first 7 rows of X are the features of the solid action; the second 7 rows are the dashed action. The weight w is initialized as $[1, 1, 1, 1, 1, 1, 10, 1, 1, 1, 1, 1, 1, 1]^\top$. The standard linear Q-learning (with ridge regularization) updates w_t as

$$w_{t+1} \leftarrow w_t + \alpha(R_{t+1} + \max_{a'} x(S_{t+1}, a)^\top w_t - x_t^\top w_t)x_t - \alpha\eta w_t, \tag{9.5}$$

where we set $\alpha = 0.01$. The variant of Algorithm 9 we use in this experiment updates w_t as

$$\begin{aligned} w_{t+1} &\leftarrow w_t + \alpha(R_{t+1} + \max_{a'} x(S_{t+1}, a)^\top \theta_t - x_t^\top w_t)x_t - \alpha\eta w_t, \\ \theta_{t+1} &\leftarrow \theta_t + \beta(w_t - \theta_t), \end{aligned} \tag{9.6}$$

where we set $\alpha = 0.01, \beta = 0.001, \theta_0 = w_0$. We consider both a fixed behavior policy and a changing behavior policy. The results are reported in Figures 9.1 and 9.2 respectively. For Figure 9.1, the behavior policy is the same as the one used in Section 3.1, which we refer to as μ_0. For Figure 9.2, the behavior policy is $0.9\mu_0 + 0.1\mu_w$, where μ_w is a softmax policy w.r.t. $X_s w$. For our algorithm with a target network, the softmax policy is computed using the target network as shown in Algorithm 9. The curves are averaged over 30 independent runs with shaded regions indicating one standard deviation. The curves marked with "standard" and "ours" correspond to the updates (9.5) and (9.6) respectively.

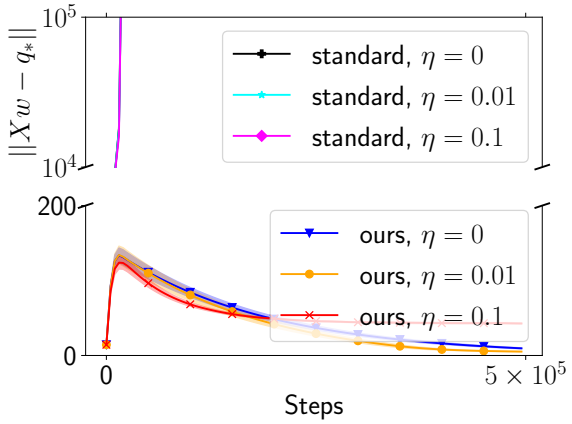

Figure 9.1: Linear Q-learning and its target network variant in Barid's counterexample with a fixed behavior policy.

Similar to what we have observed in Section 3.1, Figures 9.1 and 9.2 show that even with $\eta = 0$, i.e., no ridge regularization, our algorithms with target networks still converge in the tested domains. By contrast, without a target network, even when mild regularization is imposed, standard off-policy algorithms still diverge. This confirms the importance of the target network.

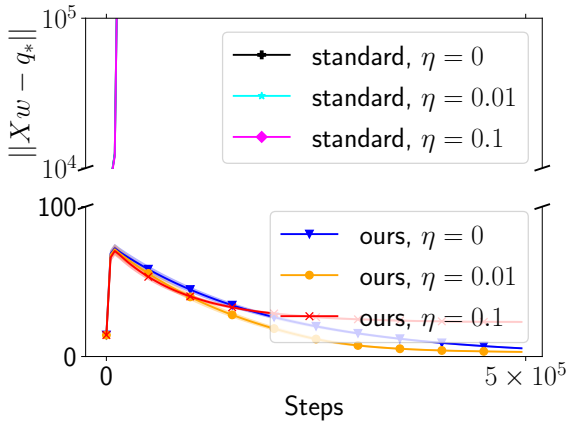

Figure 9.2: Linear Q-learning and its target network variant in Barid's counterexample with a changing behavior policy.

9.5 Discussion about Target Networks

For all the target-network-based algorithms we propose in Chapters 3, 7 and this chapter, both the target network and the ridge regularization are at play. One may wonder if it is possible to ensure convergence with only ridge regularization without the target network. In the prediction setting, the answer is affirmative. Applying ridge regularization in off-policy linear TD (2.18) directly yields

$$w_{t+1} \doteq w_t + \alpha_t \rho_t (R_{t+1} + \gamma x_{t+1}^\top w_t - x_t^\top w_t) x_t - \alpha_t \eta w_t. \tag{9.7}$$

The expected update of (9.7) is

$$\begin{aligned} \delta_w &\doteq b + (A - \eta I) w \\ &= b - X^\top D_\mu X w + \gamma X^\top D_\mu (P_\pi X w) - \eta w. \end{aligned}$$

If its Jacobian w.r.t. w, denoted as $\nabla_w \delta_w$, is negative definite, the convergence of $\{w_t\}$ is expected (see, e.g., Section 5.5 of Vidyasagar (2002)). This negative definiteness can be easily achieved by ensuring

$$\eta > \|X\|^2 \|D_\mu (I - \gamma P_\pi)\|,$$

see Diddigi et al. (2020) for similar techniques. This direct ridge regularization, however, would not work in the control setting. Consider, for example, linear Q-learning with ridge regularization (9.5). The Jacobian of its expected update is $\nabla_w (b_{\mu_w} + (A_{\pi_w, \mu_w} - \eta I) w)$. It is, however, not clear how to ensure this Jacobian to be negative definite by tuning η. By using a target network for bootstrapping, $P_\pi X w$

becomes $P_\pi X\theta$. So $\nabla_w \delta_w$ becomes $-\nabla_w(X^\top D_\mu X w + \eta w)$, which is always negative definite. Similarly, $\nabla_w(b_{\mu_w} - (A_{\pi_w,\mu_w} + \eta)w)$ becomes $-\nabla_w(X^\top D_{\mu_\theta} X w + \eta w)$ in Algorithm 9, which is always negative definite regardless of θ. The convergence of the main network $\{w_t\}$ can, therefore, be expected. The convergence of the target network $\{\theta_t\}$ is then delegated to Theorem 3.1. Now it is clear that in the deadly triad setting with linear function approximation, the target network stabilizes training by ensuring the Jacobian of the expected update to be negative definite.

One may also wonder if it is possible to ensure convergence with only the target network without ridge regularization. The answer is unclear. In our analysis, the conditions on $\|X\|$ (or equivalently, η) are only sufficient and not necessarily necessary. We do see in Figures 3.2, 9.1, and 9.2 that even with $\eta = 0$, our algorithms still converge in the tested domains. How small η can be in general and under what circumstances η can be 0 are still open problems, which we leave for future work.

In the prediction settings in Chapters 3 and 7, the introduction of the ridge regularization results in performance guarantees for the points that the proposed algorithms converge to. In the control settings in this chapter, we, however do not have any performance guarantee even with ridge regularization. This lack of optimality is one major limit of this thesis and we refer the reader to Section 14 for more discussion. Consequently, for practitioners, we recommend to treat η as an additional hyperparameter and tune η using standard hyperparameter tuning techniques to maximize the performance when computational budget is allowed. Otherwise we recommend to prioritize small η to minimize the introduced bias. The radius R_{B_1} and R_{B_2} are also new hyperparameters. The theoretical results prescribe that the radius should be larger than some threshold, which is usually problem-dependent and unknown. We, therefore, recommend practitioners to not use projection (i.e., the radius is infinite) unless divergence occurs.

Chapter 10

Control with Bidirectional Target Networks

Residual algorithms are a family of RL algorithms that address the deadly triad in limited settings and is an active research area before deep networks are widely used for function approximation in RL. In this chapter, we revisit residual algorithms and design a novel bidirectional target network, which gives rise to a few successful empirical applications of residual algorithms in deep RL settings.

10.1 Residual Gradients and Temporal Difference Learning

As discussed in Chapter 2, residual gradient algorithms use stochastic gradient descent to minimize the MSBE objective. This is in contrast to temporal difference methods, which use stochastic semi-gradient descent and minimizes the MSPBE objective. In this section, we review conventional wisdom about residual gradients (RG) and TD. We start with comparing their objectives MSBE and MSPBE. To this end, we first recall that in the on-policy setting, we have

$$\begin{aligned}\text{MSBE}(w) &\doteq \|\mathcal{T}_\pi Xw - Xw\|^2_{d_\pi},\\ \text{MSPBE}(w) &\doteq \|\Pi_{d_\pi}\mathcal{T}_\pi Xw - Xw\|^2_{d_\pi}.\end{aligned}$$

We further define the *Mean Squared Value Error* (MSVE) as

$$\text{MSVE}(w) \doteq \|v_\pi - Xw\|^2_{d_\pi},$$

which is the ultimate objective we want to minimize in the prediction problems. MSBE is considered inferior to MSPBE in the following sense.

- Sutton and Barto (2018) argue that MSBE is not learnable. Sutton and Barto (2018) show that different MDPs may have different distributions of sampled transitions due to state aliasing but the minima of MSBE can still be different. This questions the learnability of MSBE as sampled transitions are all that is available in model-free RL where we do not have access to the transition function p. By contrast, the minima of MSPBE are uniquely determined by the distribution of sampled transitions.

- Empirically, optimizing MSBE can lead to unsatisfying solutions. For example, in the A-presplit example (Sutton and Barto, 2018), the value of most states can be represented accurately by the function approximator but the MSBE minimizer does not do so, while the MSPBE minimizer does. Furthermore, empirically the MSBE minimizer can be further from the MSVE minimizer than the MSPBE minimizer (Dann et al., 2014).

MSBE is also considered superior to MSPBE in the following sense.

- Williams and Baird (1993) show MSBE can be used to bound MSVE (up to a constant). By contrast, at a point where MSPBE is minimized, MSVE can be arbitrarily large due to the capacity limit of the linear architecture (Bertsekas and Tsitsiklis, 1996).

- MSBE is an upper bound of MSPBE (Scherrer, 2010), indicating that optimizing MSBE implicitly optimizes MSPBE.

We then proceed to comparing RG and TD. RG is considered inferior to TD in the following sense.

- Due to the double sampling issue, it is usually hard to apply RG if the transition function is stochastic, while TD is compatible with both deterministic and stochastic transition functions.

- RG is usually slower than TD. Empirically, this is observed by Baird (1995), van Hasselt (2011), Gordon (1995) and Gordon (1999). Theoretically, Schoknecht and Merke (2003) prove TD converges faster than RG in a tabular setting.

- Lagoudakis and Parr (2003) argue that TD usually provides a better solution than RG, even though the value function is not as well approximated. The TD solution "preserves the shape of the value function to some extent rather than trying to fit the absolute values". Thus "the improved policy from the

corresponding approximate value function is closer to the improved policy from the exact value function" (Lagoudakis and Parr, 2003; Li, 2008; Sun and Bagnell, 2015).

RG is also considered superior to TD in the following sense.

- RG is a true gradient algorithm and enjoys convergence guarantees in most settings under mild conditions. By contrast, the divergence of TD with off-policy learning or nonlinear function approximation is well documented (Tsitsiklis and Roy, 1996a). Empirically, Munos (2003) and Li (2008) show that RG is more stable than TD.

- Schoknecht and Merke (2003) observe that RG converges faster than TD in the four-room domain (Sutton et al., 1999b) with linear function approximation. Scherrer (2010) shows empirically that the TD solution is usually slightly better than RG but in some cases fails dramatically.

Moreover, Li (2008) proves that TD makes more accurate predictions (i.e., the predicted state value is close to the true state value), while RG yields smaller temporal differences (i.e., the value predictions for a state and its successor are more consistent).

To summarize, previous insights about RG and TD, as well as their objectives, are mixed. TD has enjoyed great success in deep RL problems. There is, however, little study for RG in deep RL problems, which motivates our empirical investigation in this chapter.

10.2 Backward and Forward Bootstrapping

Since residual gradients suffer from the double sampling issue, we in this section focus on the setting where we aim to learn deterministic policies for tasks with deterministic transition functions. In this setting, the double sampling issue naturally disappears. In particular, we focus on a family of simulated robot manipulation tasks in MuJoCo (Todorov et al., 2012) and DeepMind Control Suite (DMControl, Tassa et al. (2018)).

One effective method for those robot manipulation tasks is Deep Deterministic Policy Gradient (DDPG, Lillicrap et al. (2016)). DDPG considers a deterministic policy $\pi_\theta : \mathcal{S} \to \mathbb{R}^{N_a}$, parameterized by θ, for those robot manipulation tasks, the action spaces of which are typically N_a-dimensional vectors. Let $q_w(s, a)$ parameterized by w be the estimation of the action value function. In the Markovian setting in

Definition 2.1, DDPG updates θ and w iteratively as

$$w_{t+1} \doteq w_t + \alpha_t \left(R_{t+1} + \gamma q_{\bar{w}_t}(S_{t+1}, \pi_{\bar{\theta}_t}(S_{t+1})) - q_{w_t}(S_t, A_t)\right) \nabla_w q_{w_t}(S_t, A_t),$$
$$\theta_{t+1} \doteq \theta_t + \beta_t \nabla_a q_{w_t}(S_t, a)|_{a=\pi_{\theta_t}(S_t)} \nabla_\theta \pi_{\theta_t}(S_t),$$

where $\bar{w}_t, \bar{\theta}_t$ are the target networks of w_t, θ_t respectively and the behavior policy μ_t is typically $\pi(S_t)$ with some random noise. Both target networks are updated similarly to (2.34). If one applies TD directly for learning the action value without a target network, one would update w as

$$\begin{aligned} w_{t+1} \doteq & w_t + \alpha_t \left(R_{t+1} + \gamma q_{w_t}(S_{t+1}, \pi_{\theta_t}(S_{t+1})) - q_{w_t}(S_t, A_t)\right) \\ & \times \nabla_w q_{w_t}(S_t, A_t). \end{aligned} \tag{10.1}$$

If one applies RG directly for learning the action value, one would update w as

$$\begin{aligned} w_{t+1} \doteq & w_t - \alpha_t \left(R_{t+1} + \gamma q_{w_t}(S_{t+1}, \pi_{\theta_t}(S_{t+1})) - q_{w_t}(S_t, A_t)\right) \\ & \times \left(\gamma \nabla_w q_{w_t}(S_{t+1}, \pi_{\theta_t}(S_{t+1})) - \nabla_w q_{w_t}(S_t, A_t)\right). \end{aligned} \tag{10.2}$$

To take the advantages of both TD and RG, Baird (1995) uses an additional hyper-parameter $\eta \in [0, 1]$ to unify (10.1) and (10.2) as

$$\begin{aligned} w_{t+1} \doteq & w_t - \alpha_t \left(R_{t+1} + \gamma q_{w_t}(S_{t+1}, \pi_{\theta_t}(S_{t+1})) - q_{w_t}(S_t, A_t)\right) \\ & \times \left(\gamma\eta \nabla_w q_{w_t}(S_{t+1}, \pi_{\theta_t}(S_{t+1})) - \nabla_w q_{w_t}(S_t, A_t)\right). \end{aligned} \tag{10.3}$$

When $\eta = 0$, (10.3) recovers (10.1); when $\eta = 1$, (10.3) recovers (10.2). It, however, remains an open problem that whether and how we should use target networks in (10.3).

In semi-gradient algorithms like (10.1), value propagation goes backwards in time. The value estimate of a state depends on the value estimate of its successor through bootstrapping, and a target network is usually used to stabilize this bootstrapping. Residual gradients like (10.3) instead allow value propagation both forwards and backwards. The value estimate of a state depends on the value estimate of both its successor and predecessor. Therefore, we need to stabilize the bootstrapping in both directions. To this end, we propose the *bidirectional target network* technique. Employing this in DDPG yields Bi-Res-DDPG, which updates w as

$$\begin{aligned} w_{t+1} \doteq & w_t - \alpha_t \left(R_{t+1} + \gamma q_{\bar{w}_t}(S_{t+1}, \pi_{\bar{\theta}_t}(S_{t+1})) - q_{w_t}(S_t, A_t)\right) \\ & \times \left(-\nabla_w q_{w_t}(S_t, A_t)\right) \\ & - \alpha_t \left(R_{t+1} + \gamma q_{w_t}(S_{t+1}, \pi_{\theta_t}(S_{t+1})) - q_{\bar{w}_t}(S_t, A_t)\right) \\ & \times \gamma\eta \nabla_w q_{w_t}(S_{t+1}, \pi_{\theta_t}(S_{t+1})). \end{aligned}$$

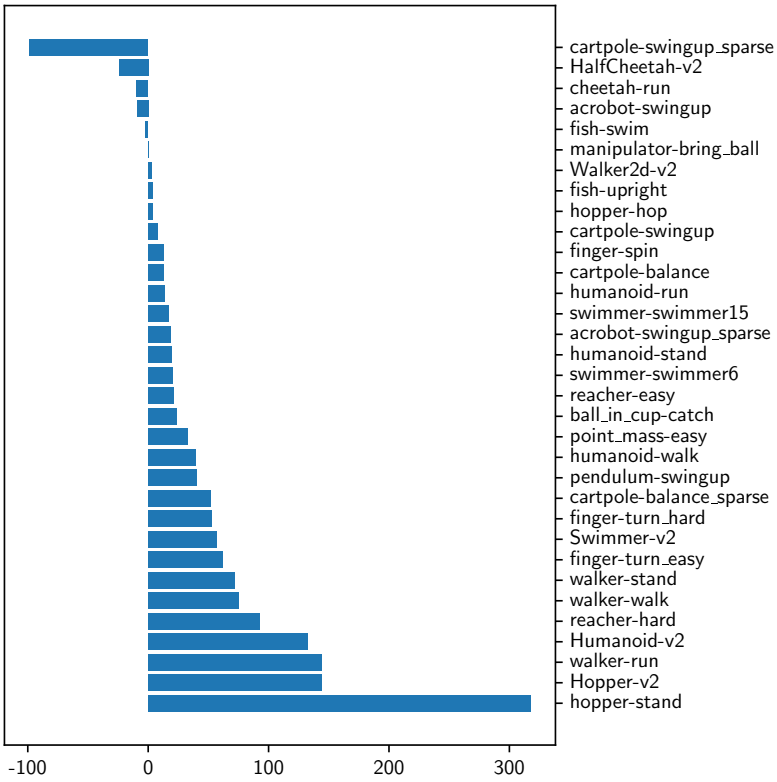

Figure 10.1: AUC improvements of Bi-Res-DDPG over DDPG on 28 DMControl tasks and 5 Mujoco tasks, computed as $\frac{\text{AUC}_{\text{Bi-Res-DDPG}} - \text{AUC}_{\text{DDPG}}}{\text{AUC}_{\text{DDPG}}}$.

The update to θ remains unchanged.

We compared Bi-Res-DDPG to DDPG in 28 DMControl tasks and 5 Mujoco tasks. Our DDPG implementation uses the same architecture and hyperparameters as Lillicrap et al. (2016), which are inherited by Bi-Res-DDPG (and all other DDPG variants in this chapter). For Bi-Res-DDPG, we tune η over $\{0, 0.05, 0.1, 0.2, 0.4, 0.8, 1\}$ on `walker-stand` and use $\eta = 0.05$ across all tasks. We perform 20 deterministic evaluation episodes every 10^4 training steps and generate the evaluation curves over 5 independent runs. We report the improvement of AUC (area under the curve) of the evaluation curves in Figure 10.1. AUC serves as a proxy for learning speed (e.g., see Example 8.2 in Sutton and Barto (2018)). Bi-Res-DDPG achieves a 20% (41%) AUC improvement over the original DDPG in terms of the median (mean). Our DDPG baseline reaches the same performance level as the DDPG baseline in Fujimoto et al. (2018) and Buckman et al. (2018) in Mujoco tasks.

To further investigate how target networks affect residual gradients, we study several variants of DDPG. Those variants have the same generic form that updates

w as

$$w_{t+1} \doteq w_t + \alpha_t \left(R_{t+1} + \Delta\right) \left(\gamma\eta \nabla_w q_{w_t}(S_{t+1}, \pi_{\theta_t}(S_{t+1})) - \nabla_w q_{w_t}(S_t, A_t)\right)$$

but differ in how Δ is computed. We use "T" and "O" to denote the target network and the online network respectively and have the following variants

$$\begin{aligned}
\text{Res-DDPG: } \Delta &\doteq \gamma q_{w_t}(S_{t+1}, \pi_{\theta_t}(S_{t+1})) - q_{w_t}(S_t, A_t), \\
\text{TO-Res-DDPG: } \Delta &\doteq \gamma q_{\bar{w}_t}(S_{t+1}, \pi_{\bar{\theta}_t}(S_{t+1})) - q_{w_t}(S_t, A_t), \\
\text{OT-Res-DDPG: } \Delta &\doteq \gamma q_{w_t}(S_{t+1}, \pi_{\theta_t}(S_{t+1})) - q_{\bar{w}_t}(S_t, A_t), \\
\text{TT-Res-DDPG: } \Delta &\doteq \gamma q_{\bar{w}_t}(S_{t+1}, \pi_{\bar{\theta}_t}(S_{t+1})) - q_{\bar{w}_t}(S_t, A_t).
\end{aligned}$$

Res-DDPG is a direct combination of residual gradient and DDPG without a target network. TO-Res-DDPG simply adds a residual gradient term to the original DDPG. OT-Res-DDPG stabilizes the bootstrapping for the forward value propagation. TT-Res-DDPG stabilizes bootstrapping in both directions but destroys the connection between prediction and error. By contrast, Bi-Res-DDPG stabilizes bootstrapping in both directions and maintains the connection between prediction and error.

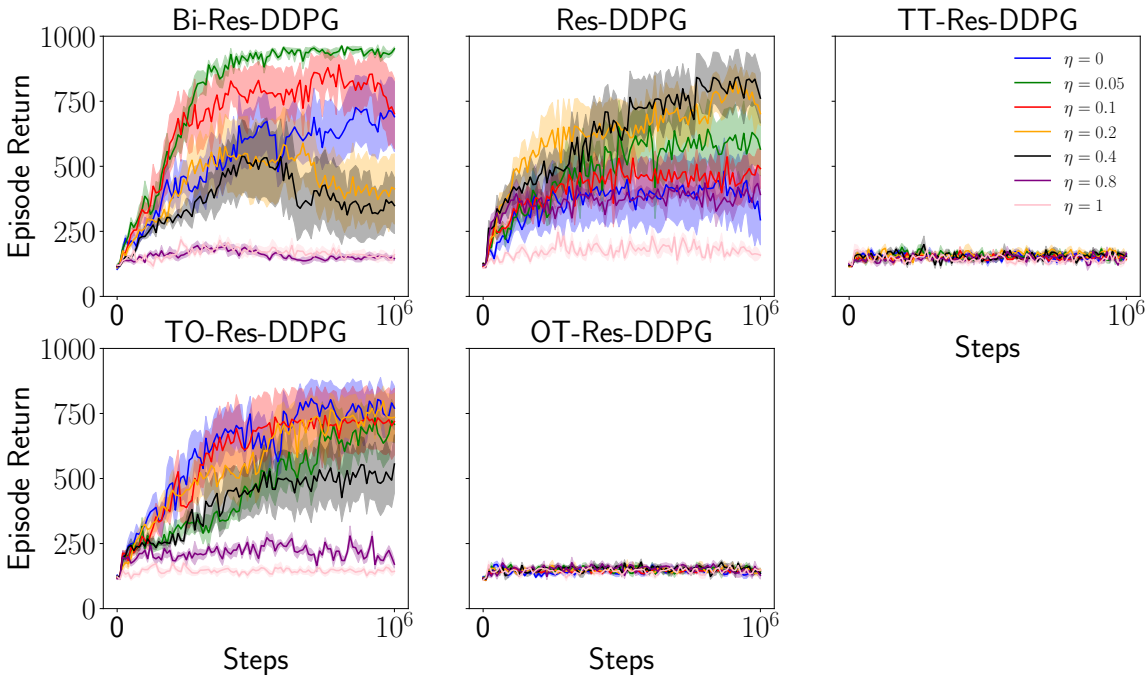

Figure 10.2: Performance of Bi-Res-DDPG variants on `walker-stand`, focusing on the role of target networks.

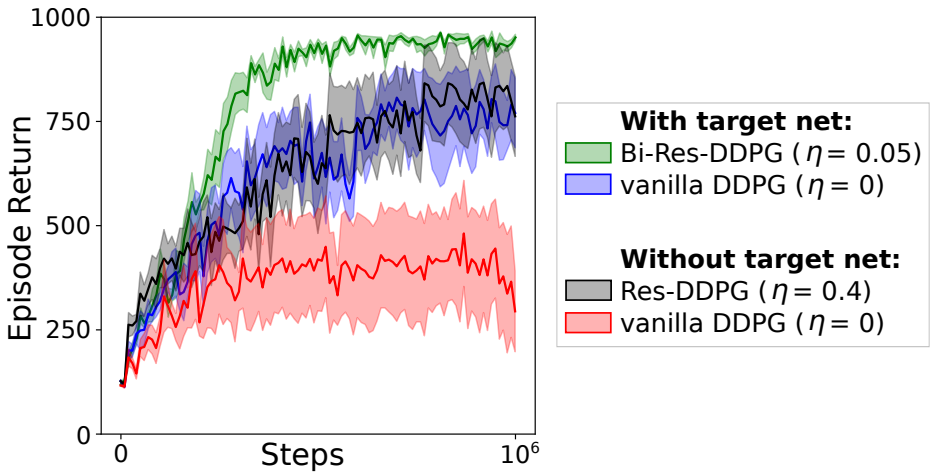

Figure 10.3: A selection of the best parameters η from Figure 10.2. Note that residual updates stabilize performance as much as the introduction of target networks.

Figure 10.2 compares these variants on `walker-stand`. The main points to note are: (1) Both Bi-Res-DDPG($\eta = 0$) and TO-Res-DDPG($\eta = 0$) are the same as vanilla DDPG. The curves are similar, verifying the stability of our implementation. (2) Res-DDPG($\eta = 0$) corresponds to vanilla DDPG without a target network, which performs poorly. This confirms that a target network is important for stabilizing training and mitigating divergence when a nonlinear function approximator is used (Mnih et al., 2015; Lillicrap et al., 2016). (3) Increasing η moderately improves Res-DDPG's performance. This complies with the argument from Baird (1995) that residual gradients help semi-gradients converge. All variants fail with a large η (e.g., 0.8 or 1). This complies with the argument from Baird (1995) that pure residual gradients are slow. (4) TO-Res-DDPG($\eta = 0$) (i.e., vanilla DDPG) is similar to Res-DDPG($\eta = 0.4$), indicating a naive combination of residual gradients and DDPG without a target network is ineffective. (5) For TO-Res-DDPG, $\eta = 0$ achieves the best performance, indicating adding a residual gradient term to DDPG directly is ineffective. To summarize, these variants confirm the necessity of the bidirectional target network. To better understand the role of residual updates, we summarize the results of Figure 10.2 in Figure 10.3. Res-DDPG does not have a target network and outperforms DDPG without a target network. Res-DDPG also increases the stability. Bi-Res-DDPG has target networks and also outperforms DDPG with a target network, as well as increases the stability. This comparison confirms the importance of residual updates.

10.3 Discussion

We also evaluated a Bi-Res version of DQN in three Arcade Learning Environments (ALE, Bellemare et al. (2013a)). The performance was similar to the original DQN. One of the many differences between DMControl and ALE is that rewards in ALE are much more sparse. This might indicate that the forward value propagation in residual gradients is less likely to yield a performance boost with sparse rewards.

We do not expect residual gradients to improve the performance of semi-gradient algorithms in all tasks. However, our results do show that the residual gradients together with the bidirectional target network is beneficial in many settings. Despite the popularity of semi-gradient methods, we do believe residual gradients have long been underestimated and deserve more study by the community.

There are of course many other studies on residual methods. We name a few in the following. Geist et al. (2017) show that for policy-based methods, maximizing the average reward is better than minimizing the Bellman residual. Schoknecht and Merke (2002) show RG converges with a problem-dependent constant learning rate when combined with certain function approximators. Dabney and Thomas (2014) extend RG with natural gradients. However, this chapter appears to be the first to contrast residual gradients and semi-gradients in deep RL problems and demonstrate the benefits of residual gradients in deep RL .

Chapter 11

Control with Truncated Followon Traces

In this chapter, we extend the use of truncated followon traces from prediction in Chapter 5 to control.

11.1 Emphatic Approximate Value Iteration

The study of the canonical approximate value iteration (De Farias and Van Roy, 2000) is essential to the study of the on-policy control algorithm SARSA (Melo et al., 2008; Zou et al., 2019). Similarly, in this section, we study approximate value iteration from an off-policy perspective, which prepares us for the off-policy control algorithm in the next section.

In dynamic programming, one classical method for finding v_* is value iteration, which applies the optimal Bellman operator (2.8) repeatedly. For a vector $v \in \mathbb{R}^{|\mathcal{S}|}$, let π_v denote a greedy policy w.r.t. $r(s,a) + \sum_{s'} p(s'|s,a)v(s')$. Then value iteration can be regarded as applying the Bellman operator $\mathcal{T}_\pi$ with $\pi = \pi_v$ to v. Or more precisely, at the k-th iteration, we have value estimation v_k. We then compute the value estimation at the next iteration as

$$v_{k+1} \doteq \mathcal{T}_{\pi_{v_k}} v_k.$$

When linear function approximation is considered, v_k is represented as Xw_k. However, the new value $\mathcal{T}_{\pi_{v_k}} v_k$ may not lie in the column space of X. To this end, we use an additional projection operator to compute w_{k+1} such that

$$Xw_{k+1} = \Pi_{d_{\pi_{Xw_k}}} \mathcal{T}_{\pi_{Xw_k}} Xw_k,$$

or equivalently,

$$v_{k+1} = \Pi_{d_{\pi_{v_k}}} \mathcal{T}_{\pi_{v_k}} v_k.$$

This scheme for computing the sequence $\{v_k\}$, or $\{w_k\}$, is referred to as *approximate value iteration* (De Farias and Van Roy, 2000). Unfortunately, if π_v is the aforementioned greedy policy, the approximate value iteration operator

$$\mathcal{H}(v) \doteq \Pi_{d_{\pi_v}} \mathcal{T}_{\pi_v} v$$

does not necessarily have a fixed point. However, if π_v is continuous in v, De Farias and Van Roy (2000) show that $\mathcal{H}$ has at least one fixed point. SARSA with linear function approximation is an incremental and stochastic way for implementing $\mathcal{H}$.

The canonical approximate value iteration operator $\mathcal{H}$ is in a sense on-policy in that the projection operator is defined w.r.t. a norm induced by the policy at the current iteration. To develop control algorithms for the off-policy setting, we in this section study approximation value iteration from an off-policy perspective, i.e., with a projection operator defined w.r.t. a different norm. Let π_w and μ_w be target and behavior policies respectively. They depend on w, the parameters used for estimating the value function, through the value function estimate $v = Xw \in \mathbb{R}^{|\mathcal{S}|}$, e.g., they can be softmax policies such as

$$\pi_w(a|s) \doteq \frac{\exp\left(\left(r(s,a) + \gamma \sum_{s'} p(s'|s,a) x(s')^\top w\right)/\iota\right)}{\sum_{s_0,a_0} \exp\left(\left(r(s_0,a_0) + \gamma \sum_{s_1} p(s_1|s_0,a_0) x(s_1)^\top w\right)/\iota\right)},$$

with different temperatures ι. We consider the iterates $\{v_k \doteq Xw_k\}$ generated by

$$v_{k+1} \doteq \Pi_{f_{\pi_{w_k},\mu_{w_k},n}} \mathcal{T}_{\pi_{w_k}} v_k,$$

where $f_{\pi_w,\mu_w,n}$ is defined in (5.3). We call this scheme *emphatic approximate value iteration* as the projection operator is defined w.r.t. the norm induced by the (truncated) followon trace. In the rest of this section, we show that emphatic approximate value iteration adopts at least one fixed point. With Λ_μ denoting the closure of $\left\{\mu_w \mid w \in \mathbb{R}^K\right\}$ and Λ_π denoting the closure of $\left\{\pi_w \mid w \in \mathbb{R}^K\right\}$, we make the following assumptions.

Assumption 11.1. *Both π_w and μ_w are continuous in w.*

Assumption 11.2. *For any $\mu \in \Lambda_\mu$, the Markov chain induced by μ is ergodic and $\mu(a|s) > 0$ holds for all (s,a).*

Assumption 11.1 appears to be necessary in analyzing approximate value iteration. If π_w is not continuous in w, even the canonical approximate value iteration can fail to have a fixed point (De Farias and Van Roy, 2000). The ergodicity assumption of all the policies in the closure in Assumption 11.2 is similar to Assumption 9.1.

We now define two helper functions to understand how n should be selected in emphatic approximate value iteration.

$$n_1(\pi, \mu) \doteq \frac{\ln(\lambda_{min,\pi,\mu} d_{\mu,min}) - \ln\left(d_{\mu,max}^2 \|\gamma P_\pi - I\| \|m_{\pi,\mu}\|_1\right)}{\ln \gamma} - 1,$$
$$n_2(\pi, \mu) \doteq \frac{\ln\left(\kappa_{\pi,\mu} d_{\mu,min} \min_s i(s) d_\mu(s)\right) - \ln\left(d_{\mu,max}^2 \left\|I - \gamma P_\pi^\top\right\|_\infty \|m_{\pi,\mu}\|_1\right)}{\ln \gamma} - 1.$$

Here we write κ defined in Lemma 5.6 as $\kappa_{\pi,\mu}$ to explicitly acknowledge its dependence on π and μ. Similarly, $\lambda_{min,\pi,\mu}$ refers to λ_{min} defined in Lemma 5.3. It is easy to see that n_1 and n_2 correspond to the conditions of n in Lemmas 5.3 and 5.6 respectively. Assumption 11.2 ensures that n_1 and n_2 are well defined on $\Lambda_\pi \times \Lambda_\mu$. The invariant distribution d_μ is continuous in μ (see, e.g., Lemma C.3). The minimum eigenvalue $\lambda_{min,\pi,\mu}$ is continuous in the elements of the matrix (see, e.g., Corollary 8.6.2 of Golub and Loan 1996) and thus is also continuous in μ and π. And both Λ_μ and Λ_π are compact. We, therefore, have $\sup_{\mu \in \Lambda_\mu, \pi \in \Lambda_\pi} \max\{n_1(\pi, \mu), n_2(\pi, \mu)\} < \infty$ by the extreme value theorem. This allows us to select n as suggested by the following lemma.

Lemma 11.1. *Let Assumptions 2.3, 11.1, and 11.2 hold. If*

$$n > \sup_{\mu \in \Lambda_\mu, \pi \in \Lambda_\pi} \max\{n_1(\pi, \mu), n_2(\pi, \mu)\}, \tag{11.1}$$

then there exists at least one w_* such that

$$X w_* = \Pi_{f_{\pi_{w_*}, \mu_{w_*}, n}} \mathcal{T}_{\pi_{w_*}} X w_*.$$

The proof of Theorem 11.1 is provided in B.24, which follows the same steps of De Farias and Van Roy (2000) but generalizes their results from (on-policy) approximate value iteration to emphatic approximate value iteration.

11.2 Truncated Emphatic Expected SARSA

We now present our control algorithm, truncated emphatic expected SARSA. In this section we mainly work on the action value so we use the overloaded notations for

action value functions defined in Section 2.13. The followon trace F_t is now defined as

$$F_t \doteq i_t + \gamma \rho_t F_{t-1},$$

which is the same as the followon trace used in ELSTDQ(λ) in White (2017). Correspondingly, the truncated trace is defined as

$$F_{t,n} \doteq \begin{cases} \sum_{j=0}^{n} \gamma^j \rho_{t-j+1:t} i_{t-j} & t \geq n \\ F_t & t < n \end{cases}.$$

The truncated emphasis $m_{\pi,\mu,n}$ is now in $\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ and defined as

$$m_{\pi,\mu,n}(s,a) \doteq \lim_{t\to\infty} \mathbb{E}[F_{t,n}|S_t = s, A_t = a].$$

Other notation is also overloaded accordingly, e.g.,

$$m_{\pi,\mu} \doteq \lim_{n\to\infty} m_{\pi,\mu,n},\ f_{\pi,\mu,n} \doteq D_\mu m_{\pi,\mu,n},\ f_{\pi,\mu} \doteq D_\mu m_{\pi,\mu}.$$

Previous theoretical results also hold with the overloaded notations for state-action pairs. In particular, we have

Lemma 11.2. *Let Assumptions 2.3, 11.1, and 11.2 hold. Define*

$$n_1(\pi,\mu) \doteq \frac{\ln(\lambda_{min,\pi,\mu} d_{\mu,min}) - \ln\left(d_{\mu,max}^2 \|\gamma P_\pi - I\| \|m_{\pi,\mu}\|_1\right)}{\ln\gamma} - 1,$$
$$n_2(\pi,\mu) \doteq \frac{\ln\left(\kappa_{\pi,\mu} d_{\mu,min} \min_{s,a} i(s,a) d_\mu(s,a)\right) - \ln\left(d_{\mu,max}^2 \left\|I - \gamma P_\pi^\top\right\|_\infty \|m_{\pi,\mu}\|_1\right)}{\ln\gamma} - 1,$$

where $\lambda_{min,\pi,\mu}$ is the minimum eigenvalue of

$$\frac{1}{2}\left(D_{f_{\pi,\mu}}(I - \gamma P_\pi) + (I - \gamma P_\pi^\top) D_{f_{\pi,\mu}}\right),$$

$d_{\mu,min} \doteq \min_{s,a} d_\mu(s,a), d_{\mu,max} \doteq \max_{s,a} d_{\mu,max}(s,a), \kappa_{\pi,\mu} \doteq \min_{s,a} \frac{d_\mu(s,a) i(s,a)}{f_{\pi,\mu}(s,a)}$. *If*

$$n > \sup_{\mu\in\Lambda_\mu, \pi\in\Lambda_\pi} \max\left\{n_1(\pi,\mu), n_2(\pi,\mu)\right\}$$

holds, then

(i). For any $\mu \in \Lambda_\mu, \pi \in \Lambda_\pi$, $X^\top D_{f_{\pi,\mu,n}}(\gamma P_\pi - I)X$ is n.d.,

(ii). For any $\mu \in \Lambda_\mu, \pi \in \Lambda_\pi$, $\Pi_{f_{\pi,\mu,n}} \mathcal{T}_\pi$ is a $\sqrt{\gamma}$ contraction in $\|\cdot\|_{f_{\pi,\mu,n}}$,

(iii). There exists at least one w_* such that

$$Xw_* = \Pi_{f_{\pi_{w_*},\mu_{w_*},n}} \mathcal{T}_{\pi_{w_*}} Xw_*. \tag{11.2}$$

We use $\mathcal{W}_*$ to denote the set of all such w_*.

The proof of Lemma 11.2 is omitted since it is a verbatim repetition of the proofs of Lemmas 5.3, 5.6, and 11.1.

The iterative update scheme (11.2) is emphatic approximate value iteration applied to action-value estimation. To implement this scheme incrementally in a learning sense, we propose truncated emphatic expected SARSA (Algorithm 12). When computing $F_{t,n}$, we require that the previous importance sampling ratios be recomputed with the current weight w_t. This requirement is mainly for the ease of asymptotic analysis and is eliminated in projected truncated emphatic expected SARSA, for which we provide a finite sample analysis.

Algorithm 12: Truncated emphatic expected SARSA

$S_0 \sim p_0(\cdot)$
$A_0 \sim \mu_{w_0}(\cdot|S_0)$
$t \leftarrow 0$
while *True* **do**
  Execute A_t, get R_{t+1}, S_{t+1}
  $A_{t+1} \sim \mu_{w_t}(\cdot|S_{t+1})$
  $\rho_t \leftarrow \frac{\pi_{w_t}(A_t|S_t)}{\mu_{w_t}(A_t|S_t)}$
  $F_{t,n} \leftarrow 0$
  for $k = 0, \ldots, n$ **do**
    $F_{t,n} \leftarrow i_{t-n+k} + \gamma \frac{\pi_{w_t}(A_{t-n+k}|S_{t-n+k})}{\mu_{w_t}(A_{t-n+k}|S_{t-n+k})} F_{t,n}$
  end
  $w_{t+1} \leftarrow w_t + \alpha_t F_{t,n}(R_{t+1} + \gamma \sum_a \pi_{w_t}(a|S_{t+1}) x(S_{t+1}, a)^\top w_t - x_t^\top w_t) x_t$
  $t \leftarrow t + 1$
end

We can now present our asymptotic convergence analysis of Algorithm 12. We first study the properties of the possible fixed points. We can rewrite (11.2) as

$$A_{w_*} w_* + b_{w_*} = 0,$$

where

$$A_w \doteq X^\top D_{f_{\pi_w,\mu_w,n}} (\gamma P_{\pi_w} - I) X,$$
$$b_w \doteq X^\top D_{f_{\pi_w,\mu_w,n}} r.$$

Consequently,

$$w_* = A_{w_*}^{-1} b_{w_*}.$$

Since Λ_μ and Λ_π are compact, both π_w and μ_w are continuous in w, the RHS of the above equation is bounded from above by the extreme value theorem. Consequently, there exists a constant $R < \infty$ such that

$$\sup_{w_* \in \mathcal{W}_*} \|w_*\| \leq R.$$

We then make several regularization conditions on the policies π_w and μ_w. For the analysis of on-policy SARSA with linear function approximation, it is commonly assumed that the policy π_w is Lipschitz continuous in w and the Lipschitz constant is not too large (Perkins and Precup, 2002; Zou et al., 2019). This technical assumption is mainly used to ensure that a small change in the value estimate does not result in a big difference in the policy thus enforces certain smoothness of the overall learning process. Without such assumptions, even on-policy linear SARSA can chatter and fail to converge (Gordon, 1996, 2001). In this section, we adopt similar assumptions in our off-policy setting.

Assumption 11.3. *Both μ_w and π_w are Lipschitz continuous in w, i.e., there exist constants L_μ and L_π such that for any $s \in \mathcal{S}, a \in \mathcal{A}$,*

$$|\pi_w(a|s) - \pi_{w'}(a|s)| \leq L_\pi \|w - w'\|,$$
$$|\mu_w(a|s) - \mu_{w'}(a|s)| \leq L_\mu \|w - w'\|.$$

The Lipschitz continuity of the policies immediately implies the Lipschitz continuity of A_w and b_w governing the expected updates of Algorithm 12.

Lemma 11.3. *Let Assumptions 11.2 and 11.3 hold. There exist positive constants C_1, C_2, C_3, and C_4 such that for any w, w'*

$$\|A_w - A_{w'}\| \leq (C_1 L_\mu + C_2 L_\pi)\|w - w'\|,$$
$$\|b_w - b_{w'}\| \leq (C_3 L_\mu + C_4 L_\pi)\|w - w'\|.$$

The proof Lemma 11.3 is provided in Section B.25. Under the conditions of Lemma 11.2, for any w, the matrix

$$M(w) \doteq \frac{1}{2}\left(X^\top D_{f_{\pi_w,\mu_w,n}}(I - \gamma P_{\pi_w})X + X^\top (I - \gamma P_{\pi_w}^\top) D_{f_{\pi_w,\mu_w,n}} X\right)$$

is p.d. For a symmetric positive definite matrix M, let $\lambda(M)$ denote the smallest eigenvalue of M. For any $w \in \mathbb{R}^K$, we have $\lambda(M(w)) > 0$. By the continuity of eigenvalues in the elements of the matrix, the compactness of Λ_μ and Λ_π, and the extreme value theorem, we have

$$\inf_{w \in \mathbb{R}^K} \lambda(M(w)) > 0.$$

This allows us to make the following assumptions about the Lipschitz constants L_μ and L_π, akin to Perkins and Precup (2002); Zou et al. (2019).

Assumption 11.4. *L_μ and L_π are small enough such that*

$$\lambda'_{min} \doteq \inf_{w \in \mathcal{W}} \lambda(M(w)) - ((C_1 L_\mu + C_2 L_\pi)R + C_3 L_\mu + C_4 L_\pi) > 0.$$

With these regularizations on π_w and μ_w, we can now present a high probability asymptotic convergence analysis for Algorithm 12.

Theorem 11.4. *Let the assumptions and conditions in Lemma 11.2 hold. Let Assumptions 2.4, 11.3, and 11.4 hold. Then for any compact set $\mathcal{W} \subset \mathbb{R}^K$ and any $w \in \mathcal{W}$, there exists a constant $C_\mathcal{W}$ such that for any $w_* \in \mathcal{W}_*$, the iterates $\{w_t\}$ generated by Algorithm 12 satisfy*

$$\Pr\Big(\lim_{t \to \infty} w_t = w_* \mid w_0 = w\Big) \geq 1 - C_\mathcal{W} \sum_{t=0}^{\infty} \alpha_t^2. \tag{11.3}$$

This immediately implies that $\mathcal{W}_*$ contains only one element (under the conditions of this theorem).

In (11.3), $C_\mathcal{W}$ depends on the compact set $\mathcal{W}$ from which the weight w_0 is selected. For (11.3) to be nontrivial, the learning rates have to be small enough, depending on the choice of initial weights. The proof of Theorem 11.4 is provided in Section B.26 and depends on Theorem 13 of Benveniste et al. (1990).[1]

We now analyze the convergence rate of projected truncated emphatic expected SARSA (Algorithm 13). Unlike Algorithm 12, when computing $F_{t,n}$ in Algorithm 13, we do *not* need to recompute previous importance sampling ratios. Similar to Assumption 11.4, we make the following assumption about the Lipschitz constants L_μ and L_π for analyzing Algorithm 13.

[1] It might be possible to obtain an almost sure convergence of Algorithm 12 like Theorem 5.4 by invoking Theorem 17 of Benveniste et al. (1990). Doing so requires verifying (1.9.5) of Benveniste et al. (1990). If how Melo et al. (2008) verify (1.9.5) was documented in the context of on-policy SARSA with linear function approximation, it is expected that (1.9.5) can also be similarly verified in the context of Algorithm 12.

Algorithm 13: Projected truncated emphatic expected SARSA

Initialize w_0 such that $\|w_0\| \leq R$
$S_0 \sim p_0(\cdot)$
$A_0 \sim \mu_{w_0}(\cdot|S_0)$
$t \leftarrow 0$
while *True* **do**
  Execute A_t, get R_{t+1}, S_{t+1}
  $A_{t+1} \sim \mu_{w_t}(\cdot|S_{t+1})$
  $\rho_t \leftarrow \frac{\pi_{w_t}(A_t|S_t)}{\mu_{w_t}(A_t|S_t)}$
  $F_{t,n} \leftarrow 0$
  for $k = 0, \ldots, n$ **do**
    $F_{t,n} \leftarrow i_{t-n+k} + \gamma\rho_{t-n+k}F_{t,n}$
  end
  $w_{t+1} \leftarrow \Pi_R \left(w_t + \alpha_t F_{t,n}(R_{t+1} + \gamma \sum_a \pi_{w_t}(a|S_{t+1})x(S_{t+1}, a)^\top w_t - x_t^\top w_t)x_t\right)$
  $t \leftarrow t + 1$
end

Assumption 11.5. *L_μ and L_π are not too large such that*

$$\lambda''_{min} \doteq \inf_{w_* \in \mathcal{W}_*} \lambda(M(w_*)) - ((C_1 L_\mu + C_2 L_\pi)R + C_3 L_\mu + C_4 L_\pi) > 0.$$

When defining λ'_{min} in Assumption 11.4, the infimum is taken over all possible w. When defining λ''_{min} in Assumption 11.5, the infimum is taken over only $\mathcal{W}_*$. This improvement is made possible by the introduction of the projection Π_R.

Theorem 11.5. *Let the assumptions and conditions in Lemma 11.2 hold. Let Assumptions 11.3 and 11.5 hold. Set the learning rate $\{\alpha_t\}$ in Algorithm 12 to*

$$\alpha_t \doteq \frac{1}{2\alpha_\lambda(t+1)}, \tag{11.4}$$

where $\alpha_\lambda \in (0, \lambda''_{min})$ is some constant. Then for any $w_* \in \mathcal{W}_*$, for sufficiently large t (in the sense that $t - \mathcal{O}(\ln t) > n$), the iterates $\{w_t\}$ generated by Algorithm 13 satisfy

$$\mathbb{E}\left[\|w_t - w_*\|^2\right] = \mathcal{O}\left(\frac{\ln^3 t}{t}\right).$$

This immediately implies that $\mathcal{W}_*$ contains only one element (under the conditions of this theorem).

The proof of Theorem 11.5 and the constants hidden by $\mathcal{O}(\cdot)$ are detailed in Section B.27. The proof follows the same steps as Zou et al. (2019) but generalizes the analysis

of the on-policy SARSA in Zou et al. (2019) to the off-policy setting and includes backward traces, which are not included in Zou et al. (2019).

In this section, we present (projected) truncated emphatic expected SARSA as a convergent off-policy control algorithm with linear function approximation. Importantly, in Algorithms 12 and 13, the behavior policy is a function of the current action-value estimates and thus changes every time step and can be very different from the target policy. These two features are common in practice (see, e.g. Mnih et al. (2015)) but rarely appreciated in existing literature. For example, in Greedy-GQ (Maei et al., 2010; Wang and Zou, 2020), a control algorithm in the family of the gradient TD methods, the behavior policy is assumed to be fixed. In the convergent analysis of linear Q-learning (Melo et al., 2008; Lee and He, 2019b), the behavior policy is assumed to be sufficiently close to the policy that linear Q-learning is expected to converge to.

11.3 Empirical Results

In this section, we empirically investigate Algorithm 13 and its β-variant using (4.1). We first use Baird's counterexample (Figure 3.1) as our testbed. In particular, we consider two settings: control with a fixed behavior policy and control with a changing behavior policy. The hyperparameter tuning protocol and the reporting protocol are the same as that of Section 5.3.

In the control setting with a fixed behavior policy, we benchmark Algorithm 13 with different selection of n, as well as its β-variant (cf. (4.1)). In particular, we set the radius of the ball for projection to be infinity (i.e., the projection is now an identity mapping). Consequently, when $n = \infty$, our implementation of Algorithm 13 becomes a straightforward extension of ETD to the control setting. We use the same behavior policy as the prediction setting in Section 5.3. The target policy is a softmax policy with a temperature τ:

$$\pi(\texttt{dashed}|s) \doteq \frac{\exp\left(q(s, \texttt{dashed})/\tau\right)}{\exp\left(q(s, \texttt{dashed})/\tau\right) + \exp\left(q(s, \texttt{solid})/\tau\right)}.$$

We test three different temperatures $\tau \in \{0.01, 0.1, 1\}$. When τ approaches 0, the target policies become more and more greedy. Consequently, Algorithm 13 approaches Q-learning with truncated traces. As shown in Figure 11.1, neither the naive off-policy expected SARSA (i.e., $n = 0$) nor the naive extension of ETD(0) (i.e., $n = \infty$) makes any progress in this setting. By contrast, our truncated empathic expected SARSA consistently converges, with lower variance than its β-variant (Table 11.1).

	$n=\infty$	$n=0$	$n=2$	$n=4$	$n=8$	$\beta=0.8$
$\tau=0$	-	-	10^4	10^3	10^6	10^{11}
$\tau=0.01$	-	-	10^4	10^3	10^6	10^{11}
$\tau=0.1$	-	-	10^4	10^3	10^6	10^{11}

Table 11.1: Average variance of curves in Figure 11.1. Here $n=4$ has smaller variance than $n=2$ because the former converges slightly faster. We follow a similar reporting protocol as Table 5.1.

	$n=\infty$	$n=0$	$n=2$	$n=4$	$n=8$	$\beta=0.8$
$\tau=0$	-	-	10^3	10^2	10^6	10^6
$\tau=0.01$	-	-	10^3	10^2	10^6	10^6
$\tau=0.1$	-	-	10^3	10^2	10^6	10^6

Table 11.2: Average variance of curves in Figure 11.2. Here $n=4$ has smaller variance than $n=2$ because the former converges slightly faster. We follow a similar reporting protocol as Table 5.1.

In the control setting with a changing behavior policy, we still benchmark Algorithm 13 with a different selection of n and its β-variant. The target policy is still the softmax policy with a temperature τ. The behavior policy is now a mixture policy. At each time step, with probability 0.9, the behavior policy is the same as the behavior policy used in the prediction setting in Section 5.3; with probability 0.1, the behavior policy is a softmax policy with temperature 1. As shown by Figure 11.2 and Table 11.2, the results in this setting are similar to the previous setting with a fixed behavior policy but the variance with $n \in \{2,4\}$ is reduced. This is because the behavior policy is now related to the target policy, i.e., the off-policyness is reduced.

We further evaluate truncated emphatic expected SARSA in the CartPole domain (Figure 11.3), which is a classical nonsynthetic control problem. We use tile coding (Sutton, 1995) to map the four-dimensional observation (velocity, acceleration, angular velocity, angular acceleration) to a binary vector in $\mathbb{R}^{1024}$ and then apply linear function approximation. In particular, we use the tile coding software recommended in Chapter 10.1 of Sutton and Barto (2018). We benchmark Algorithm 13 and its β-variant (cf. (4.1)), following the same hyperparameter tuning protocol as in Baird's counterexample. We use $\gamma=0.99$ and $i(s)=1$. The target policy is a softmax policy with temperature $\tau=0.01$. The behavior policy is a ϵ-softmax policy with $\epsilon=0.95$ and $\tau=1$. In other words, at each time step, with probability 0.95, the agent selects an action according to a uniformly random policy; with probability 0.05, the agent

selects an action according to a softmax policy with temperature $\tau = 1$. We grant large randomness to the behavior policy to enlarge the off-policyness of the problem, making it more challenging. We evaluate the agent every 5×10^3 steps during the training process for 10 episodes and report the averaged undiscounted episodic return. Figure 11.4 (Left) investigates the effect of different truncation length. We recall that the learning rate α is tuned from Λ_α maximizing the evaluation return at the end of the training. With $n = \infty$ (i.e., no truncation), the agent barely learns anything. With $n = 0$ (i.e., naive off-policy expected SARSA without followon trace), the agent reaches a reasonable performance level but using $n = 4$ performs better. Using $n = 2$ performs better than using $n = 4$ in the middle of the training but the performance drops near the end of the training. We conjecture that this may suggest that a truncation length of 2 is not enough to stabilize the off-policy training in the tested problem. Figure 11.4 (Right) further investigates the soft truncation using (4.1). We recall that β is tuned from $\{0.1, 0.2, 0.4, 0.8\}$. Using the soft truncation with $\beta = 0.2$ performs similar to using the hard truncation with $n = 4$. It can, however, be computed that the data points of the curve with $\beta = 0.2$ has an average variance around 1.8×10^4 while that of $n = 4$ is around 7×10^3. This suggests that our proposed hard truncation might be a better option for variance reduction than the existing soft truncation for the tested problem.

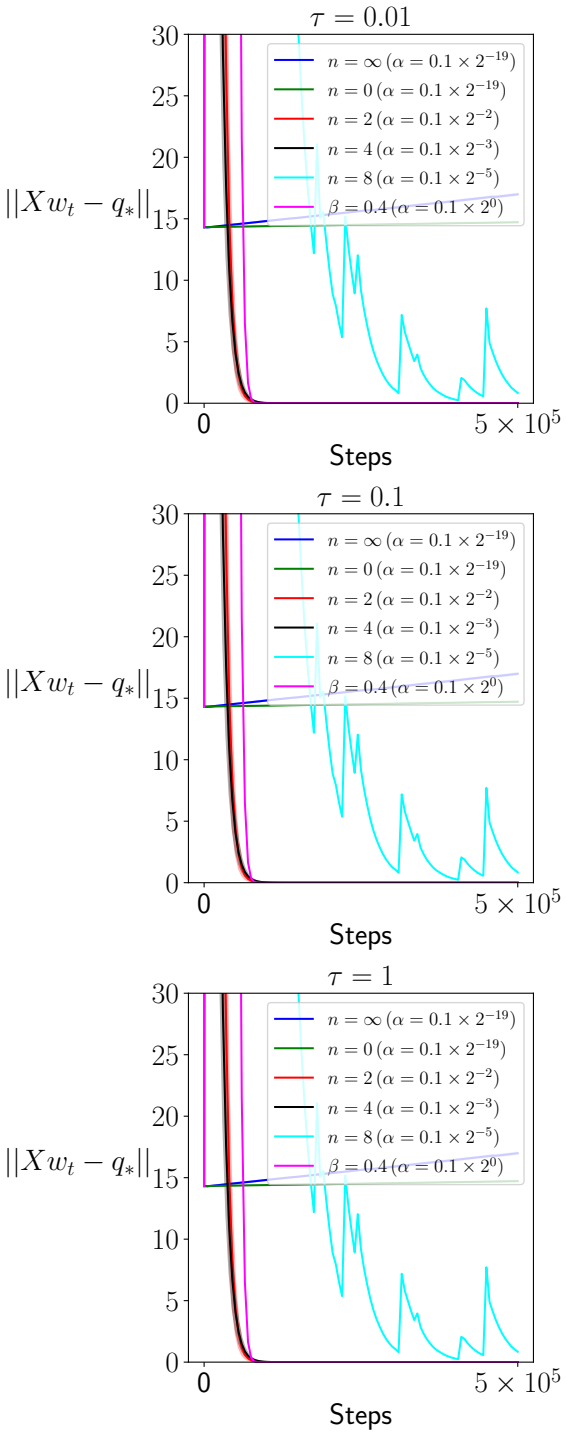

Figure 11.1: Truncated emphatic expected SARSA and its β-variant in the control setting with a fixed behavior policy. The shaded regions are invisible for some curves because their standard errors are too small.

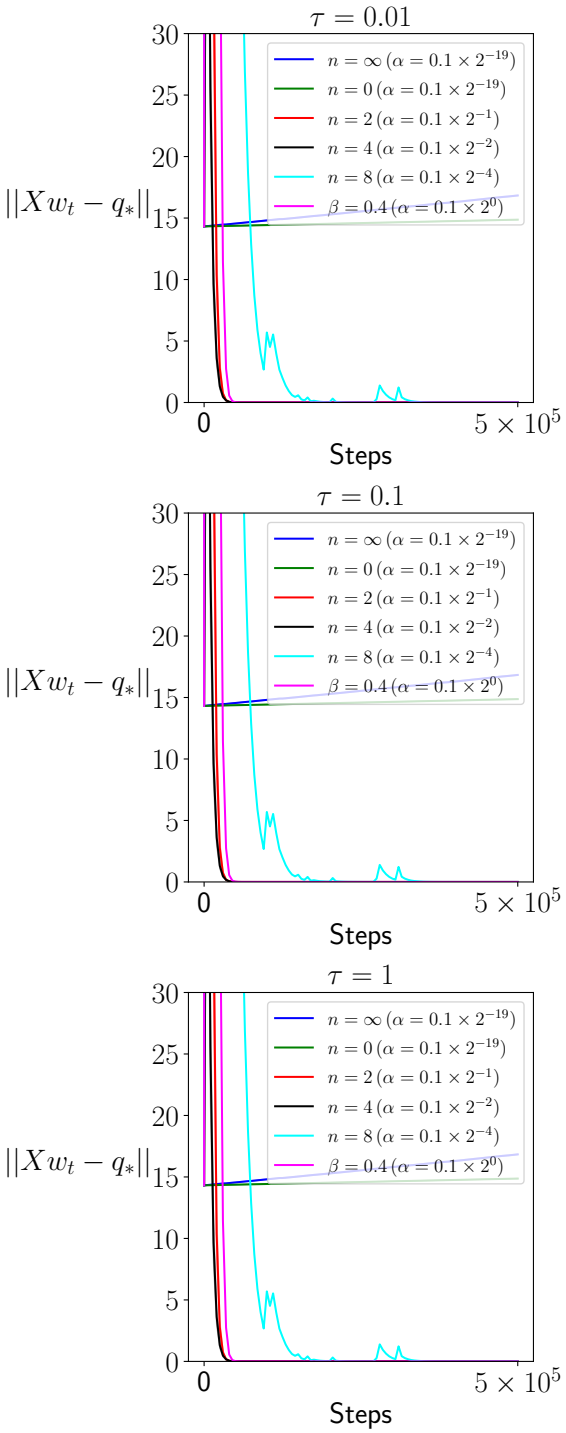

Figure 11.2: Truncated emphatic expected SARSA and its β-variant in the control setting with a changing behavior policy.

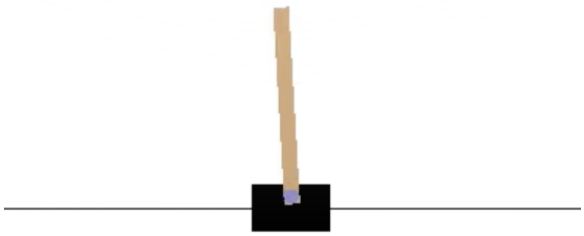

Figure 11.3: CartPole. At each time step, we observe the `velocity`, `acceleration`, `angular velocity`, and `angular acceleration` of the pole and move the car `left` or `right` to keep the pole balanced. The reward is +1 every time step. An episode ends if a maximum of 1000 steps is reached or the pole falls.

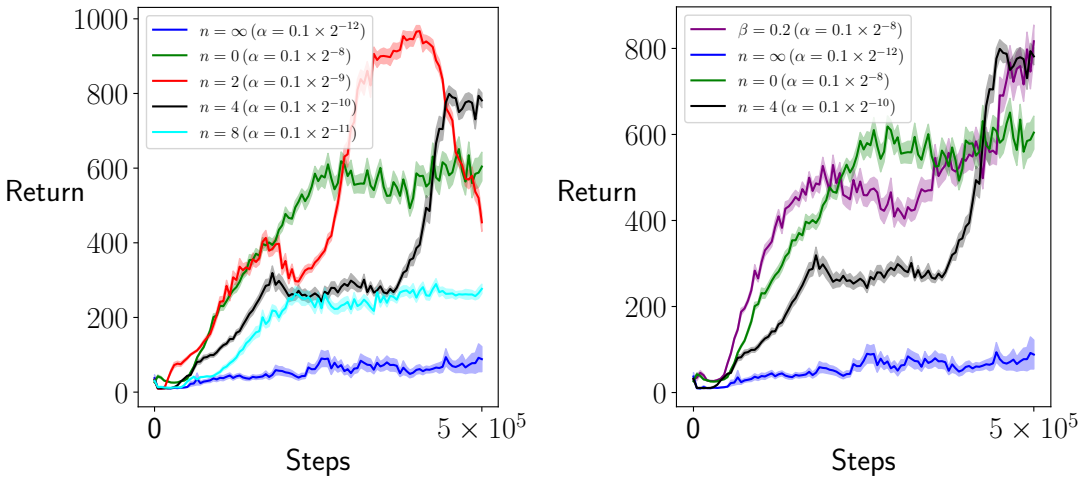

Figure 11.4: Truncated emphatic expected SARSA and its β-variant in the CartPole domain.

Part IV

Policy-Based Off-Policy Control

In this part, we focus on control problems with policy-based methods, where we optimize a parameterized policy directly in the context of the deadly triad, instead of deriving a policy from the estimation of the action value function.

Chapter 12

Control with Learned Emphasis

In this chapter, we consider optimizing the excursion objective, a widely used by rarely correctly optimized objective, with actor critic methods. We also extend the excursion objective to a more general average value objective and provide a corresponding policy gradient theorem.

12.1 Incomplete Gradient Estimators of the Excursion Objective

In the Markovian setting in Definition 2.2, one commonly used objective for off-policy control is the *excursion objective* (Degris et al., 2012; Ghiassian et al., 2018)

$$J_{\pi,\mu} \doteq \sum_s d_\mu(s) i(s) v_\pi(s). \tag{12.1}$$

We briefly describe the motivation behind this objective. In the off-policy setting we consider, states are obtained by following the behavior policy μ. At a particular state s, an agent might want to do an excursion by following the target policy for next steps. The performance of this excursion is indicated by the value function of the target policy $v_\pi(s)$. The performance of such excursions, weighted by both the state visitation $d_\mu(s)$ and the user's preference $i(s)$, is exactly the excursion objective.

Assuming the policy π is parameterized by $\theta \in \mathbb{R}^{K_3}$, Imani et al. (2018) compute the policy gradient of the excursion objective via the *off-policy policy gradient theorem* as

$$\nabla_\theta J_{\pi,\mu} = \mathbb{E}_{s\sim d_\mu(\cdot), a\sim\mu(\cdot|s)}[m_{\pi,\mu}(s) q_\pi(s,a) \rho_\theta(s,a) \nabla_\theta \log \pi(a|s)],$$

where $\rho_\theta(s,a) \doteq \frac{\pi_\theta(a|s)}{\mu(a|s)}$. One direct approach for optimizing $J_{\pi,\mu}$ is therefore to perform stochastic gradient ascent following $\nabla_\theta J_{\pi,\mu}$. At time step t, $\rho_\theta(A_t|S_t)\nabla_\theta \log \pi(A_t|S_t;\theta)$

is immediately available; $q_\pi(S_t, A_t)$ could be estimated by a parameterized function $q_w(S_t, A_t)$ trained with GTD or ETD. It is, however, not straightforward to obtain an estimate for $m_{\pi,\mu}(S_t)$.

Existing works usually deal with this $m_{\pi,\mu}$ in two ways. In Off-Policy Actor-Critic (Off-PAC, Degris et al. (2012)), this $m_{\pi,\mu}$ is ignored and θ is updated as

$$\theta_{t+1} \doteq \theta_t + \beta_t \rho_t q_w(S_t, A_t) \nabla_\theta \log \pi(A_t|S_t; \theta_t).$$

This naive approach has been widely used and great empirical success has been witnessed (Silver et al., 2014; Lillicrap et al., 2016; Wang et al., 2017b; Ciosek and Whiteson, 2020; Fujimoto et al., 2018; Espeholt et al., 2018). We, however, do not have any convergence guarantee for this naive approach under the presence of function approximation. In Actor-Critic with Emphatic weightings (ACE, Imani et al. (2018)), this $m_{\pi,\mu}(S_t)$ is approximated by the followon trace F_t and θ is updated as

$$\theta_{t+1} \doteq \theta_t + \beta_t F_t \rho_t q_w(S_t, A_t) \nabla_\theta \log \pi(A_t|S_t; \theta_t).$$

The motivation behind this approximation is (2.27). However, F_t is an unbiased estimation for $m_{\pi,\mu}(S_t)$ only in the limiting sense assuming the target policy π is fixed. In actor critic algorithms, the policy π changes every step. It is unclear whether the followon trace F_t can adapt quickly enough to the new policy. After all the followon trace is always a scalar while the emphasis is a vector of size $|\mathcal{S}|$. It might be too ambitious to approximate a vector with a scalar. Further, as discussed in Section 4.1, the followon trace F_t can have possibly unbounded variance. Consequently, so far we do not have any convergence guarantee for ACE either.

It thus remains an open problem to optimize the excursion objective with convergence guarantees, in the context of the deadly triad.

12.2 Backward and Forward Critics

In this thesis, we propose to learn the emphasis $m_{\pi,\mu}$ with GEM (Algorithm 2) and use the approximated emphasis for computing the gradient of the excursion objective. Instead of using Algorithm 2 directly, we additionally add ridge regularization to account for the fact that the policy changes every step. It can be seen soon that this additional ridge regularization plays a key role in the convergence analysis. It is also worth mentioning that the weight of the introduced ridge regularization can be arbitrarily small, indicating that the bias introduced by ridge regularization can be arbitrarily small. We similarly use GTD to learn the action value function q_π, with

additional ridge regularization. We refer to the resulting algorithm as Convergent Off-Policy Actor-Critic (COF-PAC, Algorithm 14).

Algorithm 14: Convergent off-policy actor critic

$t \leftarrow 0$
$S_0 \sim p_0(\cdot), A_0 \sim \mu(\cdot|S_0)$
while *True* **do**
  Execute A_t, get R_{t+1}, S_{t+1}
  Sample $A_{t+1} \sim \mu(\cdot|S_{t+1})$
  $x_t \leftarrow x(S_t), x_{t+1} \leftarrow x(S_{t+1})$
  $\kappa_{t+1} \leftarrow \kappa_t + \alpha_t(i(S_{t+1}) + \gamma\rho_t x_t^\top w_t - x_{t+1}^\top w_t - x_{t+1}^\top \kappa_t)x_{t+1}$
  $w_{t+1} \leftarrow w_t + \alpha_t\left((x_{t+1} - \gamma\rho_t x_t)x_{t+1}^\top \kappa_t - \eta w_t\right)$
  $\tilde{x}_t \leftarrow \tilde{x}(S_t, A_t), \tilde{x}_{t+1} \leftarrow \tilde{x}(S_{t+1}, A_{t+1})$
  $\tilde{\kappa}_{t+1} \leftarrow \tilde{\kappa}_t + \alpha_t(R_{t+1} + \gamma\rho_{t+1}\tilde{x}_{t+1}^\top u_t - \tilde{x}_t^\top u_t - \tilde{x}_t^\top \tilde{\kappa}_t)\tilde{x}_t$
  $u_{t+1} \leftarrow u_t + \alpha_t\left((\tilde{x}_t - \gamma\rho_{t+1}\tilde{x}_{t+1})\tilde{x}_t^\top \tilde{\kappa}_t - \eta u_t\right)$
  $\theta_{t+1} \leftarrow \theta_t + \beta_t\Gamma_1(w_t)\Gamma_2(u_t)\rho_t(w_t^\top x_t)(u_t^\top \tilde{x}_t)\nabla \log \pi_\theta(A_t|S_t)$
  $t \leftarrow t + 1$
end

In Algorithm 14, we train $w \in \mathbb{R}^{K_1}$ such that $x(s)^\top w$ approximates $m_{\pi,\mu}(s)$ similarly to GEM (Algorithm 2) with the help of an auxiliary weight κ, where $x : \mathcal{S} \to \mathbb{R}^{K_1}$ denotes the state feature function. We train $u \in \mathbb{R}^{K_2}$ such that $\tilde{x}(s,a)^\top u$ approximates $q_\pi(s,a)$ similarly to GTD with the help of an auxiliary weight $\tilde{\kappa}$, where $\tilde{x} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{K_2}$ denotes the state-action feature function. The hyperparameter η is the weight for the ridge regularization. We additionally use adaptive learning rates $\Gamma_1 : \mathbb{R}^{K_1} \to \mathbb{R}$ and $\Gamma_2 : \mathbb{R}^{K_2} \to \mathbb{R}$ to ensure θ changes slowly enough and make the following assumption.

Assumption 12.1. *For $i = 1, 2$,*

1. $0 < \Gamma_i(d) < \infty, \forall d$
2. *For all $\|d\| < C_1$, there exists $C_2 > 0$ such that $\Gamma_i(d) \geq C_2$*
3. *Γ_i is Lipschitz continuous.*

Konda (2002) provides an example for satisfying Assumption 12.1. Let $C_0 > 0$ be some constant. Konda (2002) defines

$$\Gamma_i(d) \doteq \begin{cases} 1, & \|d\| < C_0 \\ \frac{1+C_0}{1+\|d\|}, & \|d\| \geq C_0 \end{cases}.$$

In Algorithm 14, the weight u is the canonical critic which is in a forward sense in that it summarizes future rewards. The weight w plays a similar role to u but summarizes previous interests as discussed in Section 4.4. We, therefore, refer to w as a backward critic. We now proceed to analyzing the limiting behavior of the iterates θ_t in Algorithm 14. To this end, we first analyze the behavior of the two critics.

Proposition 12.1. *Let Assumptions 2.3, 2.4, 2.5, 2.6, 2.8 and 12.1 hold. Then the iterates $\{\kappa_t\}$ and $\{w_t\}$ generated by Algorithm 14 satisfy*

$$\begin{aligned}
&\sup_t \|\kappa_t\| < \infty, \\
&\sup_t \|w_t\| < \infty, \\
&\lim_{t\to\infty} \left\| \begin{bmatrix} C_\mu & -A_{\pi_{\theta_t},\mu}^\top \\ A_{\pi_{\theta_t},\mu} & \eta I \end{bmatrix} \begin{bmatrix} \kappa_t \\ w_t \end{bmatrix} - \begin{bmatrix} X^\top D_\mu i \\ 0 \end{bmatrix} \right\| = 0 \quad a.s. \quad .
\end{aligned}$$

The proof of Proposition 12.1 is provided in Section B.28. We recall that the $A_{\pi,\mu}, C_\mu, X, D_\mu, i$ in Proposition 12.1 are introduced in Definition 2.5. By simple block matrix inversion, it is easy to see that Proposition 12.1 implies

$$\lim_{t\to\infty} \left\| w_{\theta_t,\eta}^* - w_t \right\| = 0,$$

where

$$w_{\theta,\eta}^* \doteq -\left(A_{\pi_\theta,\mu} C_\mu^{-1} A_{\pi_\theta,\mu}^\top + \eta I\right)^{-1} A_{\pi_\theta,\mu} C_\mu^{-1} X^\top D i.$$

If $\eta = 0$ and $A_{\pi_\theta,\mu}$ is nonsingular, it can be easily computed that

$$X w_{\theta,0}^* = \Pi_{d_\mu} \hat{\mathcal{T}}_{\pi_\theta,\mu} X w_{\theta,0}^*.$$

In other words, the backward critic $\{w_t\}$ is able to track the minimizer of the MSPBE-like objective (4.6) induced by the current policy π_{θ_t}, up to regularization bias. Consequently, when we analyze the behavior of $\{\theta_t\}$, we can approximately use $w_{\theta_t,\eta}^*$ in place of w_t. We prove Proposition 12.1 with a convergence result about stochastic approximation algorithms from Konda (2002), which is provided in Section A.1 for completeness. One important step in the proof is to verify that the matrix multiplying $\begin{bmatrix} \kappa_t \\ w_t \end{bmatrix}$ in Proposition 12.1 is strictly positive definite, which is impossible if $\eta = 0$. This motivates the introduction of the ridge regularization. This ridge regularization is essential in the convergence of GEM under a slowly changing target policy.

Similarly, for the canonical forward critic u, we have

Proposition 12.2. *Let Assumptions 2.3, 2.4, 2.5, 2.6, 2.8 and 12.1 hold. Then the iterates* $\{\tilde{\kappa}_t\}$ *and* $\{u_t\}$ *generated by Algorithm 14 satisfy*

$$\begin{aligned}
&\sup_t \|\tilde{\kappa}_t\| < \infty, \\
&\sup_t \|u_t\| < \infty, \\
&\lim_{t\to\infty} \left\| \begin{bmatrix} \tilde{C}_\mu & -\tilde{A}_{\pi_{\theta_t},\mu} \\ \tilde{A}^\top_{\pi_{\theta_t},\mu} & \eta I \end{bmatrix} \begin{bmatrix} \tilde{\kappa}_t \\ u_t \end{bmatrix} - \begin{bmatrix} \tilde{X}^\top \tilde{D}_\mu r \\ 0 \end{bmatrix} \right\| = 0 \quad a.s. \quad .
\end{aligned}$$

The proof of Proposition 12.2 is identical to that of Proposition 12.1 up to change of notations and is thus omitted. We recall that the $\tilde{A}_{\pi,\mu}, \tilde{C}_\mu, \tilde{X}, \tilde{D}_\mu, r$ in Proposition 12.2 are introduced in Definition 2.6 and Remark 1. Similarly, by matrix inversion, we have

$$\lim_{t\to\infty} \left\| u^*_{\theta_t,\eta} - u_t \right\| = 0,$$

where

$$u^*_{\theta,\eta} \doteq -\left(\tilde{A}^\top_{\pi_\theta,\mu} \tilde{C}^{-1}_\mu \tilde{A}_{\pi_\theta,\mu} + \eta I\right)^{-1} \tilde{A}^\top_{\pi_\theta,\mu} \tilde{C}^{-1}_\mu \tilde{X}^\top \tilde{D}_\mu r.$$

If $\eta = 0$ and $A_{\pi_\theta,\mu}$ is nonsingular, it can be easily computed that

$$\tilde{X} u^*_{\theta,0} = \tilde{\Pi}_{\tilde{d}_\mu} \tilde{\mathcal{T}}_{\pi_\theta} \tilde{X} u^*_{\theta,0}.$$

In other words, the forward critic $\{u_t\}$ is able to track the minimizer of the MSPBE (4.6) induced by the current policy π_{θ_t}, up to regularization bias. Consequently, when we analyze the behavior of $\{\theta_t\}$, we can approximately use $u^*_{\theta_t,\eta}$ in place of u_t.

Having established the trackability of both critics, we are now ready to analyze the behavior of the policy parameters $\{\theta_t\}$. Since both critics use linear function approximation, it is inevitable to have bias due to the limited capacity of the linear architecture. In other words, the best approximation $X w^*_{\theta,0}$ is not necessarily equal to $m_{\pi_\theta,\mu}$ and the best approximation $\tilde{X} u^*_{\theta,0}$ is not necessarily equal to q_{π_θ}. The ridge regularization also introduces extra bias, though it can be arbitrarily small. Overall, we use the following term

$$\begin{aligned}
&b(\theta) \qquad\qquad (12.2) \\
\doteq &\mathbb{E}_{(s,a)\sim\tilde{d}_\mu(\cdot)} \left[\left(m_{\pi,\mu}(s) q_\pi(s,a) - \left(x(s)^\top w^*_{\theta,\eta}\right) \left(\tilde{x}(s,a)^\top u^*_{\theta,\eta}\right) \right) \rho_\theta(s,a) \nabla_\theta \log \pi_\theta(a|s) \right]
\end{aligned}$$

to characterize the bias we have introduced when estimating the gradient $\nabla_\theta J_{\pi_\theta,\mu}$. After making the following assumption on how the policy π is parameterized by θ, we arrive at the convergence of Algorithm 14.

Assumption 12.2. *There exists a constant $C_0 < \infty$ such that $\forall(s, a, \theta, \theta')$,*

$$\begin{aligned}\|\nabla_\theta \pi_\theta(a|s)\| \leq & C_0,\\ \left|\frac{\partial^2 \pi_\theta(a|s)}{\partial\theta_i \partial\theta_j}\right| \leq & C_0,\\ |\pi_\theta(a|s) - \pi_{\theta'}(a|s)| \leq & C_0\|\theta - \theta'\|,\\ \|\nabla_\theta \pi_\theta(a|s) - \nabla_\theta \pi_{\theta'}(a|s)\| \leq & C_0\|\theta - \theta'\|.\end{aligned}$$

Theorem 12.3. *Let Assumptions 2.3, 2.4, 2.5, 2.6, 2.8, 12.1, and 12.2 hold. Then the iterates $\{\theta_t\}$ generated by COF-PAC (Algorithm 14) satisfy*

$$\liminf_t \left(\left\|\nabla_\theta J_{\pi_{\theta_t},\mu}\right\| - \|b(\theta_t)\|\right) \leq 0 \quad a.s.,$$

i.e., the iterates $\{\theta_t\}$ visit small neighborhoods of the set $\{\theta : ||\nabla J(\theta)|| \leq ||b(\theta)||\}$ infinitely many times.

The proof of Theorem 12.3 is provided in Section B.30. According to Theorem 12.3, COF-PAC reaches the same convergence level as the canonical on-policy actor-critic (Konda, 2002). Together with the fact that $\nabla_\theta J_{\pi_\theta,\mu}$ is Lipschitz continuous and β_t is diminishing, it is easy to see θ_t will eventually remain in the neighborhood of $\{\theta : \|\nabla_\theta J_{\pi_\theta,\mu}\| \leq \|b(\theta)\|\}$ for arbitrarily long time. In other words, the policy parameters will eventually visit the neighborhood of the stationary points of the excursion objective, depending on function approximation and regularization bias.

The bias $b(\theta)$ involves the difference between the minimizer of the MSPBE and the true action value function. As discussed in Kolter (2011), it is in general hard to bound such a difference. We can, however, provide a bound for $b(\theta)$ assuming that the behavior policy is not too far from the target policies in the following sense:

Assumption 12.3. *For any θ, the following two matrices*

$$\begin{bmatrix} C_\mu & X^\top P_{\pi_\theta}^\top D_\mu X \\ X^\top D_\mu P_{\pi_\theta} X & C_\mu \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \tilde{C}_\mu & \tilde{X}^\top \tilde{P}_{\pi_\theta}^\top \tilde{D}_\mu \tilde{X} \\ \tilde{X}^\top \tilde{D}_\mu \tilde{P}_{\pi_\theta} \tilde{X} & \tilde{C}_\mu \end{bmatrix}$$

are positive semidefinite.

Proposition 12.4. *Let Assumptions 2.3 and 12.3 hold. Assume for any π, $A_{\pi,\mu}$ and $\tilde{A}_{\pi,\mu}$ are nonsingular. Then we have*

$$\begin{aligned}\|b(\theta)\|_{d_\mu} = \mathcal{O}(\eta) &+ \mathcal{O}\left(\frac{1+\gamma\|P_{\pi_\theta}\|_{d_\mu}}{1-\gamma}\left\|m_{\pi_\theta,\mu} - \Pi_{d_\mu} m_{\pi_\theta,\mu}\right\|_{d_\mu}\right)\\ &+ \mathcal{O}\left(\frac{1+\gamma\left\|\tilde{P}_{\pi_\theta}\right\|_{\tilde{d}_\mu}}{1-\gamma}\left\|q_{\pi_\theta} - \tilde{\Pi}_{\tilde{d}_\mu} q_{\pi_\theta}\right\|_{d_\mu}\right).\end{aligned}$$

The proof of Proposition 12.4 is provided in Section B.31 and is a natural extension of Proposition 4.3.

12.3 Compatible Features

Although Proposition 12.4 gives a bound for the bias, it is restrictive in that it applies only if the target policies are not too far away from the behavior policy. The reason is that we use linear function approximation but $m_{\pi,\mu}$ and q_π might not lie in the column space of the feature matrix. However, if there is flexibility in what features we use, it is indeed possible that the bias diminishes, even if $m_{\pi,\mu}$ and q_π still do not lie in the column space. Those features are usually referred to as *compatible features*(Sutton et al., 1999a; Konda, 2002) and we in this section develop compatible features in the context of Algorithm 14.

Let $\hat{m}_\theta, \hat{q}_\theta$ be estimates for $m_{\pi,\mu}, q_\pi$. The bias introduced by using those estimates is essentially

$$b(\theta) = \mathbb{E}_{(s,a)\sim\tilde{d}_\mu(\cdot)}\left[\left(m_{\pi,\mu}(s)q_\pi(s,a) - \hat{m}_\theta(s)\hat{q}_\theta(s,a)\right)\rho_\theta(s,a)\nabla_\theta \log \pi_\theta(a|s)\right]. \tag{12.3}$$

The definition of bias in (12.2) is a special case of (12.3) where the estimates are made based on the features X and $\tilde{X}$. We can expand $b(\theta)$ as

$$\begin{aligned} b(\theta) =& \mathbb{E}_{(s,a)\sim\tilde{d}_\mu(\cdot)}\left[\left(m_{\pi,\mu}(s)q_\pi(s,a) - m_{\pi,\mu}(s)\hat{q}_\theta(s,a)\right)\rho_\theta(s,a)\nabla_\theta \log \pi_\theta(a|s)\right] \\ &+ \mathbb{E}_{(s,a)\sim\tilde{d}_\mu(\cdot)}\left[\left(m_{\pi,\mu}(s)\hat{q}_\theta(s,a) - \hat{m}_\theta(s)\hat{q}_\theta(s,a)\right)\rho_\theta(s,a)\nabla_\theta \log \pi_\theta(a|s)\right] \\ =& \underbrace{\sum_{s,a} d_\mu(s)m_{\pi,\mu}(s)\mu(a|s)\left(q_\pi(s,a) - \hat{q}_\theta(s,a)\right)\rho_\theta(s,a)\nabla_\theta \log \pi(a|s)}_{b_1(\theta)} \\ &+ \underbrace{\sum_s d_\mu(s)\left(m_\pi(s) - \hat{m}_\theta(s)\right)\sum_a \nabla_\theta \pi(a|s)\hat{q}_\theta(s,a)}_{b_2(\theta)}. \end{aligned}$$

Since

$$\rho_\theta(s,a)\nabla_\theta \log \pi(a|s) \in \mathbb{R}^{K_3},$$

for $i \in \{1, 2, \ldots, K_3\}$, we can define $x^\theta_{1,i}$ to be a vector in $\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ whose (s,a)-indexed element is the i-th element of $\rho_\theta(s,a)\nabla_\theta \log \pi(a|s) \in \mathbb{R}^{K_3}$. Or more intuitively, $x^\theta_{i,i}$ is the i-the row of the matrix

$$\rho_\theta(\cdot,\cdot)\nabla_\theta \log \pi(\cdot|\cdot) \in \mathbb{R}^{K_3\times|\mathcal{S}\times\mathcal{A}|}.$$

Consequently, the i-th element of $b_1(\theta)$ can be expressed as

$$\begin{aligned} b_{1,i}(\theta) =& \sum_{s,a} d_\mu(s) m_{\pi,\mu}(s) \mu(a|s) \left(q_\pi(s,a) - \hat{q}_\theta(s,a)\right) x^\theta_{1,i}(s,a) \\ =& \left\langle q_\pi - \hat{q}_\theta, x^\theta_{1,i} \right\rangle_{\Psi_1}, \end{aligned}$$

where we use Ψ_1 to denote the subspace in $\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ spanned by $\left\{x^\theta_{1,i}\right\}_{i=1,\ldots,K_3}$ and $\langle\cdot,\cdot\rangle_{\Psi_1}$ denotes an inner product in the subspace Ψ_1 defined as

$$\langle y, y' \rangle_{\Psi_1} \doteq \sum_{s,a} d_\mu(s) m_{\pi,\mu}(s) \mu(s,a) y(s,a) y'(s,a).$$

This inner product also induces a norm, which we denote as $\|\cdot\|_{\Psi_1}$, as well as a projection onto this subspace, which we denote as Π_{Ψ_1}. Then we can further expand $b_{1,i}(\theta)$ as

$$\begin{aligned} b_{1,i}(\theta) =& \left\langle q_\pi - \Pi_{\Psi_1} q_\pi, x^\theta_{1,i} \right\rangle_{\Psi_1} + \left\langle \Pi_{\Psi_1} q_\pi - \hat{q}_\theta, x^\theta_{1,i} \right\rangle_{\Psi_1} \\ =& \left\langle \Pi_{\Psi_1} q_\pi - \hat{q}_\theta, x^\theta_{1,i} \right\rangle_{\Psi_1}, \end{aligned}$$

where the last equality results from Pythagoras. This indicates that if our estimate satisfies

$$\hat{q}_\theta = \Pi_{\Psi_1} q_\pi,$$

then the bias term $b_1(\theta)$ vanishes, even if $\hat{q}_\theta$ is still not equal to q_π. One way to learn such an estimate $\hat{q}_\theta$ is to use linear function approximation with features $\left\{x^\theta_{1,i}\right\}_{i=,\ldots,K_3}$, using ETD($\lambda$) (Sutton et al., 2016) with $\lambda = 1$.

Similarly, for $i \in \{1, 2, \ldots, K_3\}$, we can define $x^\theta_{2,i}$ to be a vector in $\mathbb{R}^{|\mathcal{S}|}$ whose s-indexed element is the i-th element of $\sum_a \nabla_\theta \pi(a|s) \hat{q}_\theta(s,a)$. Or more intuitively, $x^\theta_{2,i}$ is the i-the row of the matrix

$$\sum_a \nabla_\theta \pi(a|\cdot) \hat{q}_\theta(\cdot, a) \in \mathbb{R}^{K_3 \times |\mathcal{S}|}.$$

Consequently, the i-th element of $b_2(\theta)$ can be expressed as

$$b_{2,i}(\theta) = \left\langle m_\pi - \hat{m}_\theta, x^\theta_{2,i} \right\rangle_{\Psi_2},$$

where we use Ψ_2 to denote the subspace in $\mathbb{R}^{|\mathcal{S}|}$ spanned by $\left\{x^\theta_{2,i}\right\}_{i=1,\ldots,K_3}$ and $\langle\cdot,\cdot\rangle_{\Psi_2}$ denotes an inner product in the subspace Ψ_2 defined as

$$\langle y, y' \rangle_{\Psi_2} \doteq \sum_s d_\mu(s) y(s) y'(s).$$

The corresponding induced norm and projection are denoted as $\|\cdot\|_{\Psi_2}$ and Π_{Ψ_2}. We similarly conclude that if

$$\hat{m}_\theta = \Pi_{\Psi_2} m_{\pi,\mu},$$

then the bias term $b_2(\theta)$ vanishes. One way to learn such an estimate $\hat{m}_\theta$ is to use linear function approximation with features $\left\{x^\theta_{2,i}\right\}_{i=1,\dots,K_3}$, using GEM($\lambda$) with $\lambda = 1$. Here GEM(λ) is an analogue of GTD(λ) in Yu (2017).

The compatible features $x^\theta_{1,i}$ and $x^\theta_{2,i}$ depend on the policy parameter θ thus change every time step. Ideally, an off-policy actor critic should use those compatible features and train the backward and forward critics with GEM(1) and GTD(1) respectively to eliminate the bias due to the capacity limit of the linear architecture. We, however, leave the investigation of such an algorithm for future work.

12.4 Beyond the Excursion Objective: A New Average Value Objective

The excursion objective (12.1) can be regarded as an average value objective in the sense that it measures the value function of the target policy with the state distribution of the behavior policy and the interest function. Using the state distribution of the behavior policy to compose an objective for off-policy learning is a common practice, see, e.g., the off-policy MSPBE (2.22). In on-policy settings, where $\pi = \mu$, the excursion objective degenerates to

$$\sum_s d_\pi(s) i(s) v_\pi(s), \tag{12.4}$$

akin to the on-policy MSPBE (2.10). This objective (12.4) is usually referred to as the alternative life objective (Ghiassian et al., 2018). It measures the value function of the target policy with the state distribution of the target policy, which is available only if the agent had an alternative life such that it follows the target policy instead of the behavior policy. The alternative life objective can be regarded as an extension of the average reward objective. To see this, consider the setting where $i(s) \equiv i_c$ for

some constant i_c. We then have

$$\begin{aligned}\sum_s d_\pi(s)i(s)v_\pi(s) =& i_c d_\pi^\top v_\pi \\ =& i_c d_\pi^\top (r_\pi + \gamma P_\pi v_\pi) \\ =& i_c d_\pi^\top r_\pi + \gamma i_c d_\pi^\top P_\pi v_\pi \\ =& i_c d_\pi^\top r_\pi + \gamma i_c d_\pi^\top v_\pi \\ \implies d_\pi^\top r_\pi =& \frac{1-\gamma}{i_c} i_c d_\pi^\top v_\pi.\end{aligned}$$

In other words, if the interest function is constant, the excursion objective degenerates to the average reward, up to some constant multiplier. This degeneration also requires that the discount factor γ is constant. When γ is a function of (s, a, s') (see, e.g., White (2017)), this degeneration does not hold even if the interest function is constant. In this section, we focus on the setting where γ is a constant for simplifying presentation and refer the reader to Zhang et al. (2019) for the treatment of a transition-dependent discount factor.

We have introduced Algorithm 14 for optimizing the excursion objective in previous sections. It is, however, remain an open problem to optimize the alternative life objective in the off-policy setting, similar to optimizing the on-policy MSBPE (2.10) in the off-policy setting. We, in this section, make progress towards this open problem via providing a new unifying average value objective with an additional hyperparameter $\hat{\gamma} \in [0, 1]$ such that this new unifying objective recovers the excursion objective and the alternative life objective when $\hat{\gamma} = 0$ and $\hat{\gamma} = 1$ respectively. We then provide the corresponding off-policy policy-gradient theorem for $\hat{\gamma} \in [0, 1)$.

To this end, we first seek to unify d_π and d_μ. We borrow ideas from Gelada and Bellemare (2019), who define a new transition matrix

$$P_{\hat{\gamma}} \doteq \hat{\gamma} P_\pi + (1 - \hat{\gamma}) 1 d_\mu^\top.$$

Following this transition matrix, an agent either proceeds to a successor state according to P_π w.p. $\hat{\gamma}$ or gets reset to a state according to d_μ w.p. $1 - \hat{\gamma}$. Gelada and Bellemare (2019) show that the chain induced by $P_{\hat{\gamma}}$ is ergodic under mild conditions. Let $d_{\hat{\gamma}}$ denote its ergodic distribution, Gelada and Bellemare (2019) show that

$$d_{\hat{\gamma}} = \begin{cases} (1 - \hat{\gamma})(I - \hat{\gamma} P_\pi^\top)^{-1} d_\mu, & \hat{\gamma} < 1 \\ d_\pi, & \hat{\gamma} = 1 \end{cases}.$$

With the help of $d_{\hat{\gamma}}$, we now introduce our new average value objective

$$J_{\hat{\gamma}} \doteq \sum_s d_{\hat{\gamma}}(s) i(s) v_\pi(s).$$

It can be easily seen that $J_{\hat{\gamma}}$ recovers the excursion objective when $\hat{\gamma} = 0$ and the alternative life objective when $\hat{\gamma} = 1$. Thanks to the continuity of the ergodic distribution w.r.t. the transition matrix, we have

$$\lim_{\hat{\gamma} \to 0} J_{\hat{\gamma}} = \sum_s d_\mu(s) i(s) v_\pi(s),$$
$$\lim_{\hat{\gamma} \to 1} J_{\hat{\gamma}} = \sum_s d_\pi(s) i(s) v_\pi(s).$$

This indicates that $J_{\hat{\gamma}}$ is a good interpolation between the excursion objective and the alternative life objective. The following theorem gives the gradient of $J_{\hat{\gamma}}$.

Theorem 12.5. *Let Assumptions 2.1 and 2.7 hold. For $\hat{\gamma} < 1$, we have*

$$\frac{\partial}{\partial \theta_i} J_{\hat{\gamma}} = \sum_s d_\mu(s) h(s) \sum_a q_\pi(s,a) \frac{\partial}{\partial \theta_i} \pi(a|s) + \sum_s d_\mu(s) i(s) v_\pi(s) g_i(s),$$

where

$$h \doteq D_\mu^{-1} (I - \gamma P_\pi^\top)^{-1} diag(d_{\hat{\gamma}}) i,$$
$$g_i \doteq \hat{\gamma} D_\mu^{-1} (I - \hat{\gamma} P_\pi^\top)^{-1} \left(\frac{\partial}{\partial \theta_i} P_\pi^\top \right) d_{\hat{\gamma}}.$$

The proof of Theorem 12.5 is provided in Section B.32. Inspired by Imani et al. (2018), where the followon trace is used to estimate the gradient of the excursion objective, we now provide a trace-based method to estimate $\nabla_\theta J_{\hat{\gamma}}$. Consider the Markovian setting in Definition 2.2. Fix the target policy π. Define

$$\begin{aligned}
\tau_{\hat{\gamma}}(s) &\doteq \frac{d_{\hat{\gamma}}(s)}{d_\mu(s)}, \\
F_t^{(1)} &\doteq i(S_t) \tau_{\hat{\gamma}}(S_t) + \gamma \rho_{t-1} F_{t-1}^{(1)}, \\
F_t^{(2)} &\doteq \tau_{\hat{\gamma}}(S_{t-1}) \rho_{t-1} \nabla_\theta \log \pi(A_{t-1}|S_{t-1}) + \hat{\gamma} \rho_{t-1} F_{t-1}^{(2)}, \\
Z_t &\doteq \rho_t F_t^{(1)} q_\pi(S_t, A_t) \nabla_\theta \log \pi(A_t|S_t) + \hat{\gamma} i(S_t) v_\pi(S_t) F_t^{(2)}.
\end{aligned}$$

Then we have

Proposition 12.6. *Let Assumptions 2.1 and 2.7 hold. Assume μ and π are fixed. Then*

$$\lim_{t \to \infty} \mathbb{E}[Z_t] = \nabla_\theta J_{\hat{\gamma}}.$$

The proof of Proposition 12.6 is provided in Section B.33. Proposition 12.6 asserts that Z_t can be used as an estimate for $\nabla J_{\hat{\gamma}}$, provided that both μ and π are fixed. Though Z_t is biased for any finite t, it is consistent when t goes to infinity. For computing Z_t, one need to know $v_\pi, q_\pi, \tau_{\hat{\gamma}}$. The value functions v_π and q_π can be learned using standard off-policy learning methods. The ratio $\tau_{\hat{\gamma}}$ can be learned with reverse RL methods and we refer the reader to Gelada and Bellemare (2019) for details. When computing Z_t, $F_t^{(1)}$ is simply the canonical followon trace with a specifically designed interest function $i(s)\tau_{\hat{\gamma}}(s)$. $F_t^{(2)}$ also has the same structure as the followon trace. $F_t^{(2)}$ is, however, a vector instead of a scalar. This is because the "interest" $\tau_{\hat{\gamma}}(S_{t-1})\rho_{t-1}\nabla_\theta \log \pi(A_{t-1}|S_{t-1})$ is now a vector.

Though Theorem 12.5 computes the gradient of $J_{\hat{\gamma}}$ and Proposition 12.6 gives a practical way to estimate the gradient, so far we do not have any convergent algorithm for optimizing $J_{\hat{\gamma}}$, which we leave for future work. A practical algorithm that uses Z_t for optimizing $J_{\hat{\gamma}}$ and the corresponding empirical investigation are available in Zhang et al. (2019), which we do not include in this thesis since we believe they deviate from the main topic of this thesis.

12.5 Other Off-Policy Actor-Critic Algorithms

Instead of the excursion objective, Maei (2018) proposes the Gradient Actor-Critic algorithm under a different objective,

$$\sum_s d_\mu(s) v_w(s),$$

for off-policy learning with function approximation. This objective differs from the excursion objective in that it replaces the true value function v_π with an estimate v_w. Consequently, the optimal policy under this objective depends on the features used to approximate the value function, and this approximation of the excursion objective can be arbitrarily poor. Maei (2018) tries to show the convergence of a GTD critic under a slowly changing target policy with results from Konda (2002). In this thesis, we show that GTD has to be regularized before the results from Konda (2002) can take over. Furthermore, the policy gradient estimator Maei (2018) proposes is also based on the followon trace. That estimator tracks the true gradient only in a limiting sense under a fixed π (Maei, 2018, Theorem 2) and has potentially unbounded variance, similar to how F_t tracks $m_{\pi,\mu}(S_t)$. It is unclear if that policy gradient estimator can track the true policy gradient under a changing π or not. To address this issue, we instead use function approximation to learn the emphasis directly.

A line of policy-based off-policy algorithms involves reward shaping via policy entropy. In particular, SBEED (Dai et al., 2018) is an off-policy actor-critic algorithm with a finite-sample analysis on the statistical error. The convergence analysis of SBEED (Theorem 5 in Dai et al. (2018)) is conducted within a bi-level optimization framework, assuming the exact solution of the inner optimization problem can be obtained. With function approximation, requiring the exact solution is usually impractical due to representation error. Even if the exact solution is representable, computing it explicitly is still expensive (cf. solving a least-squares regression problem with a large feature matrix exactly). By contrast, our COF-PAC adopts a two-timescale perspective, where we do not need to exactly solve the optimization problems for the two critics every step. Other works in this line of research include Nachum et al. (2017); O'Donoghue et al. (2017); Schulman et al. (2017a); Nachum et al. (2018); Haarnoja et al. (2017, 2018), which are mainly developed in a tabular setting and do not have a convergence analysis under function approximation.

Liu et al. (2019) propose to reweight the Off-PAC update via the density ratio τ_γ. As discussed in previous sections, many methods can be used to learn this density ratio. The convergence of those density ratio learning algorithms under a slowly changing target policy is, however, unclear. For GradientDICE with linear function approximation, it is possible to employ our arguments for proving Theorem 12.1 to prove its convergence under a slowly changing target policy and thus give a convergent analysis for this reweighted Off-PAC in a two-timescale form. We leave this for future work.

In AlgaeDICE, Nachum et al. (2019b) propose a new objective for off-policy actor-critic and reformulate the policy optimization problem into a minimax problem. Primal-dual algorithms can then take over. Nachum et al. (2019b) show the primal variable works similarly to an actor, and the dual variable works similarly to a critic. It is possible to provide a two-timescale convergent analysis for AlgaeDICE when the dual variable is linear and the primal variable is nonlinear using arguments from this paper, which we also leave for future work.

Chapter 13

Related Work

13.1 Interest, Emphasis, Density Ratio, and Importance Sampling Ratio

Though the interest function i is introduced by Sutton et al. (2016) to specify user's preference to different states, it is usually trivially set to $i(s) \equiv 1$ (Sutton et al., 2016). Section 12.4 instead provides examples of nontrivial interest functions, even vector interest functions.

One important observation to make is that if the interest function is set to be

$$i(s) = (1-\gamma)\frac{p_0(s)}{d_\mu(s)}, \tag{13.1}$$

then according to Proposition 4.1, the emphasis $m_{\pi,\mu}$ can be computed as

$$\begin{aligned} m_{\pi,\mu} =& (1-\gamma)(I - \gamma D_\mu^{-1} P_\pi^\top D_\mu)^{-1} D_\mu^{-1} p_0 \\ =& (1-\gamma) D_\mu^{-1} (I - \gamma P_\pi^\top)^{-1} p_0 \\ =& D_\mu^{-1} d_{\pi,\gamma}, \\ \implies m_{\pi,\mu}(s) =& \frac{d_{\pi,\gamma}(s)}{d_\mu(s)}. \end{aligned} \tag{13.2}$$

In other words, the emphasis is a generalization of the density ratio in the discounted setting. Unfortunately, since $d_\mu(s)$ is in general unknown, the interest function (13.1) is not available in practice. Consequently, GEM (Algorithm 2) cannot be used to learn the density ratio directly. We leave the use of (13.2) for density ratio learning for future work.

For emphasis learning, our operator $\hat{\mathcal{T}}_{\pi,\mu}$ in (4.4) is a generalization of the discounted COP-TD operator $\mathcal{T}_{\hat{\gamma}}$ in Gelada and Bellemare (2019) defined as

$$\mathcal{T}_{\hat{\gamma}} y \doteq (1-\hat{\gamma})1 + \hat{\gamma} D_\mu^{-1} P_\pi^\top D_\mu y.$$

By setting $\gamma = \hat{\gamma}$ and $i(s) \equiv 1 - \hat{\gamma}$, our $\hat{\mathcal{T}}_{\pi,\mu}$ recovers $\mathcal{T}_{\hat{\gamma}}$. Gelada and Bellemare (2019) show $\mathcal{T}_{\hat{\gamma}}$ is contractive only for small enough $\hat{\gamma}$ while Proposition 4.1 proves contraction of $\hat{\mathcal{T}}_{\pi,\mu}$ for any $\gamma < 1$. The algorithm in Gelada and Bellemare (2019) for learning $\frac{d_{\hat{\gamma}}(s)}{d_\mu(s)}$ based on $\mathcal{T}_{\hat{\gamma}}$ is not convergent while our GEM based on $\hat{\mathcal{T}}_{\pi,\mu}$ is proven to be convergent.

Learning the density ratio with function approximation dates back to Hallak and Mannor (2017); Liu et al. (2018), both of which require trajectories generated by a *single known* behavior policy. Nachum et al. (2019a); Uehara et al. (2020) relax this constraint to work with transitions from *multiple unknown* behavior policies. The methods of Nachum et al. (2019a); Uehara et al. (2020), however, apply to only the discounted setting. Zhang et al. (2020a); Mousavi et al. (2020) propose algorithms for learning the density ratio for the discounted and the average reward settings. However, neither of Zhang et al. (2020a); Mousavi et al. (2020) is provably convergent with linear function approximation. By contrast, GradientDICE (Algorithm 6) is compatible with transitions from multiple known behavior policies, applies to both discounted and average reward settings, and is provably convergent with linear function approximation. Besides learning the density ratio as a whole, Xie et al. (2019) attack the density ratio learning problem in finite horizon MDPs via learning the nominator and the denominator separately with tabular methods.

Many density ratio learning algorithms, as well as our emphasis learning algorithms, belong to the family or reverse RL in the sense that they require bootstrapping backwards in time. This backward bootstrapping has three origins. The first origin is the followon trace which is the basis of many algorithms proposed in this thesis. The second origin is related to learning the stationary distribution of a policy, which dates back to Wang et al. (2007a,b) in dual dynamic programming for stable policy evaluation and policy improvement. The third origin is an application of RL in web page ranking (Yao and Schuurmans, 2013) to learn the authority score function for a web page.

Instead of using density ratios, products of important sampling ratios can also be used for off-policy prediction, see, e.g., Precup et al. (2001); Thomas et al. (2015); Jiang and Li (2016) and Chapter 3 of Thomas (2015). Those methods, however, in general suffer from the deadly triad. The followon trace is also essentially products of important sampling ratios. It, however, properly gates the products with γ^l where l is the length of the products. Consequently, the resulting emphatic TD methods are convergent.

13.2 Regularization

Ridge regularization is a key ingredient in many algorithms proposed in this thesis. It is used mainly for three purposes. First, ridge regularization is used to ensure the solution to least squares does not change too fast. See, for example, the algorithms based on target networks. Second, ridge regularization is used to account for the potential singularity of $A_{\pi,\mu}$ with $\gamma = 1$ and $\bar{A}_{\pi,\mu}$ to ensure the uniqueness of the solutions. See, for example, GradientDICE and Diff-GQ2. This is especially important for the average-reward setting because of the lack of contraction of the differential Bellman operators. Third, ridge regularization is used to help establish the uniform positive definiteness of the critic updates and thus establish its ability to track the changing policy. See, for example, COF-PAC.

Regularization is indeed widely used in RL. For example, Kolter and Ng (2009); Johns et al. (2010); Petrik et al. (2010); Painter-Wakefield et al. (2012); Liu et al. (2012) use Lasso regularization in policy evaluation, mainly for feature selection. There also previous works that introduce regularization in MSPBE objectives. Mahadevan et al. (2014) introduce the proximal GTD learning framework to integrate GTD algorithms with first-order optimization-based regularization via saddle-point formulations and proximal operators. Yu (2017) introduces a general regularization term in MSPBE objectives for improving robustness. Du et al. (2017) introduce ridge regularization to improve the convexity of MSPBE. However, the analysis of Mahadevan et al. (2014); Yu (2017); Du et al. (2017) is conducted with the saddle-point formulation of the GTD objective (Liu et al., 2015; Macua et al., 2015) and requires a fixed target policy, which is impractical in control settings. We are the first to establish the tracking ability of GTD-style algorithms (i.e., the updates of the backward and forward critics in COF-PAC) under a slowly changing target policy by introducing ridge regularization. Without this ridge regularization, we are not aware of any existing work establishing this tracking ability.

13.3 Differential Value Functions

As a performance metric, the average reward receives far less attention than the discounted total rewards. One reason might be that the differential Bellman equations (e.g., (2.5)) are not contractive. Consequently, solving them is fundamentally harder than solving the canonical Bellman equations. Another reason might be that the applications of the average reward is much less than the applications of the discounted

total rewards. It is, however, worth mentioning that there is indeed argument that the discounted total rewards should be deprecated when function approximation is at play and the average reward should be more promoted (Chapter 10.3 of Sutton and Barto (2018)). Nevertheless, few methods exist for learning differential value functions. These are either on-policy methods with linear function approximation or off-policy methods. For example, Tsitsiklis and Roy (1999) prove the convergence of differential linear TD in the on-policy setting, which is later on used by Konda and Tsitsiklis (1999); Bhatnagar et al. (2009) in on-policy actor-critic algorithms for training the critic. Yu and Bertsekas (2009) provide a least-squares style algorithm for learning the differential value function in the on-policy setting and prove its convergence. This algorithm is later on extended by Abbasi-Yadkori et al. (2019). In the off-policy setting, early tabular methods without convergence guarantees include Schwartz (1993); Singh (1994). Convergent tabular methods are later on developed by Abounadi et al. (2001); Wan et al. (2021). By contrast, the algorithms presented in this thesis are all convergent with off-policy learning and linear function approximation.

Chapter 14

Conclusions

Bootstrapping, function approximation, and off-policy learning are three arguably indispensable techniques for any large scale RL application. Their combination, however, can result in instability of the resulting algorithms. This is the notorious deadly triad. In this thesis, we have proposed many new algorithms that theoretically address the deadly triad. In many RL settings, we are the first to address the deadly triad. In settings where we are not the first, our proposed algorithms exhibit many benefits over existing ones. That being said, there are still many open problems regarding the deadly triad. In the following we name a few.

- The deadly triad is still not addressed in many RL settings, e.g., policy-based control under the average-reward performance metric and policy-based control with a changing behavior policy. For the former, one possible approach could be to use the density ratio learned via GraidentDICE to reweight the average-reward on-policy actor-critic in Konda and Tsitsiklis (1999). The analysis should be analogous to the analysis of COF-PAC, where the backward critic will be GradientDICE instead of GEM. For the latter, techniques used in analyzing Algorithms 13 and 9 should be helpful.

- Most convergence analysis presented in this thesis is asymptotic, which confirms the stability of the proposed algorithms but provides little information regarding their efficiency. One straightforward future work is to provide finite sample analysis for those algorithms. Existing analysis of temporal difference methods in Dalal et al. (2018); Lakshminarayanan and Szepesvári (2018); Bhandari et al. (2018); Srikant and Ying (2019); Zou et al. (2019), of actor-critic methods in Wu et al. (2020); Xu et al. (2020); Qiu et al. (2021); Huang and Jiang (2021); Xu et al. (2021), and of general stochastic approximation algorithms in Chen et al. (2020b, 2021); Zhang et al. (2021a) should be helpful.

- Optimality is not well addressed in this thesis. Due to the use of linear function approximation, the exact value function is usually not representable. Consequently, for prediction settings, MSPBEs and their variants are used as the objectives for many proposed algorithms in this thesis. There are, however, many different MSPBEs, e.g., the on-policy MSPBE (2.10), the off-policy MSPBE (2.22), and the MSPBEs induced by $f_{\pi,\mu}$ and $f_{\pi,\mu,n}$. It is not clear which one should be used to define optimality. We argue that if we have to use an MSPBE, then the ideal one should be the on-policy MSPBE. This is because we believe that off-policy learning should be considered as a solution technique to achieve some goal. Consequently, it should not be in the goal itself. We, however, do not have any off-policy algorithms that are able to optimize the on-policy MSPBE, which is a possibility for future work. MSPBE is of course not the only objective for prediction. Investigating other objectives and designing corresponding algorithms are also possible future work. For value-based control, algorithms presented in this thesis aim to find TD fixed points or their variants. However, bounding the performance of those TD fixed points is a long-standing open problem, even for the on-policy setting. Even worse, those TD fixed points do not necessarily exist especially when the behavior policy is changing. Investigating their existence under relaxed assumptions is also an important open problem. The optimality of first-order policy-based control methods has been well studied recently in tabular settings (Agarwal et al., 2020; Mei et al., 2020; Laroche and Tachet, 2021; Zhang et al., 2021a). This optimality in the context of the deadly triad is, however, less understood.

- Finite horizon MDPs are not covered. In this thesis, we consider only infinite horizon MDPs. However, many problems are typically modeled as finite horizon MDPs, e.g., Atari games (Bellemare et al., 2013b) and many robot manipulation tasks (Brockman et al., 2016). Finite horizon MDPs are considerably harder than infinite horizon MDPs due to the lack of a discount factor and the dependence of the policy on the time step. And the undiscounted total rewards (Puterman, 2014), instead of the discount total rewards or the average reward, is the dominant performance metric in finite horizon MDPs. Extending the results in this thesis from infinite horizon MDPs to finite horizon MDPs is nontrivial and may require new techniques, which we leave for future work. It is, however, worth mentioning that many algorithms designed for infinite horizon MDPs empirically work well for problems with finite horizon, see, e.g.,

Mnih et al. (2015, 2016); Schulman et al. (2017b). Any many problems with finite horizon can now be alternatively modeled as infinite horizon MDPs via exploiting a transition-dependent discount function instead of a constant discount factor (White, 2017). White (2017); Yu et al. (2018) pioneer the theoretical study of RL algorithms with a discount function. We leave the extension of the results in this thesis to transition-dependent discount functions for future work.

Deep networks have recently been widely used in RL and have enjoyed great empirical success. This thesis, however, considers the deadly triad limited to linear function approximation, which is indeed restrictive. Perhaps the most challenging and fruitful future work is to break the deadly triad with deep networks. Wai et al. (2020) pioneer this direction via analyzing a combination of GTD and an over-parameterized two-layer ReLU (Nair and Hinton, 2010) network, making use of recent advances in over-parameterized networks in the deep learning community (Neyshabur et al., 2018; Allen-Zhu et al., 2019; Allen-Zhu et al., 2019; Zou et al., 2018; Cao and Gu, 2019). GTD is, however, only one of the many algorithms that break the deadly triad in the linear setting. This thesis has proposed several other algorithms that are superior to GTD in various aspects. It is then natural to ask: can those algorithms be combined with deep networks? Cai et al. (2019) provide a hint to this question by combining on-policy TD with an over-parameterized two-layer ReLU network. Central to the analysis of Cai et al. (2019) is a local linearization of the over-parameterized network. Investigating such local linearization in off-policy setting for algorithms proposed in this thesis is therefore a promising research direction. We hope the study in this thesis with linear function approximation can provide useful insight for the local linearization. Cai et al. (2019); Wai et al. (2020) consider only over-parameterized two-layer networks. Practitioners, however, usually use deeper networks with finite width. Investigating the deadly triad with those networks requires new tools to analyze their dynamics, which is a challenging but impactful direction for future work. Besides this theoretical direction, empirically investigating the techniques in this thesis in the context of deep RL is also a possible future work. For example, Jiang et al. (2021) verify the effectiveness of the emphatic methods in deep RL settings. One natural extension would be to use truncated emphatic methods for variance reduction in deep RL settings.

The deadly triad is a long-standing fundamental issue in RL with many open problems from different perspectives. We hope this thesis can serve as a stepping stone towards the ultimate victory against the deadly triad.

Bibliography

Abbasi-Yadkori, Y., Bartlett, P., Bhatia, K., Lazic, N., Szepesvari, C., and Weisz, G. (2019). POLITEX: Regret bounds for policy iteration using expert prediction. In *Proceedings of the International Conference on Machine Learning*.

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Abel, D., Dabney, W., Harutyunyan, A., Ho, M. K., Littman, M., Precup, D., and Singh, S. (2021). On the expressivity of markov reward. In *Advances in Neural Information Processing Systems*.

Abounadi, J., Bertsekas, D., and Borkar, V. S. (2001). Learning algorithms for markov decision processes with average cost. *SIAM Journal on Control and Optimization*.

Achiam, J., Knight, E., and Abbeel, P. (2019). Towards characterizing divergence in deep q-learning. *arXiv preprint arXiv:1903.08894*.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2020). Optimality and approximation with policy gradient methods in markov decision processes. In *Proceedings of the Conference on Learning Theory*.

Allen-Zhu, Z., Li, Y., and Liang, Y. (2019). Learning and generalization in over-parameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems*.

Allen-Zhu, Z., Li, Y., and Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *Proceedings of the International Conference on Machine Learning*.

Asis, K. D., Chan, A., Pitis, S., Sutton, R. S., and Graves, D. (2020). Fixed-horizon temporal difference methods for stable reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Baird, L. C. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the International Conference on Machine Learning*.

Bastani, H., Drakopoulos, K., Gupta, V., Vlachogiannis, J., Hadjicristodoulou, C., Lagiou, P., Magiorkinis, G., Paraskevis, D., and Tsiodras, S. (2021). Efficient and targeted covid-19 border testing via reinforcement learning. *Nature*.

Bellemare, M. G., Candido, S., Castro, P. S., Gong, J., Machado, M. C., Moitra, S., Ponda, S. S., and Wang, Z. (2020). Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013a). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013b). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*.

Bellman, R. (1966). Dynamic programming. *Science*.

Benveniste, A., Métivier, M., and Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximations*. Springer.

Bertsekas, D. and Tsitsiklis, J. (2015). *Parallel and distributed computation: numerical methods*. Athena Scientific.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific Belmont, MA.

Bhandari, J., Russo, D., and Singal, R. (2018). A finite time analysis of temporal difference learning with linear function approximation. In *Proceedings of the Conference on Learning Theory*.

Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., and Lee, M. (2009). Natural actor–critic algorithms. *Automatica*.

Borkar, V. S. (2009). *Stochastic approximation: a dynamical systems viewpoint*. Springer.

Boyan, J. A. (1999). Least-squares temporal difference learning. In *Proceedings of the International Conference on Machine Learning*.

Bradtke, S. J. and Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. (2018). Sample-efficient reinforcement learning with stochastic ensemble value expansion. In *Advances in Neural Information Processing Systems*.

Cai, Q., Yang, Z., Lee, J. D., and Wang, Z. (2019). Neural temporal-difference and q-learning provably converge to global optima. *arXiv preprint arXiv:1905.10027*.

Cao, Y. and Gu, Q. (2019). Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in Neural Information Processing Systems*.

Carvalho, D., Melo, F. S., and Santos, P. (2020). A new convergent variant of q-learning with linear function approximation. In *Advances in Neural Information Processing Systems*.

Chen, S., Devraj, A. M., Lu, F., Busic, A., and Meyn, S. P. (2020a). Zap q-learning with nonlinear function approximation. In *Advances in Neural Information Processing Systems*.

Chen, Z., Maguluri, S. T., Shakkottai, S., and Shanmugam, K. (2020b). Finite-sample analysis of contractive stochastic approximation using smooth convex envelopes. *arXiv preprint arXiv:2002.00874*.

Chen, Z., Maguluri, S. T., Shakkottai, S., and Shanmugam, K. (2021). A lyapunov theory for finite-sample guarantees of asynchronous q-learning and td-learning variants. *arXiv preprint arXiv:2102.01567*.

Chen, Z., Zhang, S., Doan, T. T., Clarke, J.-P., and Maguluri, S. T. (2019). Finite-sample analysis of nonlinear stochastic approximation with applications in reinforcement learning. *arXiv preprint arXiv:1905.11425*.

Ciosek, K. and Whiteson, S. (2020). Expected policy gradients for reinforcement learning. *Journal of Machine Learning Research*.

Dabney, W. and Thomas, P. S. (2014). Natural temporal difference learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Dai, B., Shaw, A., Li, L., Xiao, L., He, N., Liu, Z., Chen, J., and Song, L. (2018). SBEED: convergent reinforcement learning with nonlinear function approximation. In *Proceedings of the International Conference on Machine Learning*.

Dalal, G., Szörényi, B., Thoppe, G., and Mannor, S. (2018). Finite sample analyses for td(0) with function approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Dann, C., Neumann, G., and Peters, J. (2014). Policy evaluation with temporal differences: a survey and comparison. *Journal of Machine Learning Research*.

De Farias, D. P. and Van Roy, B. (2000). On the existence of fixed points for approximate value iteration and temporal-difference learning. *Journal of Optimization Theory and Applications*.

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de las Casas, D., Donner, C., Fritz, L., Galperti, C., Huber, A., Keeling, J., Tsimpoukelli, M., Kay, J., Merle, A., Moret, J.-M., Noury, S., Pesamosca, F., Pfau, D., Sauter, O., Sommariva, C., Coda, S., Duval, B., Fasoli, A., Kohli, P., Kavukcuoglu, K., Hassabis, D., and Riedmiller, M. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*.

Degris, T., White, M., and Sutton, R. S. (2012). Linear off-policy actor-critic. In *Proceedings of the International Conference on Machine Learning*.

Diddigi, R. B., Kamanchi, C., and Bhatnagar, S. (2020). A convergent off-policy temporal difference algorithm. In *Proceedings of the European Conference on Artificial Intelligence*.

Du, S. S., Chen, J., Li, L., Xiao, L., and Zhou, D. (2017). Stochastic variance reduction methods for policy evaluation. In *Proceedings of the International Conference on Machine Learning*.

Du, S. S., Lee, J. D., Mahajan, G., and Wang, R. (2020). Agnostic q-learning with function approximation in deterministic systems: Near-optimal bounds on approximation error and sample complexity. In *Advances in Neural Information Processing Systems*.

Du, S. S., Luo, Y., Wang, R., and Zhang, H. (2019). Provably efficient q-learning with function approximation via distribution shift error checking oracle. In *Advances in Neural Information Processing Systems*.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., and Kavukcuoglu, K. (2018). IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures. In *Proceedings of the International Conference on Machine Learning*.

Fan, J., Wang, Z., Xie, Y., and Yang, Z. (2020). A theoretical analysis of deep q-learning. In *Proceedings of the Annual Conference on Learning for Dynamics and Control*.

Fujimoto, S., van Hoof, H., and Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *Proceedings of the International Conference on Machine Learning*.

Geist, M., Piot, B., and Pietquin, O. (2017). Is the bellman residual a bad proxy? In *Advances in Neural Information Processing Systems*.

Gelada, C. and Bellemare, M. G. (2019). Off-policy deep reinforcement learning by bootstrapping the covariate shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Geramifard, A., Bowling, M., and Sutton, R. S. (2006). Incremental least-squares temporal difference learning. In *Proceedings of the National Conference on Artificial Intelligence*.

Ghiassian, S., Patterson, A., White, M., Sutton, R. S., and White, A. (2018). Online off-policy prediction. *arXiv preprint arXiv:1811.02597*.

Golub, G. H. and Loan, C. F. V. (1996). *Matrix Computations, Third Edition*. Johns Hopkins University Press.

Gordon, G. J. (1995). Stable function approximation in dynamic programming. In *Proceedings of the International Conference on Machine Learning*.

Gordon, G. J. (1996). Chattering in sarsa (lambda)-a cmu learning lab internal report.

Gordon, G. J. (1999). *Approximate solutions to Markov decision processes*. PhD thesis, Carnegie Mellon University.

Gordon, G. J. (2001). Reinforcement learning with function approximation converges to a region. In *Advances in neural information processing systems*.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *Proceedings of the International Conference on Machine Learning*.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning*.

Hallak, A. and Mannor, S. (2017). Consistent on-line off-policy evaluation. In *Proceedings of the International Conference on Machine Learning*.

Hallak, A., Tamar, A., Munos, R., and Mannor, S. (2016). Generalized emphatic temporal difference learning: Bias-variance analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Horn, R. A. and Johnson, C. R. (2012). *Matrix analysis (2nd Edition)*. Cambridge university press.

Huang, J. and Jiang, N. (2021). On the convergence rate of off-policy policy optimization methods with density-ratio correction. *arXiv preprint arXiv:2106.00993*.

Imani, E., Graves, E., and White, M. (2018). An off-policy policy gradient theorem using emphatic weightings. In *Advances in Neural Information Processing Systems*.

Jiang, N. and Li, L. (2016). Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Jiang, R., Zahavy, T., White, A., Xu, Z., Hessel, M., Blundell, C., and van Hasselt, H. (2021). Emphatic algorithms for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Jiang, R., Zhang, S., Chelu, V., White, A., and van Hasselt, H. (2022a). Learning expected emphatic traces for deep rl. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jiang, R., Zhang, S., Chelu, V., White, A., and van Hasselt, H. (2022b). Learning expected emphatic traces for deep RL. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Proceedings of the Conference on Learning Theory*.

Johns, J., Painter-Wakefield, C., and Parr, R. (2010). Linear complementarity for regularized policy evaluation and improvement. In *Advances in Neural Information Processing Systems*.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*.

Kirk, D. E. (2004). *Optimal control theory: an introduction*. Courier Corporation.

Kolter, J. Z. (2011). The fixed points of off-policy TD. In *Advances in Neural Information Processing Systems*.

Kolter, J. Z. and Ng, A. Y. (2009). Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the International Conference on Machine Learning*.

Konda, V. R. (2002). *Actor-Critic Algorithms*. PhD thesis, Massachusetts Institute of Technology.

Konda, V. R. and Tsitsiklis, J. N. (1999). Actor-critic algorithms. In *Advances in Neural Information Processing Systems*.

Kushner, H. and Yin, G. G. (2003). *Stochastic approximation and recursive algorithms and applications*. Springer Science & Business Media.

Lagoudakis, M. G. and Parr, R. (2003). Least-squares policy iteration. *Journal of Machine Learning Research*.

Lakshminarayanan, C. and Szepesvári, C. (2018). Linear stochastic approximation: How far does constant step-size and iterate averaging go? In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Laroche, R. and Tachet, R. (2021). Dr Jekyll and Mr Hyde: the strange case of off-policy policy updates. In *Advances in Neural Information Processing Systems*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*.

Lee, D. and He, N. (2019a). Target-based temporal-difference learning. In *Proceedings of the International Conference on Machine Learning*.

Lee, D. and He, N. (2019b). A unified switching system perspective and ode analysis of q-learning algorithms. *arXiv preprint arXiv:1912.02270*.

Levin, D. A. and Peres, Y. (2017). *Markov chains and mixing times*. American Mathematical Soc.

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Li, L. (2008). A worst-case comparison between temporal difference and residual gradient with linear function approximation. In *Proceedings of the International Conference on Machine Learning*.

Li, L., Littman, M. L., Walsh, T. J., and Strehl, A. L. (2011). Knows what it knows: a framework for self-aware learning. *Machine Learning*.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations*.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. (2015). Finite-sample analysis of proximal gradient TD algorithms. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Liu, B., Mahadevan, S., and Liu, J. (2012). Regularized off-policy td-learning. In *Advances in Neural Information Processing Systems*.

Liu, Q., Li, L., Tang, Z., and Zhou, D. (2018). Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. (2019). Off-policy policy gradient with state distribution correction. *arXiv preprint arXiv:1904.08473*.

Macua, S. V., Chen, J., Zazo, S., and Sayed, A. H. (2015). Distributed policy evaluation under multiple behavior strategies. *IEEE Transactions on Automatic Control*.

Maei, H. R. (2011). *Gradient temporal-difference learning algorithms*. PhD thesis, University of Alberta.

Maei, H. R. (2018). Convergent actor-critic algorithms under off-policy training and function approximation. *arXiv preprint arXiv:1802.07842*.

Maei, H. R. and Sutton, R. S. (2010). Gq (*lambda*): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *Proceedings of the Conference on Artificial General Intelligence*.

Maei, H. R., Szepesvári, C., Bhatnagar, S., and Sutton, R. S. (2010). Toward off-policy learning control with function approximation. In *Proceedings of the International Conference on Machine Learning*.

Mahadevan, S., Liu, B., Thomas, P. S., Dabney, W., Giguere, S., Jacek, N., Gemp, I., and Liu, J. (2014). Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces. *arXiv preprint arXiv:1405.6757*.

Marbach, P. and Tsitsiklis, J. N. (2001). Simulation-based optimization of markov reward processes. *IEEE Transactions on Automatic Control*.

Mathieu, M., Ozair, S., Srinivasan, S., Gulcehre, C., Zhang, S., Jiang, R., Paine, T. L., Zolna, K., Powell, R., Schrittwieser, J., Choi, D., Georgiev, P., Toyama, D. K., Huang, A., Ring, R., Babuschkin, I., Ewalds, T., Bordbar, M., Henderson, S., Colmenarejo, S. G., van den Oord, A., Czarnecki, W. M., de Freitas, N., and Vinyals, O. (2021). Starcraft II unplugged: Large scale offline reinforcement learning. In *Deep RL Workshop NeurIPS 2021*.

Mei, J., Xiao, C., Szepesvári, C., and Schuurmans, D. (2020). On the global convergence rates of softmax policy gradient methods. In *Proceedings of the International Conference on Machine Learning*.

Melo, F. S., Meyn, S. P., and Ribeiro, M. I. (2008). An analysis of reinforcement learning with function approximation. In *Proceedings of the International Conference on Machine Learning*.

Meyn, S. P. and Tweedie, R. L. (2012). *Markov chains and stochastic stability*. Springer Science & Business Media.

Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Nazi, A., et al. (2021). A graph placement methodology for fast chip design. *Nature*.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*.

Morimura, T., Uchibe, E., Yoshimoto, J., Peters, J., and Doya, K. (2010). Derivatives of logarithmic stationary distributions for policy gradient reinforcement learning. *Neural Computation*.

Mousavi, A., Li, L., Liu, Q., and Zhou, D. (2020). Black-box off-policy estimation for infinite-horizon reinforcement learning. In *Proceedings of the International Conference on Learning Representations*.

Munos, R. (2003). Error bounds for approximate policy iteration. In *Proceedings of the International Conference on Machine Learning*.

Nachum, O., Chow, Y., Dai, B., and Li, L. (2019a). Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*.

Nachum, O., Dai, B., Kostrikov, I., Chow, Y., Li, L., and Schuurmans, D. (2019b). Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. (2017). Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. (2018). Trust-pcl: An off-policy trust region method for continuous control. In *Proceedings of the International Conference on Learning Representations*.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning*.

Nemirovski, A., Juditsky, A. B., Lan, G., and Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. (2018). Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*.

O'Donoghue, B., Munos, R., Kavukcuoglu, K., and Mnih, V. (2017). Combining policy gradient and q-learning. In *Proceedings of the International Conference on Learning Representations*.

Painter-Wakefield, C., Parr, R., and Durham, N. (2012). L1 regularized linear temporal difference learning. *Technical report: Department of Computer Science, Duke University, Durham, NC, TR-2012–01*.

Perkins, T. J. and Precup, D. (2002). A convergent form of approximate policy iteration. In *Advances in Neural Information Processing Systems*.

Peters, J. and Schaal, S. (2008). Natural actor-critic. *Neurocomputing*.

Petrik, M., Taylor, G., Parr, R., and Zilberstein, S. (2010). Feature selection using regularization in approximate linear programs for markov decision processes. In *Proceedings of the International Conference on Machine Learning*.

Precup, D., Sutton, R. S., and Dasgupta, S. (2001). Off-policy temporal difference learning with function approximation. In *Proceedings of the International Conference on Machine Learning*.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qiu, S., Yang, Z., Ye, J., and Wang, Z. (2021). On finite-time convergence of actor-critic algorithm. *IEEE Journal on Selected Areas in Information Theory*.

Romoff, J., Henderson, P., Touati, A., Ollivier, Y., Brunskill, E., and Pineau, J. (2019). Separating value functions across time-scales. *arXiv preprint arXiv:1902.01883*.

Rummery, G. A. and Niranjan, M. (1994). *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering Cambridge, UK.

Russell, S. and Norvig, P. (2002). *Artificial intelligence: a modern approach*. Prentice Hall.

Scherrer, B. (2010). Should one compute the temporal difference fix point or minimize the bellman residual? the unified oblique projection view. In *Proceedings of the International Conference on Machine Learning*.

Schoknecht, R. and Merke, A. (2002). Convergent combinations of reinforcement learning with linear function approximation. In *Advances in Neural Information Processing Systems*.

Schoknecht, R. and Merke, A. (2003). TD(0) converges provably faster than the residual gradient algorithm. In *Proceedings of the International Conference on Machine Learning*.

Schulman, J., Abbeel, P., and Chen, X. (2017a). Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017b). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schwartz, A. (1993). A reinforcement learning method for maximizing undiscounted rewards. In *Proceedings of the International Conference on Machine Learning*.

Shapiro, A., Dentcheva, D., and Ruszczynski, A. (2014). *Lectures on Stochastic Programming - Modeling and Theory, Second Edition*.

Sherman, J. and Morrison, W. J. (1950). Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T. P., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. A. (2014). Deterministic policy gradient algorithms. In *Proceedings of the International Conference on Machine Learning*.

Singh, S. P. (1994). Reinforcement learning algorithms for average-payoff markovian decision processes. In *Proceedings of the National Conference on Artificial Intelligence*.

Srikant, R. and Ying, L. (2019). Finite-time error bounds for linear stochastic approximation andtd learning. In *Proceedings of the Conference on Learning Theory*.

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density ratio estimation in machine learning*. Cambridge University Press.

Sun, W. and Bagnell, J. A. (2015). Online bellman residual algorithms with predictive error guarantees. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*.

Sutton, R. S. (1995). Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems*.

Sutton, R. S. (2004). The reward hypothesis. In *http://incompleteideas.net/rlai.cs.ualberta.ca/RLAI/rewardhypothesis.html*.

Sutton, R. S. (2009). The grand challenge of predictive empirical abstract knowledge. In *Working Notes of the IJCAI-09 Workshop on Grand Challenges for Reasoning from Experiences*.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction (2nd Edition)*. MIT press.

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., and Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the International Conference on Machine Learning*.

Sutton, R. S., Mahmood, A. R., and White, M. (2016). An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (1999a). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.

Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. (2011). Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*.

Sutton, R. S., Precup, D., and Singh, S. P. (1999b). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*.

Sutton, R. S., Szepesvári, C., and Maei, H. R. (2008). A convergent o(n) temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems*.

Szepesvári, C. and Smart, W. D. (2004). Interpolation-based q-learning. In *Proceedings of the International Conference on Machine Learning*.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T. P., and Riedmiller, M. A. (2018). Deepmind control suite. *arXiv preprint arXiv:1801.00690*.

Thomas, P. S. (2015). Safe reinforcement learning. *University of Massachusetts Amherst*.

Thomas, P. S., Theocharous, G., and Ghavamzadeh, M. (2015). High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Tikhonov, A. N., Goncharsky, A., Stepanov, V., and Yagola, A. G. (2013). *Numerical methods for the solution of ill-posed problems*. Springer Science & Business Media.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *Proceedings of the International Conference on Intelligent Robots and Systems*.

Tsitsiklis, J. N. and Roy, B. V. (1996a). Analysis of temporal-diffference learning with function approximation. In *IEEE Transactions on Automatic Control*.

Tsitsiklis, J. N. and Roy, B. V. (1996b). Feature-based methods for large scale dynamic programming. *Machine Learning*.

Tsitsiklis, J. N. and Roy, B. V. (1999). Average cost temporal-difference learning. *Automatica*.

Uehara, M., Huang, J., and Jiang, N. (2020). Minimax weight and q-function learning for off-policy evaluation. In *Proceedings of the International Conference on Machine Learning*.

van Hasselt, H., Doron, Y., Strub, F., Hessel, M., Sonnerat, N., and Modayil, J. (2018). Deep reinforcement learning and the deadly triad. *arXiv preprint arXiv:1812.02648*.

van Hasselt, H. P. (2011). *Insights in reinforcement rearning : formal analysis and empirical evaluation of temporal-difference learning algorithms*. PhD thesis, Utrecht University, Netherlands.

Vidyasagar, M. (2002). *Nonlinear systems analysis*. SIAM.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gülçehre, Ç., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T. P., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. (2019). Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nature*.

Wai, H.-T., Yang, Z., Wang, Z., and Hong, M. (2020). Provably efficient neural gtd for off-policy learning. *Advances in Neural Information Processing Systems*.

Wan, Y., Naik, A., and Sutton, R. S. (2021). Learning and planning in average-reward markov decision processes. In *Proceedings of the International Conference on Machine Learning*.

Wang, T., Bowling, M., and Schuurmans, D. (2007a). Dual representations for dynamic programming and reinforcement learning. In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*.

Wang, T., Lizotte, D. J., Bowling, M. H., and Schuurmans, D. (2007b). Stable dual dynamic programming. In *Advances in Neural Information Processing Systems*.

Wang, Y., Chen, W., Liu, Y., Ma, Z., and Liu, T. (2017a). Finite sample analysis of the GTD policy evaluation algorithms in markov setting. In *Advances in Neural Information Processing Systems*.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. (2019). Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*.

Wang, Y. and Zou, S. (2020). Finite-sample analysis of greedy-gq with linear function approximation under markovian noise. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and de Freitas, N. (2017b). Sample efficient actor-critic with experience replay. In *Proceedings of the International Conference on Learning Representations*.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, King's College, Cambridge.

Wen, Z. and Roy, B. V. (2013). Efficient exploration and value function generalization in deterministic systems. In *Advances in Neural Information Processing Systems*.

White, M. (2017). Unifying task specification in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*.

Williams, R. J. and Baird, L. C. (1993). Tight performance bounds on greedy policies based on imperfect value functions. Technical report, Citeseer.

Wu, Y., Zhang, W., Xu, P., and Gu, Q. (2020). A finite-time analysis of two time-scale actor-critic methods. In *Advances in Neural Information Processing Systems*.

Xie, T., Ma, Y., and Wang, Y. (2019). Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. In *Advances in Neural Information Processing Systems*.

Xu, P. and Gu, Q. (2020). A finite-time analysis of q-learning with neural network function approximation. In *Proceedings of the International Conference on Machine Learning*.

Xu, T., Wang, Z., and Liang, Y. (2020). Improving sample complexity bounds for (natural) actor-critic algorithms. In *Advances in Neural Information Processing Systems*.

Xu, T., Yang, Z., Wang, Z., and Liang, Y. (2021). Doubly robust off-policy actor-critic: Convergence and optimality. *arXiv preprint arXiv:2102.11866*.

Yang, L. and Wang, M. (2019). Sample-optimal parametric q-learning using linearly additive features. In *Proceedings of the International Conference on Machine Learning*.

Yang, L. and Wang, M. (2020). Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *Proceedings of the International Conference on Machine Learning*.

Yang, Z., Fu, Z., Zhang, K., and Wang, Z. (2019). Convergent reinforcement learning with function approximation: A bilevel optimization perspective.

Yao, H. and Schuurmans, D. (2013). Reinforcement ranking. *arXiv preprint arXiv:1303.5988*.

Yu, H. (2010). Convergence of least squares temporal difference methods under general conditions. In *Proceedings of the International Conference on Machine Learning*.

Yu, H. (2012). Least squares temporal difference methods: An analysis under general conditions. *SIAM Journal on Control and Optimization*.

Yu, H. (2015). On convergence of emphatic temporal-difference learning. In *Proceedings of the Conference on Learning Theory*.

Yu, H. (2017). On convergence of some gradient-based temporal-differences algorithms for off-policy learning. *arXiv preprint arXiv:1712.09652*.

Yu, H. and Bertsekas, D. P. (2009). Convergence results for some temporal difference methods based on least squares. *IEEE Transactions on Automatic Control*.

Yu, H., Mahmood, A. R., and Sutton, R. S. (2018). On generalized bellman equations and temporal-difference learning. *Journal of Machine Learning Research*.

Zhang, R., Dai, B., Li, L., and Schuurmans, D. (2020a). Gendice: Generalized offline estimation of stationary values. In *Proceedings of the International Conference on Learning Representations*.

Zhang, S., Boehmer, W., and Whiteson, S. (2019). Generalized off-policy actor-critic. In *Advances in Neural Information Processing Systems*.

Zhang, S., Boehmer, W., and Whiteson, S. (2020b). Deep residual reinforcement learning. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*.

Zhang, S., des Combes, R. T., and Laroche, R. (2021a). Global optimality and finite sample analysis of softmax off-policy actor critic under state distribution mismatch. *arXiv preprint arXiv:2111.02997*.

Zhang, S., Laroche, R., van Seijen, H., Whiteson, S., and des Combes, R. T. (2022a). A deeper look at discounting mismatch in actor-critic algorithms. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*.

Zhang, S., Liu, B., and Whiteson, S. (2020c). GradientDICE: Rethinking generalized offline estimation of stationary values. In *Proceedings of the International Conference on Machine Learning*.

Zhang, S., Liu, B., and Whiteson, S. (2021b). Mean-variance policy iteration for risk-averse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhang, S., Liu, B., Yao, H., and Whiteson, S. (2020d). Provably convergent two-timescale off-policy actor-critic with function approximation. In *Proceedings of the International Conference on Machine Learning*.

Zhang, S., Tachet, R., and Laroche, R. (2022b). On the chattering of sarsa with linear function approximation. *arXiv preprint arXiv:2202.06828*.

Zhang, S., Veeriah, V., and Whiteson, S. (2020e). Learning retrospective knowledge with reverse reinforcement learning. In *Advances in Neural Information Processing Systems*.

Zhang, S., Wan, Y., Sutton, R. S., and Whiteson, S. (2021c). Average-reward off-policy policy evaluation with function approximation. In *Proceedings of the International Conference on Machine Learning*.

Zhang, S. and Whiteson, S. (2019). DAC: the double actor-critic architecture for learning options. In *Advances in Neural Information Processing Systems*.

Zhang, S. and Whiteson, S. (2022). Truncated emphatic temporal difference methods for prediction and control. *Journal of Machine Learning Research*.

Zhang, S., Yao, H., and Whiteson, S. (2021d). Breaking the deadly triad with a target network. In *Proceedings of the International Conference on Machine Learning*.

Zhang, S., Zhang, Z., and Maguluri, S. T. (2021e). Finite sample analysis of average-reward td learning and q-learning. *Advances in Neural Information Processing Systems*.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. (2018). Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*.

Zou, S., Xu, T., and Liang, Y. (2019). Finite-sample analysis for SARSA with linear function approximation. In *Advances in Neural Information Processing Systems*.

Appendix A

Stochastic Approximation

A.1 Results from Konda (2002)

Consider a stochastic process $\{Y_t\}$ taking values in a finite space $\mathcal{Y}$. Let $P_\theta \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$ be a parameterized transition kernel in $\mathcal{Y}$, parameterized by θ. Consider iterates $\{w_t\}$ evolving in $\mathbb{R}^K$ according to

$$w_{t+1} \doteq w_t + \alpha_t(h_{\theta_t}(Y_t) - G_{\theta_t}(Y_t)w_t), \tag{A.1}$$

where $h_\theta : \mathcal{Y} \to \mathbb{R}^K$ and $G_\theta : \mathcal{Y} \to \mathbb{R}^{K\times K}$ are vector- and matrix-valued functions parameterized by θ.

Assumption A.1.

$$\Pr(Y_{t+1}|Y_0, \theta_0, w_0, \ldots, Y_t, \theta_t, w_t) = \Pr(Y_{t+1} = y|Y_t, \theta_t) = P_{\theta_t}(Y_t, Y_{t+1})$$

Assumption A.2. *The learning rate sequence $\{\alpha_t\}$ is deterministic, non-increasing, and satisfies*

$$\sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty$$

Assumption A.3. *The (random) sequence of parameters $\{\theta_t\}$ satisfies*

$$\|\theta_{t+1} - \theta_t\| \leq \beta_t H_t,$$

for some nonnegative process $\{H_t\}$ with bounded moments and deterministic sequence $\{\beta_t\}$ such that

$$\sum_t \left(\frac{\beta_t}{\alpha_t}\right)^d < \infty$$

for some $d > 0$.

Assumption A.4. *For each θ, there exist*

$$\bar{h}(\theta) \in \mathbb{R}^K, \bar{G}(\theta) \in \mathbb{R}^{K\times K}, \hat{h}_\theta : \mathcal{Y} \to \mathbb{R}^K, \hat{G}_\theta : \mathcal{Y} \to \mathbb{R}^{K\times K}$$

such that for each $y \in \mathcal{Y}$,

$$\hat{h}_\theta(y) = h_\theta(y) - \bar{h}(\theta) + \sum_{y'} P_\theta(y, y')\hat{h}_\theta(y'),$$
$$\hat{G}_\theta(y) = G_\theta(y) - \bar{G}(\theta) + \sum_{y'} P_\theta(y, y')\hat{G}_\theta(y').$$

Assumption A.5. $\sup_\theta \left\|\bar{h}(\theta)\right\| < \infty, \sup_\theta \left\|\bar{G}(\theta)\right\| < \infty$

Assumption A.6. *For any $y \in \mathcal{Y}$,* $\max\left\{\left\|\hat{h}_\theta(y)\right\|, \|h_\theta(y)\|, \left\|\hat{G}_\theta(y)\right\|, \|G_\theta(y)\|\right\} < \infty$

Assumption A.7. *$\bar{h}(\theta)$ and $\bar{G}(\theta)$ are Lipschitz continuous in θ*

Assumption A.8. *For any $y \in \mathcal{Y}$, $h_\theta(y), \hat{h}_\theta(y), G_\theta(y), \hat{G}_\theta(y)$ are Lipschitz continuous in θ*

Assumption A.9. *There exists some $C_0 > 0$ such that for all w and θ,*

$$w^\top \bar{G}(\theta) w \geq C_0 \|w\|^2$$

Theorem A.1. *Under Assumptions A.1 - A.9, the iterates $\{w_t\}$ generated by (A.1) satisfy*

$$\lim_{t\to\infty} \left\|\bar{h}(\theta_t) - \bar{G}(\theta_t) w_t\right\| = 0 \quad a.s..$$

Theorem A.1 is a simplified version of Theorem 3.2 in Konda (2002) and we refer the reader to Konda (2002) for the detailed proof. In this thesis, we further provide the following corollary for simplifying our proofs.

Assumption A.10. *Let Λ_P be the closure of $\left\{P_\theta \mid \theta \in \mathbb{R}^K\right\}$. For any $P \in \Lambda_P$, the Markov chain induced by P is ergodic. We use d_P to denote the corresponding stationary distribution and write d_θ as shorthand for d_{P_θ}. We define*

$$\bar{G}(\theta) \doteq \sum_{y\in\mathcal{Y}} d_\theta(y) G_\theta(y),$$
$$\bar{h}(\theta) \doteq \sum_{y\in\mathcal{Y}} d_\theta(y) h_\theta(y).$$

Assumption A.11. $\sup_{\theta\in\mathbb{R}^K, y\in\mathcal{Y}} \max\{\|h_\theta(y)\|, \|G_\theta(y)\|\} < \infty$

Assumption A.12. *There exists some constant C_0 such that $\forall\theta, y, y'$,*

$$\begin{aligned}\|h_\theta(y) - h_{\theta'}(y)\| &\le C_0\|\theta - \theta'\|,\\ \|G_\theta(y) - G_{\theta'}(y)\| &\le C_0\|\theta - \theta'\|,\\ \|P_\theta - P_{\theta'}\| &\le C_0\|\theta - \theta'\|.\end{aligned}$$

Corollary A.2. *Let Assumptions A.1 - A.3, A.9, and A.10 - A.12 hold. Then the iterates $\{w_t\}$ generated by (A.1) satisfy*

$$\lim_{t\to\infty} \left\|\bar{h}(\theta_t) - \bar{G}(\theta_t)w_t\right\| = 0 \quad a.s..$$

Proof. It suffices to prove that Assumptions A.10 - A.12 imply Assumptions A.4 - A.8. In this proof we focus on verifying h-related properties. The verification of those G-related properties is the same up to change of notations and is therefore omitted.

For any $\theta \in \mathbb{R}^K$ and any $i \in \{1, \dots, K\}$, consider a Markov Reward Process (MRP) with a transition matrix P_θ and a reward function $h_\theta(y)_i$, where $h_\theta(y)_i$ denotes the i-th element of $h_\theta(y) \in \mathbb{R}^K$. Then the average reward of this MRP is

$$\bar{h}(\theta)_i \doteq \sum_y d_\theta(y) h_\theta(y)_i.$$

In this way, we have defined a vector $\bar{h}(\theta) \in \mathbb{R}^K$ whose i-th element is $\bar{h}(\theta)_i$. We then define

$$\hat{h}_\theta(y) \doteq \mathbb{E}\left[\sum_{k=0}^{\infty} h_\theta(Y_k) - \bar{h}(\theta) \mid Y_0 = y, Y_{k+1} \sim P_\theta(Y_k, \cdot)\right].$$

It is then easy to see that $\hat{h}_\theta(y)_i$, the i-th element of $\hat{h}_\theta(y)$, is exactly the differential value function of this MRP. By the differential Bellman equation, we have

$$\hat{h}_\theta(y)_i = h_\theta(y)_i - \bar{h}(\theta)_i + \sum_{y'} P_\theta(y, y')\hat{h}_\theta(y')_i.$$

Assumption A.4 is then verified.

Assumption A.5 follows immediately from the definition of $\bar{h}(\theta)$ and Assumption A.11.

The differential value function is related to the reward function as

$$\hat{h}_{\theta,i} = (I - P_\theta + 1d_\theta^\top)^{-1}(I - 1d_\theta^\top)h_{\theta,i}, \tag{A.2}$$

see, e.g., (8.2.2) in Puterman (2014) for a proof. Here $\hat{h}_{\theta,i}$ denotes a vector in $\mathbb{R}^{|\mathcal{Y}|}$ whose y-th element is $\hat{h}_\theta(y)_i$ and $h_{\theta,i}$ denotes a vector in $\mathbb{R}^{|\mathcal{Y}|}$ whose y-th element is $h_\theta(y)_i$. Assumption A.10 ensures $\hat{h}_{\theta,i}$ is well defined for each $P_\theta \in \Lambda_P$. Since Λ_P is a compact set, the extreme value theorem asserts that

$$\sup_\theta \left\| (I - P_\theta + 1d_\theta^\top)^{-1}(I - 1d_\theta^\top) \right\| < \infty,$$

which, together with Assumption A.11, verifies Assumption A.6.

Assumption A.7 follows immediately from Assumptions A.12 and A.10 and Lemmas C.1 and C.3.

By the extreme value theorem, we have

$$\sup_{P \in \Lambda_P} \left\| (I - P + 1d_P^\top)^{-1} \right\| < \infty.$$

Then Lemmas C.1, C.2, and C.3 confirm the Lipschitz continuity of $(I - P_\theta + 1d_\theta^\top)^{-1}$ in θ. The Lipschitz continuity of $\hat{h}_\theta(y)$ in Assumption A.8 then follows easily from (A.2), which completes the proof. □

A.2 Results from Borkar (2009)

Consider iterates $\{w_t\}$ evolving in $\mathbb{R}^K$ according to

$$w_{t+1} \doteq w_t + \alpha_t \left(h(w_t) + M_{t+1} + \epsilon_t \right). \tag{A.3}$$

Assumption A.13. *The map $h : \mathbb{R}^K \to \mathbb{R}^K$ is Lipschitz continuous*

Assumption A.14. *The learning rates $\{\alpha_t\}$ are positive scalars satisfying*

$$\sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty$$

Assumption A.15. *$\{M_t\}$ is a martingale difference sequence w.r.t. the increasing family of σ-fields*

$$\mathcal{F}_t \doteq \sigma(w_0, M_1, \ldots, M_t),$$

i.e.,

$$\mathbb{E}[M_{t+1}|\mathcal{F}_t] = 0 \quad a.s. \quad \forall t > 0$$

Further, $\{M_t\}$ is square-integrable with

$$\mathbb{E}[\|M_{t+1}\|^2|\mathcal{F}_t] \leq L(1 + \|w_t\|^2) \quad a.s., \quad \forall t > 0$$

for some constant L.

Assumption A.16. $\lim_{t\to\infty} \|\epsilon_t\| = 0$ *a.s.*

Assumption A.17. $\sup_t \|w_t\| < \infty$ *a.s.*

Theorem A.3. *(The third extension of Theorem 2 in Chapter 2 of Borkar (2009)) Under Assumptions A.13 - A.17, almost surely, the sequence* $\{w_t\}$ *generated by* (A.3) *converges to a compact connected internally chain transitive invariant set of the ODE*

$$\frac{\mathrm{d}}{\mathrm{d}t} w(t) = h(w(t)).$$

In this thesis, we further provide the following corollary for simplifying our proofs. Consider iterates $\{w_t\}$ evolving in $\mathbb{R}^K$ according to

$$w_{t+1} \doteq w_t + \alpha_t \left(G_t w_t + h_t + \epsilon_t\right), \tag{A.4}$$

where $\left\{G_t \in \mathbb{R}^{K\times K}\right\}_{t=0,1,2,\ldots}$, $\left\{h_t \in \mathbb{R}^K\right\}_{t=0,1,2,\ldots}$ are two sequences of i.i.d. random variables. Define

$$\bar{G} \doteq \mathbb{E}\left[G_t\right], \bar{h} \doteq \mathbb{E}\left[h_t\right].$$

Assumption A.18. *There exists a constant* C_0 *such that*

$$\max\left\{\mathbb{E}\left[\left\|G_t - \bar{G}\right\|^2\right], \mathbb{E}\left[\left\|h_t - \bar{h}\right\|^2\right]\right\} < C_0.$$

Assumption A.19. *The real part of every eigenvalue of* $\bar{G}$ *is strictly negative.*

Corollary A.4. *Under Assumptions A.14, A.16, A.18, and A.19, almost surely,*

$$\lim_{t\to\infty} w_t = -\bar{G}^{-1}\bar{h}.$$

Proof. The updates in (A.4) can be rewritten as

$$w_{t+1} = w_t + \alpha_t \left(h(w_t) + M_{t+1} + \epsilon_t\right),$$

where

$$\begin{aligned} h(w) \doteq& \bar{G}w + \bar{h}, \\ M_{t+1} \doteq& G_t w_t + h_t - \bar{G}w_t - \bar{h}. \end{aligned}$$

We then proceed via invoking Theorem A.3.

Assumption A.13 follows immediately from the definition of h.

For Assumption A.15, we have

$$\begin{aligned}
\mathbb{E}\left[M_{t+1}|\mathcal{F}_t\right] =& \mathbb{E}\left[(G_t - \bar{G})w_t|\mathcal{F}_t\right] + \mathbb{E}\left[h_t - \bar{h}|\mathcal{F}_t\right] \\
=& \mathbb{E}\left[G_t - \bar{G}|\mathcal{F}_t\right] w_t \quad (w_t \text{ is adapted by } \mathcal{F}_t) \\
=& 0. \\
\mathbb{E}\left[\|M_{t+1}\|^2 \mid \mathcal{F}_t\right] \leq& 2\mathbb{E}\left[\left\|(G_t - \bar{G})w_t\right\|^2 \mid \mathcal{F}_t\right] + 2\mathbb{E}\left[\left\|h_t - \bar{h}\right\|^2 \mid \mathcal{F}_t\right] \\
\leq& 2\mathbb{E}\left[\left\|G_t - \bar{G}\right\|^2\right] \|w_t\|^2 + 2\mathbb{E}\left[\left\|h_t - \bar{h}\right\|^2\right] \\
\leq& 2C_0\|w_t\|^2 + 2C_0.
\end{aligned}$$

To verify Assumption A.17, we define for $c \geq 1$

$$\begin{aligned}
h_c(w) \doteq& \frac{h(cw)}{c}, \\
h_\infty(w) \doteq& \bar{G}w.
\end{aligned}$$

Then we have $h_c(w) \to h_\infty(w)$ when $c \to \infty$. Assumption A.19 ensures that the ODE

$$\frac{\mathrm{d}w(t)}{\mathrm{d}t} = h_\infty(w)$$

has 0 as its unique globally asymptotically stable equilibrium (see, e.g., Section 5.5 of Vidyasagar (2002)). Theorem 7 in Chapter 3 of Borkar (2009) then asserts that Assumption A.17 holds.

Theorem A.3 then asserts that the iterates $\{w_t\}$ generated by (A.4) converge to a compact connected internally chain transitive invariant set of the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}w(t) = h(w(t)).$$

Assumption A.19 also ensures that the invariant set of the above ODE is a singleton $\left\{-\bar{G}^{-1}\bar{h}\right\}$ (see, e.g., Section 5.5 of Vidyasagar (2002)), which completes the proof. □

A.3 Results from Bertsekas and Tsitsiklis (1996)

Consider the iterates $\{w_t\}$ evolving in $\mathbb{R}^K$ defined as

$$w_{t+1} \doteq w_t + \alpha_t \left(A(Y_t)w_t + b(Y_t)\right),$$

where $\{Y_t\}$ denote a Markov chain in a space $\mathcal{Y}$, $\{\alpha_t\}$ is a sequence of learning rates, A and b are functions from $\mathcal{Y}$ to $\mathbb{R}^{K\times K}$ and $\mathbb{R}^K$ respectively.

Assumption A.20. *$\{\alpha_t\}$ is a deterministic, positive, nonincreasing sequence such that*

$$\sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty.$$

Assumption A.21. *The chain $\{Y_t\}$ is ergodic. We use d_y to denote its stationary distribution and define*

$$\bar{A} \doteq \mathbb{E}_{y\sim d_y(\cdot)}\left[A(y)\right],$$
$$\bar{b} \doteq \mathbb{E}_{y\sim d_y(\cdot)}\left[b(y)\right].$$

Assumption A.22. *The matrix $\bar{A}$ is n.d.*

Assumption A.23. $\sup_{y\in\mathcal{Y}} \|A(y)\| < \infty, \sup_{y\in\mathcal{Y}} \|b(y)\| < \infty$

Theorem A.5. *Let Assumptions A.20 - A.23 hold. Then*

$$\lim_{t\to\infty} w_t = -\bar{A}^{-1}\bar{b} \quad a.s..$$

Theorem A.5 is a simplified version of Proposition 4.8 of Bertsekas and Tsitsiklis (1996) and we, therefore, refer the reader to Bertsekas and Tsitsiklis (1996) for a detailed proof. We note that Assumption 4.5(e) in Bertsekas and Tsitsiklis (1996) results directly from our Assumption A.21 (see, e.g., Theorem 4.9 of Levin and Peres (2017)).

A.4 Results from Benveniste et al. (1990)

Consider the iterates $\{w_t\}$ evolving in $\mathbb{R}^K$ defined as

$$w_{t+1} \doteq w_t + \alpha_t H(w_t, Y_{t+1}), \tag{A.5}$$

where $\{Y_t \in \mathbb{R}^L\}$ are random variables, $\{\alpha_t\}$ is a sequence of learning rates, H is a function from $\mathbb{R}^K \times \mathbb{R}^L$ to $\mathbb{R}^K$. We use $\mathcal{F}_t$ to denote the σ-field generated by $\{w_0, Y_0, Y_1, \ldots, Y_t\}$ and make the following assumptions:

Assumption A.24. *$\{\alpha_t\}$ is a deterministic, positive, nonincreasing sequence such that*

$$\sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty.$$

Assumption A.25. *There exists a family* $\left\{P_w \mid w \in \mathbb{R}^L\right\}$ *of parameterized transition probabilities* P_w *on* $\mathbb{R}^L$ *such that for any* $B \in \mathcal{B}(\mathbb{R}^K)$,

$$\Pr(Y_{t+1} \in B | \mathcal{F}_t) = P_{w_t}(Y_t, B).$$

Additionally, for any function f *defined on* $\mathbb{R}^L$, *we define* $(P_w f)(y) \doteq \int f(x) P_w(y, dx)$. *Here* $\mathcal{B}(\cdot)$ *denotes the Borel sets.*

Assumption A.26. *Let* D *be an open subset of* $\mathbb{R}^K$. *For any compact subset* Q *of* D, *there exists constants* C_1, q_1 *(depending on* Q*), such that for any* $w \in Q$ *and any* y, *we have*

$$\|H(w, y)\| \leq C_1(1 + \|y\|^{q_1}).$$

Assumption A.27. *There exists a function* $h : D \to \mathbb{R}^K$, *and for each* $w \in D$, *a function* $\nu_w : \mathbb{R}^L \to \mathbb{R}^K$, *such that*

(i) h *is locally Lipschitz continuous on* D

(ii) $\nu_w(y) - (P_w \nu_w)(y) = H(w, y) - h(w)$ *holds for all* $w \in D, y \in \mathbb{R}^L$

(iii) for all compact subsets Q *of* D, *there exist constants* C_2, C_3, q_2, q_3 *(depending on* Q*), such that for all* $w, w' \in Q, z \in \mathbb{R}^L$,

$$\|\nu_w(y)\| < C_2(1 + \|y\|^{q_2}),$$
$$\|(P_w \nu_w)(y) - (P_{w'} \nu_{w'})(y)\| \leq C_3 \|w - w'\|(1 + \|y\|^{q_3}).$$

Assumption A.28. *For any compact subset* Q *of* D *and any* $q > 0$, *there exists constant* C_4 *(depending on* Q, q*) such that for all* t, $y \in \mathbb{R}^L, w \in \mathbb{R}^K$,

$$\mathbb{E}\left[\mathbb{I}(\{w_k \in Q, k \leq t\})(1 + \|Y_{t+1}\|^q) \mid Y_0 = y, w_0 = w\right] \leq C_4(1 + \|y\|^q),$$

where $\mathbb{I}$ *is the indicator function.*

Assumption A.29. *There exist a function* $U \in \mathcal{C}^2(\mathbb{R}^K)$ *and* $w_* \in D$ *such that*

(i) $U(w) \to C \leq +\infty$ *if* $w \to \partial D$ *or* $\|w\| \to \infty$

(ii) $U(w) < C$ *for all* $w \in D$

(iii) $U(w) \geq 0$, *where the equality holds i.f.f.* $w = w_*$

(iv) $\left\langle \frac{\mathrm{d}U(w)}{\mathrm{d}w}, h(w) \right\rangle \leq 0$ *for all* $w \in D$*, where the equality holds i.f.f.* $w = w_*$.

Theorem A.6. *(Theorem 13 of Benveniste et al. (1990) (p. 236)) Let Assumptions A.24 - A.29 hold. For any compact* $Q \subset D$*, there exist constants* C_0, q_0 *such that for all* $w \in Q, y \in \mathbb{R}^L$*, the iterates* $\{w_t\}$ *generated by (A.5) satisfy*

$$\Pr\Big(\lim_{t\to\infty} w_t = w_* \mid Y_0 = y, w_0 = w\Big) \geq 1 - C_0(1 + \|y\|^{q_0}) \sum_{t=0}^{\infty} \alpha_t^2.$$

We now consider a special case of Theorem A.6 where the chain is finite. Consider the iterates $\{w_t\}$ evolving in $\mathbb{R}^K$ defined as

$$w_{t+1} \doteq w_t + \alpha_t \bar{H}(w_t, Y_{t+1}), \tag{A.6}$$

where $\{Y_t\}$ are random variables evolving in a *finite* space $\mathcal{Y}$, $\{\alpha_t\}$ is a sequence of learning rates, $\bar{H}$ is a function from $\mathbb{R}^K \times \mathcal{Y}$ to $\mathbb{R}^K$. Without loss of generality, let $\mathcal{Y} \doteq \{1, 2, \ldots, N\} \subset \mathbb{R}$. We make the following assumptions.

Assumption A.30. $\{\alpha_t\}$ *is a deterministic, positive, nonincreasing sequence such that*

$$\sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty.$$

Assumption A.31. *There exists a family* $\left\{\bar{P}_w \in \mathbb{R}^{N\times N} \mid w \in \mathbb{R}^K\right\}$ *of parameterized transition matrices such that the random variables* $\{Y_t\}$ *evolve according to*

$$Y_{t+1} \sim \bar{P}_{w_t}(Y_t, \cdot).$$

Let Λ_w *be the closure of* $\left\{\bar{P}_w \mid w \in \mathbb{R}^K\right\}$*, for any* $P \in \Lambda_w$*, the Markov chain in* $\mathcal{Y}$ *induced by the transition matrix* P *is ergodic. We use* d_P *to denote the invariant distribution of the chain induced by* P*. In particular,* d_w *denotes the invariant distribution of the chain induced by* $\bar{P}_w$*. We define*

$$h(w) \doteq \sum_{y\in\mathcal{Y}} d_w(y)\bar{H}(w, y).$$

Assumption A.32. $\bar{P}_w$ *is Lipschitz continuous in* w*. For any compact* $Q \subset \mathbb{R}^K$ *and any* $y \in \mathcal{Y}$*,* $\bar{H}(w, y)$ *is Lipschitz continuous in* w *on* Q*.*

Assumption A.33. *There exist function* $U \in \mathcal{C}^2(\mathbb{R}^K)$ *and* $w_* \in \mathbb{R}^K$ *such that*

(i) $U(w) \to \infty$ *when* $\|w\| \to \infty$

(ii) $U(w) < \infty$ *for all* $w \in \mathbb{R}^K$

(iii) $U(w) \geq 0$, *where the equality holds i.f.f.* $w = w_*$

(iv) $\left\langle \frac{\mathrm{d}U(w)}{\mathrm{d}w}, h(w) \right\rangle \leq 0$ *for all* $w \in \mathbb{R}^K$, *where the equality holds i.f.f.* $w = w_*$.

Corollary A.7. *Under Assumptions A.30 - A.33, for any compact set* $Q \subset \mathbb{R}^K$, *there exists constants* C_0 *(depending on* Q*) such that for all* $w \in Q, y \in \mathcal{Y}$, *the iterates* $\{w_t\}$ *generated by (A.6) satisfy*

$$\Pr\Big(\lim_{t\to\infty} w_t = w_* \mid Y_0 = y, w_0 = w\Big) \geq 1 - C_0 \sum_{t=0}^{\infty} \alpha_t^2.$$

Proof. We proceed by expressing (A.6) in the form of (A.5) and invoking Theorem A.6. Let

$$H(w, y) \doteq \begin{cases} \bar{H}(w, y) & y \in \mathcal{Y} \\ h(w) & y \notin \mathcal{Y} \end{cases}.$$

Then (A.6) can be rewritten as

$$w_{t+1} \doteq w_t + \alpha_t H(w_t, Y_{t+1}),$$

which has the same form as (A.5). Here the L in $\mathbb{R}^L$ is 1 and we consider D to be $\mathbb{R}^K$.

Assumption A.24 is identical to Assumption A.30. Assumption A.29 is implied by Assumption A.33 via considering $C = \infty$.

To verify Assumption A.25, let

$$P_w(y, B) \doteq \begin{cases} \sum_{y'} \delta_{y'}(B) \bar{P}_w(y, y') & y \in \mathcal{Y} \\ \mathcal{N}(B) & y \notin \mathcal{Y} \end{cases},$$

where $\delta_{y'}(B)$ is the Dirac measure, $\mathcal{N}(\cdot)$ denotes the normal distribution (we can use any well-defined distribution on $\mathbb{R}$ here). Then Assumption A.25 follows from Assumption A.31.

We now verify Assumption A.26. From Assumption A.32 and the finiteness of $\mathcal{Y}$, for any compact Q, $\bar{H}(w, y)$ is bounded on Q. So $h(w)$ is also bounded on Q. Then the boundedness of $H(w, y)$ on Q follows immediately.

We now verify Assumption A.27(i). First, for any compact $Q \subset \mathbb{R}^K$, $\bar{H}(w, y)$ is Lipschitz continuous in w and bounded on Q. $d_w(y)$ is apparently bounded. By Assumption A.31, for any $P \in \Lambda_w$, the chain induced by P is ergodic. It can be easily seen from Lemma C.3 that d_w is also Lipschitz continuous in w. The Lipschitz continuity of $h(w)$ on Q then follows immediately from the fact that the product of two bounded Lipschitz functions are still bounded and Lipschitz. Since we are free to choose any Q, $h(w)$ is locally Lipschitz continuous in $\mathbb{R}^K$.

We verify Assumption A.27(ii) by constructing auxiliary Markov Reward Processes (MRPs) and using standard properties of MRPs. To construct the i-th MRP ($i = 1, \ldots, K$), let $H_{w,i}$ denote a vector in $\mathbb{R}^N$ whose i-th element is $H_i(w, y)$, the i-th element of $H(w, y)$. For any $w \in \mathbb{R}^K, y \in \mathcal{Y}$, we define a vector $\bar{\nu}_w(y)$ in $\mathbb{R}^K$ by defining its i-th element $\bar{\nu}_{w,i}(y)$ as

$$\bar{\nu}_{w,i}(y) \doteq \mathbb{E}\left[\sum_{k=0}^{\infty}[H_{w,i}(Y_k) - h_i(w)] \mid Y_0 = y, Y_{k+1} \sim P_w(Y_k, \cdot)\right],$$

where $h_i(w)$ is the i-the element of $h(w)$. By definition, $\bar{\nu}_{w,i}$ is the differential value function of the MRP induced by $\bar{P}_w$ in $\mathcal{Y}$ with the reward vector being $H_{w,i}$. Since $\bar{P}_w$ induces an ergodic chain under Assumption A.31, $\bar{\nu}_{w,i}$ is always well defined. Moreover, $h_i(w)$ is the average reward of this MRP. It follows from Chapter 8.2.1 of Puterman (2014) that for any $w \in \mathbb{R}^K$ and $y \in \mathcal{Y}$,

$$\begin{aligned} \bar{\nu}_{w,i}(y) &= H_{w,i}(y) - h_i(w) + \sum_{y'} \bar{\nu}_{w,i}(y')\bar{P}_w(y, y') \\ \bar{\nu}_{w,i} &= H_{\bar{P}_w} H_{w,i}, \end{aligned} \tag{A.7}$$

where $H_P \doteq (I - P + 1d_P^\top)^{-1}(I - 1d_P^\top)$ is the fundamental matrix of the chain induced by a transition matrix P. Define

$$\nu_w(y) \doteq \begin{cases} \bar{\nu}_w(y) & y \in \mathcal{Y} \\ 0 & y \notin \mathcal{Y} \end{cases}.$$

It is then easy to verify that for $y \in \mathcal{Y}$,

$$(P_w\nu_w)(y) = \int \nu_w(x)P_w(y, dx) = \int \nu_w(x)\sum_{y'} \delta_{y'}(dx)\bar{P}_w(y, y') = \sum_{y'} \nu_w(y')\bar{P}_w(y, y').$$

For $y \notin \mathcal{Y}$, $(P_w\nu_w)(y) = 0$. For $y \in \mathcal{Y}$, Assumption A.27(ii) holds since it is just (A.7). For $y \notin \mathcal{Y}$, Assumption A.27(ii) holds as well since since both LHS and RHS are 0.

We now verify Assumption A.27(iii). Since d_P is Lipschitz continuous in P for all $P \in \Lambda_w$ and Λ_w is compact, we have $\sup_{P\in\Lambda_w} \left\|(I - P + 1d_P^\top)^{-1}\right\| < \infty$ by the extreme value theorem. Using the ergodicity of the chain induced by P and Lemma C.2, it is then easy to see that $H_{\bar{P}_w}$ is bounded and Lipschitz continuous in w. Consequently, for any compact $Q \subset \mathbb{R}^K, w \in Q, w' \in Q$, ν_w is bounded on Q and

$$\begin{aligned}\|\bar{\nu}_{w,i} - \bar{\nu}_{w',i}\| &\leq \left\|H_{\bar{P}_w} - H_{\bar{P}_{w'}}\right\| \|H_{w,i}\| + \left\|H_{\bar{P}_{w'}}\right\| \|H_{w,i} - H_{w',i}\| \\ &\leq C_1\|w - w'\| + C_2\|w - w'\|,\end{aligned}$$

where the constant C_1 results from the Lipschitz continuity of $H_{\bar{P}_w}$ and the boundedness of $H(w, y)$ on Q, the constant C_2 results from the Lipschitz continuity of $H(w, y)$ on Q and the boundedness of $H_{\bar{P}_w}$. Since by Assumption A.32, $\bar{P}_w$ in Lipschitz continuous in w, the Lipschitz continuity of $P_w\nu_w$ in Q follows immediately, which completes the verification of Assumption A.27(iii).

Assumption A.28 is trivial since $\mathcal{Y}$ is finite, which completes the proof. □

Appendix B

Proofs

B.1 Proof of Theorem 3.1

Proof. Similar to Chapter 5.4 of Borkar (2009), we consider $\dot{\Gamma}_{B_1}$, the directional derivative of Γ_{B_1}. At a point $x \in \mathbb{R}^K$, given a direction $y \in R^K$, we have

$$\begin{aligned}\dot{\Gamma}_{B_1}(x,y) &\doteq \lim_{\delta\to 0}\frac{\Gamma_{B_1}(x+\delta y)-\Gamma_{B_1}(x)}{\delta}\\ &= \begin{cases} y, & x \in int(B_1)\\ y, & x \in \partial B_1, y \in F_x(B_1)\\ -\frac{xx^\top y}{\|x\|^3}+\frac{y}{\|x\|}, & \text{otherwise}\end{cases}\end{aligned}$$

where $int(B_1)$ is the interior of B_1, ∂B_1 is the boundary of B_1,

$$F_x(B_1) \doteq \left\{y \in \mathbb{R}^K \mid \exists \delta > 0, \quad \text{s.t.} \quad x+\delta y \in B_1\right\}$$

is the feasible directions of B_1 w.r.t. x. The first two cases are trivial and are easy to deal with. The third case is complicated and is the source of the reflection term $\zeta(t)$ in (3.4). However, thanks to the projection Γ_{B_2}, we succeeded in getting rid of it.

By (3.2), $\theta_t \in B_1$ always holds. With the directional derivative, we can rewrite the update rule of $\{\theta_t\}$ as

$$\begin{aligned}\theta_{t+1} &= \Gamma_{B_1}\big(\theta_t+\beta_t(\Gamma_{B_2}(w_t)-\theta_t)\big)\\ &= \theta_t+\beta_t\frac{\Gamma_{B_1}\big(\theta_t+\beta_t(\Gamma_{B_2}(w_t)-\theta_t)\big)-\Gamma_{B_1}(\theta_t)}{\beta_t}\\ &= \theta_t+\beta_t\big(\dot{\Gamma}_{B_1}(\theta_t,\Gamma_{B_2}(w_t)-\theta_t)+o(\beta_t)\big) \quad \text{(Definition of limit)} \quad .\end{aligned}$$

We now compute $\dot{\Gamma}_{B_1}(\theta_t,\Gamma_{B_2}(w_t)-\theta_t)$. We proceed by showing that only the first two cases in $\dot{\Gamma}_{B_1}(x,y)$ can happen and the third case will never occur.

For $\theta_t \in \text{int}(B_1)$, we have

$$\dot{\Gamma}_{B_1}(\theta_t, \Gamma_{B_2}(w_t) - \theta_t) = \Gamma_{B_2}(w_t) - \theta_t. \tag{B.1}$$

For $\theta_t \in \partial B_1$, we have

$$\langle \theta_t, \Gamma_{B_2}(w_t) - \theta_t \rangle = \langle \theta_t, \Gamma_{B_2}(w_t) \rangle - R_{B_1}^2 \leq R_{B_1} R_{B_2} - R_{B_1}^2 < 0. \tag{B.2}$$

Let $y_0 \doteq \Gamma_{B_2}(w_t) - \theta_t$, (B.2) implies that we can decompose y_0 as $y_0 = y_1 + y_2$, where $\langle \theta_t, y_1 \rangle = 0$ and $\langle \theta_t, y_2 \rangle = -\|\theta_t\| \|y_2\|$. Here y_2 is the projection of y_0 onto θ_t, which is in the opposite direction of θ_t because their inner product is negative and y_1 is the remaining orthogonal component. By Pythagoras's theorem, for any $\delta > 0$,

$$\begin{aligned} \|\theta_t + \delta y_0\|^2 &= \|\delta y_1\|^2 + \|\theta_t + \delta y_2\|^2 \\ &= \delta^2 \|y_1\|^2 + \|\theta_t\|^2 - 2\delta \|\theta_t\| \|y_2\| + \delta^2 \|y_2\|^2. \end{aligned}$$

For sufficiently small δ, e.g., $\delta^2 \|y_1\|^2 - 2\delta \|\theta_t\| \|y_2\| + \delta^2 \|y_2\|^2 < 0$, we have

$$\|\theta_t + \delta y_0\|^2 < \|\theta_t\|^2 = R_{B_1}^2,$$

implying $\Gamma_{B_2}(w_t) - \theta_t \in F_{\theta_t}(B_1)$. So we have

$$\dot{\Gamma}_{B_1}(\theta_t, \Gamma_{B_2}(w_t) - \theta_t) = \Gamma_{B_2}(w_t) - \theta_t. \tag{B.3}$$

Combining (B.1) and (B.3) yields

$$\begin{aligned} \theta_{t+1} &= \theta_t + \beta_t \big(\Gamma_{B_2}(w_t) - \theta_t + o(\beta_t) \big) \\ &= \theta_t + \beta_t \left(\Gamma_{B_2}(w^*(\theta_t)) - \theta_t + (\Gamma_{B_2}(w_t) - \Gamma_{B_2}(w^*(\theta_t))) + o(\beta_t) \right) \\ &= \theta_t + \beta_t \left((w^*(\theta_t) - \theta_t + (\Gamma_{B_2}(w_t) - \Gamma_{B_2}(w^*(\theta_t))) + o(\beta_t) \right) \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Assumption 3.2)} \\ &= \theta_t + \left(h(\theta_t) + \epsilon_t \right), \end{aligned}$$

where

$$\begin{aligned} h(\theta) &\doteq w^*(\theta) - \theta, \\ \epsilon_t &\doteq \Gamma_{B_2}(w_t) - \Gamma_{B_2}(w^*(\theta_t)) + o(\beta_t). \end{aligned}$$

We now proceed by invoking Theorem A.3. Assumption A.13 is verified by Assumption 3.3. Assumption A.14 is verified by Assumption 2.5. Assumptions A.15 hold immediately because in this setting we consider we have $M_t \equiv 0$. Assumption A.16

hold thanks to Assumption 3.1 and the continuity of Γ_{B_2}. Assumption A.17 is verified directly by the projection in (3.2). By Theorem A.3, we then have, almost surely, the iterates $\{\theta_t\}$ converge to a compact connected internally chain transitive invariant set of the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta(t) = w^*(\theta(t)) - \theta(t).$$

Under Assumption 3.3, the Banach fixed-point theorem asserts that there is a unique θ^* satisfying $w^*(\theta^*) = \theta^*$, i.e., θ^* is the unique equilibrium of the ODE above. We now show θ^* is globally asymptotically stable. Consider the candidate Lyapunov function

$$V(\theta) \doteq \frac{1}{2}\|\theta - \theta^*\|^2.$$

We have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}V(\theta(t)) =& \left\langle \theta(t) - \theta^*, \frac{\mathrm{d}}{\mathrm{d}t}\theta(t) \right\rangle \\
=& \langle \theta(t) - \theta^*, w^*(\theta(t)) - \theta(t) \rangle \\
=& \langle \theta(t) - \theta^*, w^*(\theta(t)) - \theta(t) - w^*(\theta^*) + \theta^* \rangle \\
\leq& \varrho\|\theta(t) - \theta^*\|^2 - \|\theta(t) - \theta^*\|^2,
\end{aligned}$$

where $\varrho < 1$ is the Lipschitz constant of w^*. It is easy to see

- $V(\theta) \geq 0$
- $V(\theta) = 0 \iff \theta = \theta^*$
- $\frac{\mathrm{d}}{\mathrm{d}t}V(\theta(t)) \leq 0$
- $\frac{\mathrm{d}}{\mathrm{d}t}V(\theta(t)) = 0 \iff \theta = \theta^*$

Consequently, θ^* is globally asymptotically stable, and the invariant set of the ODE is a singleton $\{\theta^*\}$. We, therefore, have

$$\begin{aligned}
&\lim_{t\to\infty} \theta_t = \theta^*, \\
&\lim_{t\to\infty} w_t = \lim_{t\to\infty} w^*(\theta_t) = w^*(\theta^*) = \theta^*.
\end{aligned}$$

□

B.2 Proof of Lemma 3.2

Proof. By (3.2), $\theta_t \in B_1$ holds for all t. So

$$\begin{aligned}
&\|\theta_{t+1} - \theta_t\| \\
=&\left\|\Gamma_{B_1}\Big(\theta_t + \beta_t(\Gamma_{B_2}(w_t) - \theta_t)\Big) - \Gamma_{B_1}(\theta_t)\right\| \\
\leq&\beta_t\|\Gamma_{B_2}(w_t) - \theta_t\| \quad \text{(Nonexpansiveness of projection)} \\
\leq&\beta_t(R_{B_1} + R_{B_2}).
\end{aligned}$$

□

B.3 Proof of Theorem 3.3

Proof. Consider the Markov process $Y_t \doteq (S_t, A_t, S_{t+1})$ involving in the space

$$\mathcal{Y} \doteq \{(s, a, s') \mid s \in \mathcal{S}, a \in \mathcal{A}, s' \in \mathcal{S}, p(s'|s, a) > 0\}.$$

By Assumption 2.8, Y_t adopts a unique stationary distribution, which we refer to as $d_{\mathcal{Y}}$. We have $d_{\mathcal{Y}}(s, a, s') = d_\mu(s)\mu(a|s)p(s'|s, a)$. We define

$$\begin{aligned}
h_\theta(s, a, s') &\doteq \left(r(s, a) + \gamma \sum_{a'} \pi(a'|s')x(s', a')^\top \Gamma_{B_1}(\theta)\right) x(s, a), \\
G_\theta(s, a, s') &\doteq x(s, a)x(s, a)^\top + \eta I.
\end{aligned}$$

As $\theta_t \in B_1$ holds for all t, we can rewrite the update of $\{w_t\}$ in Algorithm 1 as

$$w_{t+1} = w_t + \alpha_t(h_{\theta_t}(Y_t) - G_{\theta_t}(Y_t)w_t).$$

The asymptotic behavior of $\{w_t\}$ is then governed by

$$\begin{aligned}
\bar{h}(\theta) &\doteq \mathbb{E}_{(s,a,s')\sim d_{\mathcal{Y}}(\cdot)}[h_\theta(s, a, s')] \\
&= X^\top D_\mu r + \gamma X^\top D_\mu P_\pi X \Gamma_{B_1}(\theta), \\
\bar{G}(\theta) &\doteq \mathbb{E}_{(s,a,s')\sim d_{\mathcal{Y}}}(\cdot)[G_\theta(s, a, s')] \\
&= X^\top D_\mu X + \eta I.
\end{aligned}$$

Define

$$w^*(\theta) \doteq \bar{G}(\theta)^{-1}\bar{h}(\theta) = (X^\top D_\mu X + \eta I)^{-1} X^\top D_\mu (r + \gamma P_\pi X \Gamma_{B_1}(\theta)). \tag{B.4}$$

We now verify Assumptions 3.1, 3.2, and 3.3 to invoke Theorem 3.1.

We first verify Assumption 3.1 via invoking Corollary A.2. The P_θ in Section A.1 is now a constant function in our prediction setting defined as

$$P_\theta((s_1,a_1,s_1'),(s_2,a_2,s_2')) \doteq \begin{cases} \mu(a_2|s_2)p(s_2'|s_2,a_2), & s_1' = s_2 \\ 0, & \text{otherwise} \end{cases}.$$

Assumption A.1 follows from the definition of Algorithm 1. Assumption A.2 follows from Assumption 2.4. Assumption A.3 follows from Assumption 2.5 and Lemma 3.2. Assumption A.9 holds because Assumption 2.3 and $\eta > 0$. Assumption A.10 holds because L_P is now a singleton. Assumptions A.11 and A.12 is self-evident according to the definition of h_θ and G_θ. Invoking Corollary A.2 then verifies Assumption 3.1.

To verify Assumption 3.3, we use SVD and get

$$D_\mu^{\frac{1}{2}} X = U^\top \Sigma V,$$

where U, V are two orthogonal matrices and $\Sigma \doteq diag([\dots,\sigma_i,\dots])$ is a diagonal matrix. Assumptions 2.8 and 2.3 imply that $\sigma_i > 0$. We have

$$\begin{aligned} &\left\|(X^\top D_\mu X + \eta I)^{-1} X^\top D_\mu^{\frac{1}{2}}\right\| \\ =&\left\|V^\top(\Sigma^2+\eta I)^{-1}\Sigma U\right\| \\ =&\left\|(\Sigma^2+\eta I)^{-1}\Sigma\right\| \\ =&\left\|diag\left(\left[\cdots,\frac{\sigma_i}{\sigma_i^2+\eta},\cdots\right]\right)\right\| \\ =&\max_i \frac{1}{\sigma_i+\eta/\sigma_i} \\ \le&\frac{1}{2\sqrt{\eta}} \end{aligned} \tag{B.5}$$

According to (B.4), it is then easy to see

$$\begin{aligned} &\|w^*(\theta_1) - w^*(\theta_2)\| \\ \le&\frac{\gamma}{2\sqrt{\eta}}\left\|D_\mu^{\frac{1}{2}} P_\pi\right\|\|X\|\|\Gamma_{B_1}(\theta_1) - \Gamma_{B_1}(\theta_2)\| \\ \le&\frac{\gamma}{2\sqrt{\eta}}\left\|D_\mu^{\frac{1}{2}} P_\pi\right\|\|X\|\|\theta_1 - \theta_2\| \\ \le&\frac{\gamma}{2\sqrt{\eta}}\left\|D_\mu^{\frac{1}{2}} P_\pi D_\mu^{-\frac{1}{2}}\right\|\left\|D_\mu^{\frac{1}{2}}\right\|\|X\|\|\theta_1 - \theta_2\| \\ =&\frac{\gamma}{2\sqrt{\eta}}\|P_\pi\|_{d_\mu}\left\|D_\mu^{\frac{1}{2}}\right\|\|X\|\|\theta_1 - \theta_2\| \quad \text{(Using (C.3))} \end{aligned}.$$

Take any $\xi \in (0,1)$, assuming

$$\|X\| \le \frac{2(1-\xi)\sqrt{\eta}}{\gamma\|P_\pi\|_{d_\mu}\max_{s,a}\sqrt{d_\mu(s,a)}}, \tag{B.6}$$

then

$$\|w^*(\theta_1) - w^*(\theta_2)\| \leq (1-\xi)\|\theta_1 - \theta_2\|.$$

Assumption 3.3, therefore, holds.

We now select proper R_{B_1} and R_{B_2} to fulfill Assumption 3.2. Plugging (B.5) and (B.6) into (B.4) yields

$$\begin{aligned}\|w^*(\theta)\| &\leq \frac{1}{2\sqrt{\eta}}\left\|D_\mu^{\frac{1}{2}} r\right\| + (1-\xi)R_{B_1} \\ &= R_{B_1} - \xi + (\frac{1}{2\sqrt{\eta}}\left\|D_\mu^{\frac{1}{2}} r\right\| + \xi - \xi R_{B_1})\end{aligned}$$

For sufficiently large R_{B_1}, e.g.,

$$R_{B_1} \geq \frac{1}{2\xi\sqrt{\eta}}\left\|D_\mu^{\frac{1}{2}} r\right\| + 1, \tag{B.7}$$

we have $\sup_\theta \|w^*(\theta)\| \leq R_{B_1} - \xi$. Selecting $R_{B_2} \in (R_{B_1} - \xi, R_{B_1})$ then fulfills Assumption 3.2.

Now Theorem 3.1 then implies that there exists a unique θ_∞ such that

$$w^*(\theta_\infty) = \theta_\infty \quad \text{and} \quad \lim_{t\to\infty} \theta_t = \lim_{t\to\infty} w_t = \theta_\infty.$$

Next we show what θ_∞ is. We define

$$f(\theta) \doteq (X^\top D_\mu X + \eta I)^{-1} X^\top D_\mu (r + \gamma P_\pi X\theta).$$

Note this is just the right side of equation (B.4) without the projection. (B.5) and (B.6) imply that f is a contraction. The Banach fixed-point theorem then asserts that f adopts a unique fixed point, which we refer to as w_η^*. We have

$$\begin{aligned}\left\|w_\eta^*\right\| = \left\|f(w_\eta^*)\right\| &\leq \frac{1}{2\sqrt{\eta}}(\|r\| + \gamma\|P_\pi\|_{d_\mu}\left\|D_\mu^{\frac{1}{2}}\right\|\|X\|\left\|w_\eta^*\right\|) \\ &\leq \frac{\|r\|}{2\sqrt{\eta}} + (1-\xi)\left\|w_\eta^*\right\| \quad \text{(Using (B.6))} \\ \implies \left\|w_\eta^*\right\| &\leq \frac{\|r\|}{2\xi\sqrt{\eta}}\end{aligned}$$

Then for sufficiently large R_{B_1}, e.g.,

$$R_{B_1} \geq \frac{\|r\|}{2\xi\sqrt{\eta}}, \tag{B.8}$$

we have $w_\eta^* = \Gamma_{B_1}(w_\eta^*)$, implying w_η^* is a fixed point of $w^*(\cdot)$ (i.e., the right side of (B.4)) as well. As $w^*(\cdot)$ is a contraction, we have $\theta_\infty = w_\eta^*$. Rewriting $f(w_\eta^*) = w_\eta^*$ yields

$$Aw_\eta^* - \eta w_\eta^* + b = 0.$$

In other words, w_η^* is the unique (due to the contraction of f) solution of

$$(A - \eta I)w + b = 0$$

Combining (B.6), (B.7), and (B.8), the desired constants are

$$C_0 \doteq \frac{2(1-\xi)\sqrt{\eta}}{\gamma \|P_\pi\|_{d_\mu} \max_{s,a} \sqrt{d_\mu(s,a)}},$$
$$C_1 \doteq \frac{\|r\|}{2\xi\sqrt{\eta}} + 1.$$

We now bound $\left\|Xw_\eta^* - q_\pi\right\|$. For any $y \in \mathbb{R}^{|\mathcal{S}|}$, we define the ridge regularized projection $\Pi_{d_\mu}^\eta$ as

$$\begin{aligned}\Pi_{d_\mu}^\eta y &\doteq X \arg\min_w \left(\|Xw - y\|_{D_\mu}^2 + \eta\|w\|^2\right) \\ &= X(X^\top D_\mu X + \eta I)^{-1} X^\top D_\mu y.\end{aligned} \tag{B.9}$$

$\Pi_{d_\mu}^\eta$ is connected with f as $\Pi_{d_\mu}^\eta \mathcal{T}_\pi(Xw) = Xf(w)$. We have

$$\begin{aligned}\left\|Xw_\eta^* - q_\pi\right\| &\leq \left\|Xw_\eta^* - \Pi_{d_\mu}^\eta q_\pi\right\| + \left\|\Pi_{d_\mu}^\eta q_\pi - q_\pi\right\| \\ &= \left\|Xf(w_\eta^*) - \Pi_{d_\mu}^\eta q_\pi\right\| + \left\|\Pi_{d_\mu}^\eta q_\pi - q_\pi\right\| \\ &= \left\|\Pi_{d_\mu}^\eta \mathcal{T}_\pi(Xw_\eta^*) - \Pi_{d_\mu}^\eta \mathcal{T}_\pi q_\pi\right\| + \left\|\Pi_{d_\mu}^\eta q_\pi - q_\pi\right\| \\ &= \left\|\Pi_{d_\mu}^\eta \gamma P_\pi X w_\eta^* - \Pi_{d_\mu}^\eta \gamma P_\pi q_\pi\right\| + \left\|\Pi_{d_\mu}^\eta q_\pi - q_\pi\right\| \\ &= \left\|\gamma \Pi_{d_\mu}^\eta P_\pi (Xw_\eta^* - q_\pi)\right\| + \left\|\Pi_{d_\mu}^\eta q_\pi - q_\pi\right\| \\ &\leq \left\|\gamma X(X^\top D_\mu X + \eta I)^{-1} X^\top D_\mu P_\pi\right\| \left\|Xw_\eta^* - q_\pi\right\| + \left\|\Pi_{d_\mu}^\eta q_\pi - q_\pi\right\| \\ &\leq (1-\xi)\left\|Xw_\eta^* - q_\pi\right\| + \left\|\Pi_{d_\mu}^\eta q_\pi - q_\pi\right\| \quad \text{(Using (B.5) and (B.6))} \quad .\end{aligned}$$

The above equation implies

$$\left\|Xw_\eta^* - q_\pi\right\| \leq \frac{1}{\xi}\left\|\Pi_{d_\mu}^\eta q_\pi - q_\pi\right\| \leq \frac{1}{\xi}\left(\left\|\Pi_{d_\mu}^\eta q_\pi - \Pi_{d_\mu} q_\pi\right\| + \left\|\Pi_{d_\mu} q_\pi - q_\pi\right\|\right).$$

We now bound $\left\|\Pi^{\eta}_{d_\mu} - \Pi_{d_\mu}\right\|$.

$$\begin{aligned}
\left\|\Pi^{\eta}_{d_\mu} - \Pi_{d_\mu}\right\| &\leq \|X\|\left\|(X^\top D_\mu X + \eta I)^{-1} - (X^\top D_\mu X)^{-1}\right\|\left\|X^\top D_\mu\right\| \qquad \text{(B.10)}\\
&= \left\|D_\mu^{-\frac{1}{2}} D_\mu^{\frac{1}{2}} X\right\|\left\|V^\top\left((\Sigma^2+\eta I)^{-1} - \Sigma^{-2}\right)V\right\|\left\|X^\top D_\mu^{\frac{1}{2}} D_\mu^{\frac{1}{2}}\right\|\\
&\leq \left\|D_\mu^{-\frac{1}{2}}\right\|\|\Sigma\|\left\|(\Sigma^2+\eta I)^{-1} - \Sigma^{-2}\right\|\|\Sigma\|\left\|D_\mu^{\frac{1}{2}}\right\|\\
&\leq \left\|D_\mu^{-\frac{1}{2}}\right\|\|\Sigma\|\left\|diag\left(\left[\dots, \frac{\eta}{\sigma_i^2(\eta+\sigma_i^2)}, \dots\right]\right)\right\|\|\Sigma\|\left\|D_\mu^{\frac{1}{2}}\right\|\\
&\leq \left\|D_\mu^{-\frac{1}{2}}\right\|\|\Sigma\| \max_i \frac{\eta}{\sigma_i^2(\eta+\sigma_i^2)}\|\Sigma\|\left\|D_\mu^{\frac{1}{2}}\right\|\\
&\leq \left\|D_\mu^{-\frac{1}{2}}\right\|\|\Sigma\|^2 \max_i \frac{\eta}{\sigma_i^4}\left\|D_\mu^{\frac{1}{2}}\right\|\\
&\leq \left\|D_\mu^{-\frac{1}{2}}\right\|\frac{\sigma_{\max}(\Sigma)^2}{\sigma_{\min}(\Sigma)^4}\left\|D_\mu^{\frac{1}{2}}\right\|\eta
\end{aligned}$$

($\sigma_{\max}(\cdot)$ and $\sigma_{\min}(\cdot)$ indicate the largest and smallest singular values)

$$\begin{aligned}
&= \frac{\sigma_{\max}(D_\mu^{\frac{1}{2}}X)^2}{\sigma_{\min}(D_\mu^{\frac{1}{2}}X)^4}\frac{\sigma_{\max}(D_\mu^{\frac{1}{2}})}{\sigma_{\min}(D_\mu^{\frac{1}{2}})}\eta\\
&\leq \frac{\sigma_{\max}(X)^2}{\sigma_{\min}(X)^4\sigma_{\min}(D_\mu)^{2.5}}\eta
\end{aligned}$$

(Using $\sigma_{\max}(XY) \leq \sigma_{\max}(X)\sigma_{\max}(Y); \sigma_{\min}(XY) \geq \sigma_{\min}(X)\sigma_{\min}(Y)$)

.

Finally, we arrive at

$$\left\|Xw^*_\eta - q_\pi\right\| \leq \frac{1}{\xi}\Big(\frac{\sigma_{\max}(X)^2}{\sigma_{\min}(X)^4\sigma_{\min}(D_\mu)^{2.5}}\|q_\pi\|\eta + \left\|\Pi_{d_\mu}q_\pi - q_\pi\right\|\Big),$$

which completes the proof. □

B.4 Proof of Proposition 4.1

Proof. First, $\hat{\mathcal{T}}_{\pi,\mu}m_{\pi,\mu} = m_{\pi,\mu}$ follows directly from (4.4).

Given any two square matrices A and B, the products AB and BA have the same eigenvalues (see, e.g., Theorem 1.3.22 in Horn and Johnson (2012)). Let $\rho(\cdot)$ denote the spectral radius. We then have

$$\rho(\gamma D_\mu^{-1}P_\pi^\top D_\mu) = \rho\big((\gamma P_\pi^\top D_\mu)D_\mu^{-1}\big) = \rho(\gamma P_\pi^\top) = \rho(\gamma P_\pi) < 1.$$

Clearly $\gamma D_\mu^{-1}P_\pi^\top D_\mu$ is a nonnegative matrix. Then Lemma C.4 implies that $\hat{\mathcal{T}}_{\pi,\mu}$ is a contraction mapping w.r.t. some weighted maximum norm. □

B.5 Proof of Theorem 4.2

Proof. Define

$$
\begin{aligned}
w_k \doteq& \begin{bmatrix} \kappa_k \\ \nu_k \end{bmatrix}, \\
G_k \doteq& \begin{bmatrix} -x'_k x'^{\top}_k & x'_k(\gamma\rho_k x_k - x'_k)^{\top} \\ (x'_k - \gamma\rho_k x_k)x'^{\top}_k & 0 \end{bmatrix}, \\
h_k \doteq& \begin{bmatrix} i'_k x'_k \\ 0 \end{bmatrix}.
\end{aligned}
$$

Then the updates to $\{\kappa_k\}, \{\nu_k\}$ in Algorithm 2 can be expressed as

$$
w_{k+1} = w_k + \alpha_k(G_k w_k + h_k).
$$

We now proceed via invoking Corollary A.4. It can be easily computed that

$$
\begin{aligned}
\bar{G} \doteq& \mathbb{E}\left[G_k\right] = \begin{bmatrix} -C_\mu & A^{\top}_{\pi,\mu} \\ -A_{\pi,\mu} & 0 \end{bmatrix}, \\
\bar{h} \doteq& \mathbb{E}\left[h_k\right] = \begin{bmatrix} X^{\top} D_\mu i \\ 0 \end{bmatrix}.
\end{aligned}
$$

Assumption A.18 holds because we consider a finite MDP so

$$
\sup_k \max\left\{\|G_k\|, \|h_k\|\right\} < \infty.
$$

It remains to verify Assumption A.19. To this end, we follow the routine of Sutton et al. (2009). According to Lemma C.6, we have

$$
\det\left(\bar{G}\right) = \det(C_\mu)\det\left(A_{\pi,\mu}C^{-1}_\mu A^{\top}_{\pi,\mu}\right) \neq 0.
$$

This means that all eigenvalues of $\bar{G}$ are nonzero. Let $\lambda \in \mathbb{C}$ be an eigenvalue of $\bar{G}$ and $y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \in \mathbb{C}^{2K}$ be the corresponding normalized eigenvector, i.e., $y^H y = 1$, where y^H denotes the complex conjugate of y. We then have

$$
\begin{aligned}
\lambda =& y^H y = y^H \bar{G} y \\
=& - y_1^H C_\mu y_1 - y_2^H A_{\pi,\mu} y_1 + y_1^H A^{\top}_{\pi,\mu} y_2.
\end{aligned}
$$

Let $\mathrm{Re}(\cdot)$ denote the real part of a complex number. We then have

$$
\mathrm{Re}\left(y_2^H A_{\pi,\mu} y_1\right) = \mathrm{Re}\left(\left(y_2^H A_{\pi,\mu} y_1\right)^H\right) = \mathrm{Re}\left(y_1^H A^{\top}_{\pi,\mu} y_2\right).
$$

Consequently, we have

$$\mathrm{Re}(\lambda) = \mathrm{Re}(-y_1^H C_\mu y_1) \leq 0.$$

Since $\lambda \neq 0$, we have

$$\mathrm{Re}(\lambda) < 0.$$

Assumption A.19 is then fulfilled.

Invoking Corollary A.4 then asserts that

$$\lim_{k\to\infty} w_k = -\bar{G}^{-1}\bar{h} \quad \text{a.s.} \quad .$$

Using block matrix inversion yields

$$\lim_{k\to\infty} \nu_k = -\left(A_{\pi,\mu}^\top\right)^{-1} X^\top D_\mu i.$$

It is easy to verify that

$$\begin{aligned} & L(v) = 0 \\ \iff & X^\top D_\mu \left(i + \gamma D_\mu^{-1} P_\pi^\top D_\mu X\nu - X\nu\right) = 0 \\ \iff & A_{\pi,\mu}^\top \nu + X^\top D_\mu i = 0, \end{aligned}$$

which completes the proof. □

B.6 Proof of Proposition 4.3

Proof. The proof follows a similar routine as the proof of Theorem 2 in Kolter (2011). We include the full proof for completeness. First notice that

$$\begin{aligned} & \left\|\Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu X\nu\right\|_{d_\mu} \leq \|X\nu\|_{d_\mu} \\ \iff & \nu^\top X^\top D_\mu P_\pi X C_\mu^{-1} X^\top P_\pi^\top D_\mu X\nu \leq \nu^\top C_\mu \nu \\ \iff & \nu^\top \left(X^\top D_\mu P_\pi X C_\mu^{-1} X^\top P_\pi^\top D_\mu X - C_\mu\right)\nu \leq 0. \end{aligned} \tag{B.11}$$

Assumption 4.1 and a property of Schur complement ensure that (B.11) holds for any ν. We then have

$$\begin{aligned} & \|X\nu_* - m_{\pi,\mu}\|_{d_\mu} \\ \leq & \left\|X\nu_* - \Pi_{d_\mu} m_{\pi,\mu}\right\|_{d_\mu} + \left\|\Pi_{d_\mu} m_{\pi,\mu} - m_{\pi,\mu}\right\|_{d_\mu} \\ = & \left\|\Pi_{d_\mu}\hat{\mathcal{T}}_{\pi,\mu}(X\nu_*) - \Pi_{d_\mu}\hat{\mathcal{T}}_{\pi,\mu} m_{\pi,\mu}\right\|_{d_\mu} + \left\|\Pi_{d_\mu} m_{\pi,\mu} - m_{\pi,\mu}\right\|_{d_\mu} \\ = & \gamma\left\|\Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu(X\nu_*) - \Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu m_{\pi,\mu}\right\|_{d_\mu} + \left\|\Pi_{d_\mu} m_{\pi,\mu} - m_{\pi,\mu}\right\|_{d_\mu} \\ \leq & \gamma\left\|\Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu(X\nu_*) - \Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu \Pi_{d_\mu} m_{\pi,\mu}\right\|_{d_\mu} \\ & + \gamma\left\|\Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu \Pi_{d_\mu} m_{\pi,\mu} - \Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu m_{\pi,\mu}\right\|_{d_\mu} + \left\|\Pi_{d_\mu} m_{\pi,\mu} - m_{\pi,\mu}\right\|_{d_\mu}. \end{aligned}$$

For the first term, there exists a $\bar{\nu}$ such that $X\bar{\nu} = \Pi_{d_\mu} m_{\pi,\mu}$. Consequently, we have

$$\begin{aligned}
&\left\|\Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu (X\nu_*) - \Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu \Pi_{d_\mu} m_{\pi,\mu}\right\|_{d_\mu} \\
=&\left\|\Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu X(\nu_* - \bar{\nu})\right\|_{d_\mu} \\
\leq&\left\|X(\nu_* - \bar{\nu})\right\|_{d_\mu} \\
=&\left\|\Pi_{d_\mu} X\nu_* - \Pi_{d_\mu} m_{\pi,\mu}\right\|_{d_\mu} \\
\leq&\left\|X\nu_* - m_{\pi,\mu}\right\|_{d_\mu}.
\end{aligned}$$

For the second term, we have

$$\begin{aligned}
&\left\|\Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu \Pi_{d_\mu} m_{\pi,\mu} - \Pi_{d_\mu} D_\mu^{-1} P_\pi^\top D_\mu m_{\pi,\mu}\right\|_{d_\mu} \\
\leq&\left\|D_\mu^{-1} P_\pi^\top D_\mu\right\|_{d_\mu} \left\|\Pi_{d_\mu} m_{\pi,\mu} - m_{\pi,\mu}\right\|_{d_\mu}.
\end{aligned}$$

Putting them together yields

$$\begin{aligned}
&(1-\gamma)\left\|X\nu_* - m_{\pi,\mu}\right\|_{d_\mu} \\
\leq&\left(1 + \left\|D_\mu^{-1} P_\pi^\top D_\mu\right\|_{d_\mu}\right) \left\|\Pi_{d_\mu} m_{\pi,\mu} - m_{\pi,\mu}\right\|_{d_\mu} \\
=&\left(1 + \left\|P_\pi\right\|_{d_\mu}\right) \left\|\Pi_{d_\mu} m_{\pi,\mu} - m_{\pi,\mu}\right\|_{d_\mu} \quad \text{(Lemma C.11)} \quad ,
\end{aligned}$$

which completes the proof. □

B.7 Proof of Theorem 4.4

Proof. Define $Y_t \doteq (S_t, A_t, S_{t+1})$ and

$$\mathcal{Y} \doteq \{(s, a, s') \mid s \in \mathcal{S}, a \in \mathcal{A}, s' \in \mathcal{S}, \mu(a|s) > 0, p(s'|s, a) > 0\}.$$

It is easy to see that the chain $\{Y_t\}$ is ergodic and its stationary distribution is $d_{\mathcal{Y}}(s, a, s') \doteq d_\mu(s)\mu(a|s)p(s'|s, a)$. The the update of $\{w_t\}$ in Algorithm 3 can be expressed as

$$w_{t+1} = w_t + \alpha_t \left(A(Y_t) w_t + b(Y_t)\right),$$

where

$$\begin{aligned}
A((s, a, s')) &\doteq \left(x(s)^\top \nu_*\right) \frac{\pi(a|s)}{\mu(a|s)} x(s) \left(\gamma x(s') - x(s)\right)^\top \\
b((s, a, s')) &\doteq \left(x(s)^\top \nu_*\right) \frac{\pi(a|s)}{\mu(a|s)} x(s) r(s, a).
\end{aligned}$$

It can then be computed that

$$\bar{A} \doteq \mathbb{E}_{(s,a,s')\sim d_{\mathcal{Y}}(\cdot)}\left[A((s,a,s'))\right] = X^\top D_\mu D_{\nu_*}(\gamma P_\pi - I)X,$$
$$\bar{b} \doteq \mathbb{E}_{(s,a,s')\sim d_{\mathcal{Y}}(\cdot)}\left[b(s,a,s')\right] = X^\top D_\mu D_{\nu_*} r_\pi.$$

We now proceed via invoking Theorem A.5, which requires to verify Assumption A.22. Under Assumption 2.3, $\bar{A}$ is n.d. if and only if

$$D_\mu D_{\nu_*}(\gamma P_\pi - I)$$

is n.d. The above matrix is n.d. if and only if

$$D_\mu D_{\nu_*}(\gamma P_\pi - I) + (\gamma P_\pi^\top - I) D_{\nu_*} D_\mu$$

is n.d. Sutton et al. (2016) show that $D_{f_{\pi,\mu}}(\gamma P_\pi - I)$ is n.d.. Consequently, by the definition of $f_{\pi,\mu}$, we know that

$$D_\mu D_{m_{\pi,\mu}}(\gamma P_\pi - I)$$

is n.d., where $D_{m_{\pi,\mu}} \doteq diag(m_{\pi,\mu})$. This implies that

$$D_\mu D_{m_{\pi,\mu}}(\gamma P_\pi - I) + (\gamma P_\pi^\top - I) D_{m_{\pi,\mu}} D_\mu$$

is n.d. Let $\lambda_{min} > 0$ denote the minimum eigenvalue of

$$D_\mu D_{m_{\pi,\mu}}(I - \gamma P_\pi) + (I - \gamma P_\pi^\top) D_{m_{\pi,\mu}} D_\mu.$$

We then have, for any z,

$$z^\top \left(D_\mu D_{m_{\pi,\mu}}(I - \gamma P_\pi) + (I - \gamma P_\pi^\top) D_{m_{\pi,\mu}} D_\mu\right) z \geq \lambda_{min} \|z\|^2.$$

Consequently,

$$\begin{aligned}
&z^\top \left(D_\mu D_{\nu_*}(I - \gamma P_\pi) + (I - \gamma P_\pi^\top) D_{\nu_*} D_\mu\right) z \\
=&z^\top \left(D_\mu D_{m_{\pi,\mu}}(I - \gamma P_\pi) + (I - \gamma P_\pi^\top) D_{m_{\pi,\mu}} D_\mu\right) z \\
&+ z^\top \left(D_\mu (D_{\nu_*} - D_{m_{\pi,\mu}})(I - \gamma P_\pi) + (I - \gamma P_\pi^\top)(D_{\nu_*} - D_{m_{\pi,\mu}}) D_\mu\right) z \\
\geq&\lambda_{min}\|z\|^2 - 2\|D_\mu\|\|I - \gamma P_\pi\|\left\|D_{\nu_*} - D_{m_{\pi,\mu}}\right\|\|z\|^2 \\
\geq&\left(\lambda_{min} - 2\max_s d_\mu(s) \max_s \left|x(s)^\top \nu_* - m_{\pi,\mu}(s)\right| \|I - \gamma P_\pi\|\right) \|z\|^2.
\end{aligned}$$

This implies that as long as

$$\max_s \left|x(s)^\top \nu_* - m_{\pi,\mu}(s)\right| < \epsilon_1 \doteq \frac{\lambda_{min}}{2\max_s d_\mu(s) \|I - \gamma P_\pi\|},$$

the matrix $D_\mu D_{\nu_*}(I - \gamma P_\pi) + (I - \gamma P_\pi^\top) D_{\nu_*} D_\mu$ is p.d., i.e., Assumption A.22 holds. Theorem A.5 then asserts that

$$\lim_{t \to \infty} w_t = w_\infty.$$

We now proceed to bounding $\|X w_\infty - v_\pi\|$. We have

$$\|X w_\infty - v_\pi\| \leq \|X w_\infty - X w_*\| + \|X w_* - v_\pi\|,$$

where

$$w_* \doteq - \left(X^\top D_\mu D_{m_{\pi,\mu}} (\gamma P_\pi - I)^{-1} X \right) X^\top D_\mu D_{m_{\pi,\mu}} r_\pi.$$

Hallak et al. (2016) show that

$$\|X w_* - v_\pi\| = \mathcal{O}\left(\left\| \Pi_{f_{\pi,\mu}} v_\pi - v_\pi \right\|_{f_{\pi,\mu}} \right).$$

It thus remains to bound $\|w_\infty - w_*\|$. Define

$$\Lambda \doteq \gamma P_\pi - I.$$

We have

$$\begin{aligned}
&\|w_\infty - w_*\| \\
\leq &\left\| \left(X^\top D_\mu D_{m_{\pi,\mu}} \Lambda X \right)^{-1} - \left(X^\top D_\mu D_{\nu_*} \Lambda X \right)^{-1} \right\| \left\| X^\top D_\mu D_{m_{\pi,\mu}} r_\pi \right\| \\
&+ \left\| \left(X^\top D_\mu D_{\nu_*} \Lambda X \right)^{-1} \right\| \left\| X^\top D_\mu D_{m_{\pi,\mu}} r_\pi - X^\top D_\mu D_{\nu_*} r_\pi \right\| \\
\leq &\left\| \left(X^\top D_\mu D_{m_{\pi,\mu}} \Lambda X \right)^{-1} \right\| \left\| \left(X^\top D_\mu D_{\nu_*} \Lambda X \right)^{-1} \right\| \|X\|^2 \|D_\mu\| \|\Lambda\| \left\| D_{m_{\pi,\mu}} - D_{\nu_*} \right\| \\
&\times \left\| X^\top D_\mu D_{m_{\pi,\mu}} r_\pi \right\| \\
&+ \left\| \left(X^\top D_\mu D_{\nu_*} \Lambda X \right)^{-1} \right\| \|X\|^2 \|D_\mu\| \left\| D_{m_{\pi,\mu}} - D_{\nu_*} \right\| \|r_\pi\| \quad \text{(Lemma C.2)} \\
= &\left\| \left(X^\top D_\mu D_{\nu_*} \Lambda X \right)^{-1} \right\| \mathcal{O}(\epsilon_1).
\end{aligned}$$

To bound $\left\| \left(X^\top D_\mu D_{\nu_*} \Lambda X \right)^{-1} \right\|$, we use Corollary 8.6.2 of Golub and Loan (1996), which states that for any two matrices Y_1 and Y_2,

$$|\sigma_{min}(Y_1 + Y_2) - \sigma_{min}(Y_1)| \leq \|Y_2\|,$$

where $\sigma_{\min}(\cdot)$ denotes the minimum singular value. Define

$$\epsilon_2 \doteq \frac{\sigma_{min}\left(X^\top D_\mu D_{m_{\pi,\mu}} \Lambda X \right)}{2 \|X\|^2 \|D_\mu\| \|\Lambda\|} > 0.$$

If

$$\max_s \left| x(s)^\top \nu_* - m_{\pi,\mu}(s) \right| < \epsilon_2,$$

we have

$$\begin{aligned}
&\left\| \left(X^\top D_\mu D_{\nu_*} \Lambda X \right)^{-1} \right\| \\
=&\frac{1}{\sigma_{min}\left(X^\top D_\mu D_{\nu_*} \Lambda X \right)} \\
=&\frac{1}{\sigma_{min}\left(X^\top D_\mu D_{m_{\pi,\mu}} \Lambda X + X^\top D_\mu (D_{\nu_*} - D_{m_{\pi,\mu}}) \Lambda X \right)} \\
\leq&\frac{1}{\sigma_{min}\left(X^\top D_\mu D_{m_{\pi,\mu}} \Lambda X \right) - \left\| X^\top D_\mu (D_{\nu_*} - D_{m_{\pi,\mu}}) \Lambda X \right\|} \\
\leq&\frac{1}{\sigma_{min}\left(X^\top D_\mu D_{m_{\pi,\mu}} \Lambda X \right) - \|X\|^2 \|D_\mu\| \|\Lambda\| \max_s |x(s)^\top \nu_* - m_{\pi,\mu}(s)|} \\
\leq&\frac{2}{\sigma_{min}\left(X^\top D_\mu D_{m_{\pi,\mu}} \Lambda X \right)}.
\end{aligned}$$

Setting $\epsilon \doteq \min\{\epsilon_1, \epsilon_2\}$ then completes the proof. □

B.8 Proof of Theorem 4.5

Proof. The proof is a combination of the proof of Theorem 4.1 and the proof of Lemma 5.1 up to change of notations and is thus omitted to avoid verbatim repetition. □

B.9 Proof of Proposition 4.6

Proof. The proof is a simplified version of the proof of Proposition 4.4 and is thus omitted to avoid verbatim repetition. □

B.10 Proof of Lemma 5.1

Proof. Let $\tau_j \doteq (s_j, a_j, s_{j-1}, a_{j-1}, \dots s_1, a_1)$, $\Gamma_j \doteq (S_{t-j}, A_{t-j}, \dots, S_{t-1}, A_{t-1})$,

$$\begin{aligned}
m_{t,n}(s) &\doteq \mathbb{E}[F_t^n | S_t = s] \qquad \text{(B.12)} \\
&= \sum_{j=0}^{n} \gamma^j \mathbb{E}\left[\rho_{t-j:t-1} i_{t-j} \middle| S_t = s \right] \\
&= \sum_{j=0}^{n} \gamma^j \sum_{\tau_j \in (\mathcal{S}\times\mathcal{A})^j} \Pr(\Gamma_j = \tau_j | S_t = s) \mathbb{E}[\rho_{t-j:t-1} i_{t-j} | \Gamma_j = \tau_j, S_t = s]
\end{aligned}$$

(Law of total expectation)

$$= \sum_{j=0}^{n} \gamma^j \sum_{\tau_j \in (\mathcal{S}\times\mathcal{A})^j} \frac{\Pr(\Gamma_j = \tau_j, S_t = s)}{\Pr(S_t = s)} \mathbb{E}[\rho_{t-j:t-1} i_{t-j} | \Gamma_j = \tau_j, S_t = s]$$

(Bayes' rule)

$$= \sum_{j=0}^{n} \gamma^j \sum_{\tau_j \in (\mathcal{S}\times\mathcal{A})^j} \frac{\Pr(\Gamma_j = \tau_j, S_t = s)}{\Pr(S_t = s)} i(s_j)\rho(s_j, a_j)\cdots\rho(s_1, a_1)$$

$$= \sum_{j=0}^{n} \gamma^j \sum_{\tau_j \in (\mathcal{S}\times\mathcal{A})^j} \frac{\Pr(S_{t-j} = s_j) P_\pi(s_j, s_{j-1})\cdots P_\pi(s_2, s_1) P_\pi(s_1, s)}{\Pr(S_t = s)} i(s_j)$$

$$= \sum_{j=0}^{n} \gamma^j \sum_{s_j} \frac{\Pr(S_{t-j} = s_j) P_\pi^j(s_j, s)}{\Pr(S_t = s)} i(s_j)$$

Assumption 2.8 implies

$$\lim_{t\to\infty} \Pr(S_t = s) = d_\mu(s).$$

Consequently,

$$\begin{aligned} m_{\pi,\mu,n}(s) &= \lim_{t\to\infty} m_{t,n}(s) \\ &= \sum_{j=0}^{n} \gamma^j \sum_{s_j \in \mathcal{S}} \frac{d_\mu(s_j) P_\pi^j(s_j, s)}{d_\mu(s)} i(s_j). \end{aligned}$$

In a matrix form,

$$m_{\pi,\mu,n} = \sum_{j=0}^{n} \gamma^j D_\mu^{-1} (P_\pi^\top)^j D_\mu i,$$

$$\lim_{n\to\infty} m_{\pi,\mu,} = D_\mu^{-1}(I - \gamma P_\pi^\top)^{-1} D_\mu i.$$

□

B.11 Proof of Lemma 5.2

Proof.

$$\begin{aligned} m_{\pi,\mu,n} - m_{\pi,\mu} &= \sum_{j=n+1}^{\infty} \gamma^j D_\mu^{-1} (P_\pi^\top)^j D_\mu i \\ &= D_\mu^{-1} \left(\sum_{j=n+1}^{\infty} \gamma^j (P_\pi^\top)^j \right) D_\mu i \\ &= D_\mu^{-1} \gamma^{n+1} (P_\pi^\top)^{n+1} (I - \gamma P_\pi^\top)^{-1} D_\mu i \\ &= \gamma^{n+1} D_\mu^{-1} (P_\pi^\top)^{n+1} D_\mu m_{\pi,\mu}, \end{aligned}$$

implying

$$\begin{aligned}
\|m_{\pi,\mu,n} - m_{\pi,\mu}\|_1 &\leq \gamma^{n+1}\left\|D_\mu^{-1}\right\|_1 \|D_\mu\|_1 \|m_{\pi,\mu}\|_1 \quad (\text{Using } \left\|P_\pi^\top\right\|_1 = \|P_\pi\|_\infty = 1) \\
&= \gamma^{n+1}\frac{d_{\mu,max}}{d_{\mu,min}}\|m_{\pi,\mu}\|_1, \\
\|f_n - f\|_\infty &\leq \|D_\mu\|_\infty \|m_{\pi,\mu,n} - m_{\pi,\mu}\|_\infty \leq d_{\mu,max}\|m_{\pi,\mu,n} - m_{\pi,\mu}\|_1 \\
&\leq d_{\mu,max}\|m_{\pi,\mu,n} - m_{\pi,\mu}\|_\infty \\
&= \gamma^{n+1}\frac{d_{\mu,max}^2}{d_{\mu,min}}\|m_{\pi,\mu}\|_1
\end{aligned}$$

□

B.12 Proof of Lemma 5.3

Proof. In this proof, we use f_n and f as shorthand for $f_{\pi,\mu,n}$ and $f_{\pi,\mu}$ for easing presentation. Let D_{f_n} be a diagonal matrix whose diagonal entry is f_n and $D_{f_n - f}$ be a diagonal matrix whose diagonal entry is $f_n - f$. We have

$$\begin{aligned}
y^\top D_{f_n}(\gamma P_\pi - I)y &= y^\top D_{f_n - f}(\gamma P_\pi - I)y + y^\top D_f(\gamma P_\pi - I)y \\
&\leq \|y\|^2 \|D_{f_n - f}\| \|\gamma P_\pi - I\| + y^\top D_f(\gamma P_\pi - I)y \\
&\leq \|y\|^2 \|f_n - f\|_\infty \|\gamma P_\pi - I\| + y^\top D_f(\gamma P_\pi - I)y
\end{aligned}$$

Similarly,

$$y^\top(\gamma P_\pi^\top - I)D_{f_n}y \leq \|y\|^2\|f_n - f\|_\infty\left\|\gamma P_\pi^\top - I\right\| + y^\top(\gamma P_\pi^\top - I)D_f y.$$

Combining the above two inequalities together, we get

$$\begin{aligned}
&\frac{1}{2}y^\top\left(D_{f_n}(\gamma P_\pi - I) + (\gamma P_\pi^\top - I)D_{f_n}\right)y \\
\leq &\|f_n - f\|_\infty \|\gamma P_\pi - I\| \|y\|^2 + \frac{1}{2}y^\top\left(D_f(\gamma P_\pi - I) + (\gamma P_\pi^\top - I)D_f\right)y \\
&\qquad\qquad (\text{Invariance of } \ell_2 \text{ norm under transpose}) \\
\leq &(\|f_n - f\|_\infty \|\gamma P_\pi - I\| - \lambda_{min})\|y\|^2 \\
&\qquad\qquad (\text{Eigendecomposition of real symmetric matrices}) \\
\leq &(\gamma^{n+1}\frac{d_{\mu,max}^2}{d_{\mu,min}}\|m\|_1\|\gamma P_\pi - I\| - \lambda_{min})\|y\|^2 \quad (\text{Lemma } 5.2)
\end{aligned}$$

As long as the condition (5.4) holds, the above inequality asserts that

$$D_{f_n}(\gamma P_\pi - I) + (\gamma P_\pi^\top - I) D_{f_n}$$

is n.d., implying $D_{f_n}(\gamma P_\pi - I)$ is n.d.. This together with Assumption 2.3 completes the proof. □

B.13 Proof of Theorem 5.4

Proof. In this proof, we use f_n as shorthand for $f_{\pi,\mu,n}$ for easing presentation. Let $Y_t \doteq (S_{t-n}, A_{t-n}, \ldots, S_t, A_t, S_{t+1})$ be a sequence of random variables generated by Algorithm 4. Let $y_t \doteq (s_{t-n}, a_{t-n}, \ldots, s_t, a_t, s_{t+1})$ and define functions

$$A(y_t) \doteq \left(\sum_{j=0}^{n} \gamma^j \left(\prod_{k=t-j}^{t-1} \frac{\pi(a_k|s_k)}{\mu(a_k|s_k)} \right) i(s_{t-j}) \right) \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)} x(s_t) \left(\gamma x(s_{t+1})^\top - x(s_t)^\top \right),$$

$$b(y_t) \doteq \left(\sum_{j=0}^{n} \gamma^j \left(\prod_{k=t-j}^{t-1} \frac{\pi(a_k|s_k)}{\mu(a_k|s_k)} \right) i(s_{t-j}) \right) \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)} x(s_t) r(s_t, a_t).$$

Here y_t is just placeholder for defining $A(\cdot)$ and $b(\cdot)$. Then the update for $\{w_t\}$ in Algorithm 4 can be expressed as

$$w_{t+1} = w_t + \alpha_t \left(A(Y_t) w_t + b(Y_t) \right).$$

We now proceed to confirming its convergence via verifying Assumptions A.20 - A.23 thus invoking Theorem A.5.

Assumption A.20 is identical to Assumption 2.4. Assumption A.21 follows directly from Assumption 2.8. And it is easy to see the invariant distribution of $\{Y_t\}$ is

$$d_{\mathcal{Y}}(y_t) = d_\mu(s_{t-n}) \mu(a_{t-n}|s_{t-n}) p(s_{t-n+1}|s_{t-n}, a_{t-n}) \cdots p(s_{t+1}|s_t, a_t). \tag{B.13}$$

Moreover,

$$\begin{aligned}
\bar{A} \doteq & \mathbb{E}_{Y_t \sim d_{\mathcal{Y}}(\cdot)} \left[A(Y_t) \right] \\
= & \mathbb{E}_{Y_t \sim d_{\mathcal{Y}}(\cdot)} \left[F_{t,n} \rho_t x_t (\gamma x_{t+1}^\top - x_t^\top) \right] \\
= & \sum_{s,a,s'} d_\mu(s) \mu(a|s) p(s'|s,a) \mathbb{E} \left[F_{t,n} \rho_t x_t (\gamma x_{t+1}^\top - x_t^\top) \middle| S_t = s, A_t = a, S_{t+1} = s' \right] \\
& \qquad \text{(Law of total expectation)} \\
= & \sum_{s,a,s'} d_\mu(s) \pi(a|s) p(s'|s,a) \mathbb{E} \left[F_{t,n} | S_t = s \right] x(s) (\gamma x(s')^\top - x(s)^\top)
\end{aligned}$$

(Conditional independence and Markov property)

$$= \sum_{s,a,s'} d_\mu(s)\pi(a|s)p(s'|s,a)m_{\pi,\mu,n}(s)x(s)(\gamma x(s')^\top - x(s)^\top) \quad \text{(Using (B.12) and (B.13))}$$

$$= \sum_{s,a,s'} f_n(s)\pi(a|s)p(s'|s,a)x(s)(\gamma x(s')^\top - x(s)^\top) \quad \text{(Definition of } f_n\text{)}$$

$$= X^\top D_{f_n}(\gamma P_\pi - I)X.$$

In the above equation, we have abused the notation slightly to use Y_t to denote random variables sampled from $d_\mathcal{Y}$. Similarly, it can be shown that

$$\bar{b} \doteq \mathbb{E}_{Y_t \sim d_\mathcal{Y}(\cdot)}[b(Y_t)] = X^\top D_{f_n} r_\pi.$$

Lemma 5.3 confirms that $\bar{A}$ is n.d., verifying Assumption A.22. Assumption A.23 is obvious since $|\mathcal{S}|, |\mathcal{A}|, n$ are all finite, which then completes the proof.

Note this procedure cannot be used to verify the convergence of the original ETD(0), where we would need to consider $Y_t = (F_t, S_t, A_t)$. Since F_t involves in $\mathbb{R}$, Assumption A.23 cannot be verified. □

B.14 Proof of Lemma 5.6

Proof. In this proof, we use f, f_n, m, m_n as shorthand for $f_{\pi,\mu}, f_{\pi,\mu,n}, m_{\pi,\mu}, m_{\pi,\mu,n}$ for easing presentation. Since $i(s) > 0$ holds for any s and P_π is nonnegative, from Lemma 5.1 it is easy to see for any $n_1 > n_2$,

$$m_{n_1}(s) > m_{n_2}(s)$$

always holds. Then by the definition of f_n,

$$f_{n_1}(s) > f_{n_2}(s)$$

holds as well. In particular, for any $n \geq 1$,

$$f(s) > f_n(s) > f_0(s) = d_\mu(s)i(s) > 0.$$

For any v, we have

$$
\begin{aligned}
\gamma\|P_\pi v\|_{f_n}^2 &= \gamma \sum_s f_n(s) \left(\sum_{s'} P_\pi(s,s') v(s') \right)^2 \\
&\leq \gamma \sum_s f_n(s) \sum_{s'} P_\pi(s,s') v^2(s') \quad \text{(Jensen's inequality)} \\
&= \gamma \sum_{s'} v^2(s') \sum_s f_n(s) P_\pi(s,s') \\
&= v^\top diag(\gamma P_\pi^\top f_n) v \\
&= v^\top diag\left(f_n - (I - \gamma P_\pi^\top) f_n\right) v \\
&= v^\top diag\left(f_n - (I - \gamma P_\pi^\top) f + (I - \gamma P_\pi^\top)(f - f_n)\right) v \\
&= \|v\|_{f_n}^2 - v^\top diag\left((I - \gamma P_\pi^\top) f\right) v + v^\top diag\left((I - \gamma P_\pi^\top)(f - f_n)\right) v \\
&= \|v\|_{f_n}^2 - \|v\|_{f_0}^2 + v^\top diag\left((I - \gamma P_\pi^\top)(f - f_n)\right) v \\
&\qquad\qquad \text{(Using } (I - \gamma P_\pi^\top) f = (I - \gamma P_\pi^\top)(I - \gamma P_\pi^\top)^{-1} D_\mu i = f_0\text{)} \\
&\leq \|v\|_{f_n}^2 - \|v\|_{f_0}^2 + \left\|(I - \gamma P_\pi^\top)(f - f_n)\right\|_\infty \|v\|^2 \\
&\qquad\qquad \text{(Property of } \ell_2 \text{ norm of a diagonal matrix)} \\
&\leq \|v\|_{f_n}^2 - \|v\|_{f_0}^2 + \left\|(I - \gamma P_\pi^\top)\right\|_\infty \gamma^{n+1} \frac{d_{\mu,max}^2}{d_{\mu,min}} \|m\|_1 \|v\|^2 \quad \text{(Lemma 5.2)} \\
&\leq \|v\|_{f_n}^2 - \|v\|_{f_0}^2 + \kappa \min_s i(s) d_\mu(s) \|v\|^2 \quad \text{(Using (5.5))} \\
&\leq \|v\|_{f_n}^2 - \|v\|_{f_0}^2 + \kappa \|v\|_{f_n}^2 \quad \text{(Using } \min_{s'} i(s') d_\mu(s') \leq f_0(s) < f_n(s) \text{)} \\
&= (1+\kappa)\|v\|_{f_n}^2 - \|v\|_{f_0}^2 \\
&= (1+\kappa)\|v\|_{f_n}^2 - \sum_s v(s)^2 d_\mu(s) i(s) \\
&= (1+\kappa)\|v\|_{f_n}^2 - \sum_s v(s)^2 f(s) \frac{d_\mu(s) i(s)}{f(s)} \\
&\leq (1+\kappa)\|v\|_{f_n}^2 - \kappa \sum_s v(s)^2 f(s) \quad \text{(Definition of } \kappa \text{ and } f(s) > 0\text{)} \\
&\leq (1+\kappa)\|v\|_{f_n}^2 - \kappa \sum_s v(s)^2 f_n(s) \quad \text{(Using } f(s) > f_n(s)\text{)} \\
&= \|v\|_{f_n}^2
\end{aligned}
$$

Consequently,

$$
\|\mathcal{T}_\pi v_1 - \mathcal{T}_\pi v_2\|_{f_n}^2 = \gamma^2 \|P_\pi(v_1 - v_2)\|_{f_n}^2 \leq \gamma \|v_1 - v_2\|_{f_n}^2,
$$

implying that $\mathcal{T}_\pi$ is a $\sqrt{\gamma}$-contraction in $\|\cdot\|_{f_n}$. Since Π_{f_n} is nonexpansive in $\|\cdot\|_{f_n}$, it is easy to see that $\Pi_{f_n}\mathcal{T}_\pi$ is a $\sqrt{\gamma}$ contraction in $\|\cdot\|_{f_n}$ as well.

For the fixed point $w_{*,n}$, we have

$$\begin{aligned} A_n w_{*,n} &= b_n \qquad \text{(B.14)} \\ \iff X^\top D_{f_n}(\gamma P_\pi - I)X w_{*,n} &= X^\top D_{f_n} r_\pi \\ \iff X^\top D_{f_n}(r_\pi + \gamma P_\pi X w_{*,n}) &= X^\top D_{f_n} X w_{*,n} \\ \iff X(X^\top D_{f_n} X)^{-1} X^\top D_{f_n}(r_\pi + \gamma P_\pi X w_{*,n}) &= X(X^\top D_{f_n} X)^{-1} X^\top D_{f_n} X w_{*,n} \\ \iff \Pi_{f_n}\mathcal{T}_\pi(X w_{*,n}) &= X w_{*,n}. \end{aligned}$$

Then,

$$\|X w_{*,n} - v_\pi\|_{f_n}^2 = \|X w_{*,n} - \Pi_{f_n} v_\pi\|_{f_n}^2 + \|\Pi_{f_n} v_\pi - v_\pi\|_{f_n}^2$$

(Pythagorean theorem)

$$\begin{aligned} &= \|\Pi_{f_n}\mathcal{T}_\pi(X w_{*,n}) - \Pi_{f_n}\mathcal{T}_\pi v_\pi\|_{f_n}^2 + \|\Pi_{f_n} v_\pi - v_\pi\|_{f_n}^2 \\ &\le \gamma \|X w_{*,n} - v_\pi\|_{f_n}^2 + \|\Pi_{f_n} v_\pi - v_\pi\|_{f_n}^2. \end{aligned}$$

Rearranging terms completes the proof. □

B.15 Proof of Lemma 6.1

Proof. Obviously, for any $\gamma \in [0,1]$, τ_γ is a solution. For $\gamma < 1$, τ_γ is the unique solution to (6.1) because $I - \gamma P_\pi^\top$ is nonsingular. We now proceed to showing the uniqueness of τ_γ when $\gamma = 1$.

Let τ be a solution to (6.1) and (6.3), i.e., $D_\mu \tau = P_\pi^\top D_\mu \tau$. This means that $D_\mu \tau$ is a left eigenvector of P_π associated with the Perron-Frobenius eigenvalue 1. Note $d_{\pi,\gamma}$ is also a left eigenvector of P_π associated with the eigenvalue 1. According to the Perron-Frobenius theorem for nonnegative irreducible matrices (Horn and Johnson, 2012), the left eigenspace of the Perron-Frobenius eigenvalue is 1-dimensional. Consequently, there exists a scalar α such that $D_\mu \tau = \alpha d_{\pi,\gamma}$. We then have

$$\alpha = \alpha 1^\top d_{\pi,\gamma} = 1^\top D_\mu \tau = d_\mu^\top \tau = 1,$$

implying $D_\mu \tau = d_\gamma$, i.e., $\tau = \tau_\gamma$, which completes the proof. □

B.16 Proof of Theorem 6.3

Proof. Let

$$d_k \doteq \begin{bmatrix} \kappa_k \\ w_k \\ \eta_k \end{bmatrix}.$$

We can then rewrite the GradientDICE updates in Algorithm 6 as

$$d_{k+1} \doteq d_k + \alpha_t(G_{k+1}d_k + g_{k+1})$$

where

$$G_{k+1} \doteq \begin{bmatrix} -x_k^\top x_k & -(x_k - \gamma x_k')x_k^\top & 0 \\ x_k(x_k^\top - \gamma x_k'^\top) & -\xi I & -\lambda x_k \\ 0 & \lambda x_k^\top & -\lambda \end{bmatrix},$$
$$g_{k+1} \doteq \begin{bmatrix} (1-\gamma)x_{0,k} \\ 0 \\ -\lambda \end{bmatrix}.$$

We now proceed via invoking Corollary A.4. It can be computed that

$$G \doteq \mathbb{E}[G_k] = \begin{bmatrix} -C & A^\top & 0 \\ -A & -\xi I & -\lambda X^\top d_\mu \\ 0 & \lambda d_\mu^\top X & -\lambda \end{bmatrix},$$
$$g \doteq \mathbb{E}[g_k] = \begin{bmatrix} (1-\gamma)X^\top d_{p_0\pi} \\ 0 \\ -\lambda \end{bmatrix}.$$

Assumption A.17 immediately holds because we consider finite MDPs. It remains to verify Assumption A.19.

We first show that the real part of any nonzero eigenvalue of G is strictly negative. Let $\zeta \in \mathbb{C}, \zeta \neq 0$ be a nonzero eigenvalue of G with normalized eigenvector x, i.e.,

$$x^H x = 1,$$

where x^H is the complex conjugate of x. Hence $x^H G x = \zeta, x \neq 0$. Let

$$x \doteq \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},$$

where $x_1 \in \mathbb{C}^K, x_2 \in \mathbb{C}^K, x_3 \in \mathbb{C}$. It is easy to verify

$$\zeta = -x_1^H C x_1 - x_2^H A x_1 + x_1^H A^\top x_2 - \xi x_2^H x_2 + \lambda x_3^H d_\mu^\top X x_2 - \lambda x_2^H X^\top d_\mu x_3 - \lambda x_3^H x_3.$$

As A is real, $A^\top = A^H$. Consequently, $(x_2^H A x_1)^H = x_1^H A^\top x_2$, yielding

$$\mathrm{Re}(x_2^H A x_1 - x_1^H A^\top x_2) = 0,$$

where $\mathrm{Re}(\cdot)$ denotes the real part. Similarly, we can show

$$\mathrm{Re}(\lambda x_3^H d_\mu^\top X x_2 - \lambda x_2^H X^\top d_\mu x_3) = 0.$$

Consequently, we have

$$\mathrm{Re}(\zeta) = \mathrm{Re}(x^H G x) = -x_1^H C x_1 - \xi x_2^H x_2 - \lambda x_3^H x_3.$$

According to Assumption 2.3, $x_1^H C x_1 \geq 0$, where the equality holds i.f.f. $x_1 = 0$. When $\gamma < 1$, we have $\xi = 0$. Then $\zeta \neq 0$ implies at least one of $\{x_1, x_3\}$ is nonzero. Consequently, we have $\mathrm{Re}(\zeta) < 0$. When $\gamma = 1$, we have $\xi > 0$. Then $\zeta \neq 0$ implies at least one of $\{x_1, x_2, x_3\}$ is nonzero. Consequently, we have $\mathrm{Re}(\zeta) < 0$.

We then show 0 is not an eigenvalue of G, which completes the proof. It suffices to show $\det(G) \neq 0$. Applying Lemma C.6 to G yields

$$\begin{aligned}
\det(G) =& -\lambda \det\left(\begin{bmatrix} -C & A^\top \\ -A & -\xi I \end{bmatrix} + \lambda^{-1}\begin{bmatrix} 0 & 0 \\ 0 & -\lambda^2 X^\top d_\mu d_\mu^\top X \end{bmatrix}\right) \\
=&(-1)^{2K+1}\lambda \det\left(\begin{bmatrix} C & -A^\top \\ A & \xi I + \lambda X^\top d_\mu d_\mu^\top X \end{bmatrix}\right) \\
=&(-1)^{2K+1}\lambda \det(C) \det\left(\xi I + \lambda X^\top d_\mu d_\mu^\top X + A C^{-1} A^\top\right).
\end{aligned}$$

Note $\lambda X^\top d_\mu d_\mu^\top X$ is always positive semidefinite. Assumption 6.1 ensures at least one of $\{\xi I, A^\top C^{-1} A\}$ is strictly positive definite, which ensures $\det(G) \neq 0$ and completes the proof. □

B.17 Proof of Proposition 6.4

Proof. According to the Perron-Frobenius theorem (cf. the proof of Theorem 6.2), it suffices to show

$$\begin{aligned}
&L_1\left(\lim_{\xi \to 0} w_{\infty,\xi}\right) = L_2\left(\lim_{\xi \to 0} w_{\infty,\xi}\right) = 0, \\
&L_1(w_{\infty,\xi}) \doteq d_\mu^\top X w_{\infty,\xi} - 1, \\
&L_2(w_{\infty,\xi}) \doteq \left\| D_\mu X w_{\infty,\xi} - P_\pi^\top X D_\mu w_{\infty,\xi} \right\|^2_{X C^{-1} X^\top},
\end{aligned}$$

as w_* is the only w satisfying $L_1(w) = L_2(w) = 0$. With the eigendecomposition of $A^\top C^{-1} A$, we can compute Ξ explicitly. Simple algebraic manipulation then yields

$$L_1(w_{\infty,\xi}) = \frac{\lambda u^\top \Lambda_\xi u}{1+\lambda u^\top \Lambda_\xi u} - 1,$$
$$L_2(w_{\infty,\xi}) = \frac{\lambda^2 u^\top \Lambda_\xi u}{(1+\lambda u^\top \Lambda_\xi u)^2} + \frac{\lambda^2 \xi u^\top \Lambda_\xi^2 u}{(1+\lambda u^\top \Lambda_\xi u)^2},$$

where $\Lambda_\xi \doteq diag([\frac{1}{\xi+\lambda_1}, \ldots, \frac{1}{\xi+\lambda_r}, \frac{1}{\xi}, \cdots, \frac{1}{\xi}])$. The desired limits then follow from the L'Hopital's rule. □

B.18 Proof of Theorem 7.1

Proof. The proof is similar to the proof of Theorem 3.3 in Section B.3. We, therefore, highlight only the difference to avoid verbatim repetition. Define

$$\theta \doteq \begin{bmatrix} \theta^r \\ \theta^w \end{bmatrix}, u \doteq \begin{bmatrix} \hat{r} \\ w \end{bmatrix},$$
$$h_\theta(s,a,s') \doteq \begin{bmatrix} r(s,a) \\ x(s,a)r(s,a) \end{bmatrix} + \begin{bmatrix} 0 & \sum_{a'} \pi(a'|s')x(s',a')^\top - x(s,a)^\top \\ -x(s,a) & x(s,a)\sum_{a'} \pi(a'|s')x(s',a')^\top \end{bmatrix} \Gamma_{B_1}(\theta),$$

$$G_\theta(s,a,s') \doteq \begin{bmatrix} 1 & 0 \\ 0 & x(s,a)x(s,a)^\top + \eta I \end{bmatrix}.$$

We can then rewrite the update of $\hat{r}$ and w in Algorithm 7 as

$$u_{t+1} = u_t + \alpha_t(h_{\theta_t}(Y_t) - G_{\theta_t}(Y_t)u_t).$$

Similarly, we define

$$\bar{h}(\theta) \doteq \mathbb{E}_{(s,a,s')\sim d_\mathcal{Y}(\cdot)}[h_\theta(s,a,s')] = \bar{h}_1 + \bar{H}_2 \Gamma_{B_1}(\theta),$$
$$\bar{h}_1 \doteq \begin{bmatrix} d_\mu^\top r \\ X^\top D_\mu r \end{bmatrix}, \bar{H}_2 \doteq \begin{bmatrix} 0 & d_\mu^\top (P_\pi - I)X \\ -X^\top d_\mu & X^\top D_\mu P_\pi X \end{bmatrix}$$
$$\bar{G}(\theta) \doteq \mathbb{E}_{(s,a,s')\sim d_\mathcal{Y}(\cdot)}[G_\theta(s,a,s')] = \begin{bmatrix} 1 & 0 \\ 0 & X^\top D_\mu X + \eta I \end{bmatrix},$$
$$u^*(\theta) \doteq \bar{G}(\theta)^{-1}\bar{h}(\theta).$$

We proceed to verifying Assumptions 3.1, 3.2, and 3.3 to invoke Theorem 3.1.

Assumption 3.1 can be verified with Corollary A.2 in the same way as the proof of Theorem 3.3 in Section B.3.

For Assumption 3.3 to hold, note

$$\begin{aligned}
\left\|\bar{G}(\theta)^{-1}\right\| &= \max\left\{1, \left\|(X^\top D_\mu X + \eta I)^{-1}\right\|\right\} \leq \max\left\{1, \frac{1}{\eta}\right\} \\
\left\|\bar{H}_2\right\|^2 &= \max_{\|u\|=1} \left\|\bar{H}_2 u\right\|^2 = \max_{\|u\|=1} \left\| \begin{bmatrix} d_\mu^\top (P_\pi - I) X w \\ -X^\top d_\mu \hat{r} + X^\top D_\mu P_\pi X w \end{bmatrix} \right\|^2 \\
&= \max_{\|u\|=1} \left\|d_\mu^\top (P_\pi - I) X w\right\|^2 + \left\|-X^\top d_\mu \hat{r} + X^\top D_\mu P_\pi X w\right\|^2 \\
&\leq \max_{\|u\|=1} \left\|d_\mu^\top (P_\pi - I) X\right\|^2 \|w\|^2 + 2\left\|X^\top d_\mu\right\|^2 \|\hat{r}\|^2 + 2\left\|X^\top D_\mu P_\pi X\right\|^2 \|w\|^2 \\
&\leq \max\left\{\left\|d_\mu^\top (P_\pi - I) X\right\|^2 + 2\left\|X^\top D_\mu P_\pi X\right\|^2, 2\left\|X^\top d_\mu\right\|^2\right\} \\
\implies \left\|\bar{H}_2\right\| &\leq \|X\| \max\left\{\left\|d_\mu^\top (P_\pi - I)\right\| + \sqrt{2}\|D_\mu P_\pi\|, \sqrt{2}\|d_\mu\|\right\}.
\end{aligned}$$

The above bounds together with Lemmas C.1 and C.2 suggests that

$$\begin{aligned}
&\|u^*(\theta_1) - u^*(\theta_2)\| \\
\leq &\max\left\{1, \frac{1}{\eta}\right\} \|X\| \max\left\{\left\|d_\mu^\top (P_\pi - I)\right\| + \sqrt{2}\|D_\mu P_\pi\|, \sqrt{2}\|d_\mu\|\right\} \|\theta_1 - \theta_2\|.
\end{aligned}$$

Take any $\xi \in (0, 1)$, assume

$$\|X\| \leq \frac{1 - \xi}{\max\left\{1, \frac{1}{\eta}\right\} \max\left\{\left\|d_\mu^\top (P_\pi - I)\right\| + \sqrt{2}\|D_\mu P_\pi\|, \sqrt{2}\|d_\mu\|\right\}}, \tag{B.15}$$

we then have

$$\|u^*(\theta_1) - u^*(\theta_2)\| \leq (1 - \xi)\|\theta_1 - \theta_2\|.$$

Assumption 3.3, therefore, holds.

We now select proper R_{B_1} and R_{B_2} to fulfill Assumption 3.2. Using (B.15), it is easy to see

$$\|u^*(\theta)\| \leq \max\left\{1, \frac{1}{\eta}\right\} \left\|\bar{h}_1\right\| + (1 - \xi) R_{B_1}.$$

For sufficiently large R_{B_1}, e.g.,

$$R_{B_1} \geq \max\left\{1, \frac{1}{\eta}\right\} \frac{\left\|\bar{h}_1\right\|}{\xi} + 1, \tag{B.16}$$

we have $\sup_\theta \|u^*(\theta)\| \leq R_{B_1} - \xi$. Selecting $R_{B_2} \in (R_{B_1} - \xi, R_{B_1})$ then fulfills Assumption 3.2.

Invoking Theorem 3.1 then implies that there exists a unique θ_∞ such that

$$u^*(\theta_\infty) = \theta_\infty \quad \text{and} \quad \lim_{t \to \infty} \theta_t = \lim_{t \to \infty} u_t = \theta_\infty.$$

Next we show what θ_∞ is. We define

$$f(\theta) \doteq \bar{G}(\theta)^{-1}(\bar{h}_1 + \bar{H}_2\theta).$$

Note this is just $u^*(\theta)$ without the projection. Under (B.15), it is easy to show f is a contraction. The Banach fixed-point theorem then asserts that f adopts a unique fixed point, which we refer to as u^*_η. Using (B.15) again, we get

$$\begin{aligned} \left\|u^*_\eta\right\| = \left\|f(u^*_\eta)\right\| &\le \max\left\{1, \frac{1}{\eta}\right\}\left\|\bar{h}_1\right\| + (1-\xi)\left\|u^*_\eta\right\| \\ \implies \left\|u^*_\eta\right\| &\le \max\left\{1, \frac{1}{\eta}\right\}\frac{\left\|\bar{h}_1\right\|}{\xi}. \end{aligned}$$

Then for sufficiently large R_{B_1}, e.g.,

$$R_{B_1} \ge \max\left\{1, \frac{1}{\eta}\right\}\frac{\left\|\bar{h}_1\right\|}{\xi}, \tag{B.17}$$

we have $u^*_\eta = \Gamma_{B_1}(u^*_\eta)$, implying u^*_η is a fixed point of $u^*(\cdot)$ as well. As $u^*(\cdot)$ is a contraction, we have $\theta_\infty = u^*_\eta$. Writing u^*_η as $\begin{bmatrix}\hat{r}^*_\eta \\ w^*_\eta\end{bmatrix}$ and expanding $f(u^*_\eta) = u^*_\eta$ yields

$$\begin{aligned} \hat{r}^*_\eta &= d_\mu^\top(r + P_\pi X w^*_\eta - X w^*_\eta), \\ (X^\top D_\mu X + \eta I) w^*_\eta &= X^\top D_\mu r - X^\top d_\mu \hat{r}^*_\eta + X^\top D_\mu P_\pi X w^*_\eta. \end{aligned}$$

Rearranging terms yields $(\bar{A} - \eta I)w^*_\eta + \bar{b} = 0$, i.e., w^*_η is the unique (due to the contraction of f) solution of $(\bar{A} - \eta I)w + \bar{b} = 0$.

We now bound $\left\|Xw^*_\eta - \bar{q}^c_\pi\right\|$. Recall the regularized projection defined in (B.9). We have

$$\begin{aligned} &\left\|Xw^*_\eta - \bar{q}^c_\pi\right\| \\ \le &\left\|Xw^*_\eta - \Pi^\eta_{d_\mu}\bar{q}^c_\pi\right\| + \left\|\Pi^\eta_{d_\mu}\bar{q}^c_\pi - \bar{q}^c_\pi\right\| \\ = &\left\|X(X^\top D_\mu X + \eta I)^{-1}(X^\top D_\mu r - X^\top d_\mu \hat{r}^*_\eta + X^\top D_\mu P_\pi X w^*_\eta) - \Pi^\eta_{d_\mu}\bar{q}^c_\pi\right\| \\ &+ \left\|\Pi^\eta_{d_\mu}\bar{q}^c_\pi - \bar{q}^c_\pi\right\| \\ = &\left\|\Pi^\eta_{d_\mu}(r + P_\pi X w^*_\eta) - \Pi^\eta_{d_\mu}(r + P_\pi \bar{q}^c_\pi)\right\| + \left\|\Pi^\eta_{d_\mu}\bar{q}^c_\pi - \bar{q}^c_\pi\right\| \quad (\text{Using } X^\top d_\mu = 0) \\ = &\left\|\Pi^\eta_{d_\mu}P_\pi(Xw^*_\eta - \bar{q}^c_\pi)\right\| + \left\|\Pi^\eta_{d_\mu}\bar{q}^c_\pi - \bar{q}^c_\pi\right\| \\ \le &\left\|X(X^\top D_\mu X + \eta I)^{-1}X^\top D_\mu P_\pi\right\|\left\|Xw^*_\eta - \bar{q}^c_\pi\right\| + \left\|\Pi^\eta_{d_\mu}\bar{q}^c_\pi - \bar{q}^c_\pi\right\| \\ \le &\frac{1}{\eta}\|X\|^2\|D_\mu P_\pi\|\left\|Xw^*_\eta - \bar{q}^c_\pi\right\| + \left\|\Pi^\eta_{d_\mu}\bar{q}^c_\pi - \bar{q}^c_\pi\right\|. \end{aligned}$$

Assuming

$$\|X\|^2 \leq \frac{(1-\xi)\eta}{\|D_\mu P_\pi\|}, \tag{B.18}$$

we have

$$\begin{aligned}\left\|Xw_\eta^* - \bar{q}_\pi^c\right\| \leq & \frac{1}{\xi}\left\|\Pi_{d_\mu}^\eta \bar{q}_\pi^c - \bar{q}_\pi^c\right\| \\ \leq & \frac{1}{\xi}\left(\frac{\sigma_{\max}(X)^2}{\sigma_{\min}(X)^4\sigma_{\min}(D_\mu)^{2.5}}\|\bar{q}_\pi^c\|\eta + \left\|\Pi_{d_\mu}\bar{q}_\pi^c - \bar{q}_\pi^c\right\|\right) \quad \text{(cf. (B.10))} \quad .\end{aligned}$$

It is then easy to see

$$\begin{aligned}& |\hat{r}_\eta^* - \hat{r}_\pi| \\ \leq & \left\|d_\mu^\top (P_\pi - I)(Xw_\eta^* - \bar{q}_\pi^c)\right\| \\ \leq & \frac{1}{\xi}\left\|d_\mu^\top (P_\pi - I)\right\|\left(\frac{\sigma_{\max}(X)^2}{\sigma_{\min}(X)^4\sigma_{\min}(D_\mu)^{2.5}}\|\bar{q}_\pi^c\|\eta + \left\|\Pi_{d_\mu}\bar{q}_\pi^c - \bar{q}_\pi^c\right\|\right).\end{aligned}$$

Taking infimum for $c \in \mathbb{R}$ then yields the desired results.

Combining (B.15), (B.16), (B.17), and (B.18), the desired constants are

$$C_0 \doteq \min\left\{\frac{1-\xi}{\max\left\{1, \frac{1}{\eta}\right\}\max\left\{\left\|d_\mu^\top (P_\pi - I)\right\| + \sqrt{2}\|D_\mu P_\pi\|, \sqrt{2}\|d_\mu\|\right\}}, \sqrt{\frac{(1-\xi)\eta}{\|D_\mu P_\pi\|}}\right\},$$

$$C_1 \doteq \max\left\{1, \frac{1}{\eta}\right\}\frac{\left\|\bar{h}_1\right\|}{\xi} + 1,$$

which completes the proof. □

B.19 Proof of Theorem 8.1

Proof. With

$$\kappa_k \doteq \begin{bmatrix}\nu_k \\ w_k\end{bmatrix},$$

we rewrite the updates to $\{w_k\}, \{\nu_k\}$ in Algorithm 8 as

$$\kappa_{k+1} = \kappa_k + \alpha_k(G_k\kappa_k + h_k),$$

where

$$\begin{aligned}G_k &\doteq \begin{bmatrix}-x_{k,1}x_{k,1}^\top & x_{k,1}(x_{k,1}'^\top - x_{k,1}^\top) - x_{k,1}(x_{k,2}'^\top - x_{k,2}^\top) \\ -(x_{k,1} - x_{k,1}')x_{k,1}^\top + (x_{k,2} - x_{k,2}')x_{k,1}^\top & -\eta I\end{bmatrix}, \\ h_k &\doteq \begin{bmatrix}r_{k,1}x_{k,1} - r_{k,2}x_{k,1} \\ 0\end{bmatrix}.\end{aligned}$$

The asymptotic behavior of $\{\kappa_k\}$ is governed by

$$\bar{G} \doteq \mathbb{E}[G_k] = \begin{bmatrix} -C & \bar{A} \\ -\bar{A}^\top & -\eta I \end{bmatrix}$$
$$\bar{h} \doteq \mathbb{E}[h_k] = \begin{bmatrix} \bar{b} \\ 0 \end{bmatrix}.$$

We now proceed by invoking Corollary A.4. Assumption A.14 is satisfied by our requirement on $\{\alpha_k\}$. Assumption A.16 holds immediately because in our setting we have $\epsilon_k \equiv 0$. Assumption A.18 holds immediately because we consider a finite MDP. To verify Assumption A.19, we first show $\det(\bar{G}) \neq 0$. Using the rule of block matrix determinant, we have

$$\det(\bar{G}) = \det(C)\det(\eta I + \bar{A}^\top C^{-1}\bar{A}).$$

Assumption 2.3 ensures that C is positive definite and $\bar{A}^\top C^{-1}\bar{A}$ is positive semidefinite, implying $\eta I + \bar{A}^\top C^{-1}\bar{A}$ is positive definite. Consequently $\det(\bar{G}) \neq 0$. Let $\lambda \in \mathbb{C}$ be an eigenvalue of $\bar{G}$. $\det(\bar{G}) \neq 0$ implies $\lambda \neq 0$. Let $z \neq 0 \in \mathbb{C}^K$ be the corresponding normalized eigenvector of λ, i.e., $z^H z = 1$, where z^H is the conjugate transpose of z. Let $z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$, we have

$$\lambda = z^H \bar{G} z = -z_1^H C z_1 - z_2^H \bar{A}^\top z_1 + z_1^H \bar{A} z_2 - \eta z_2^H I z_2.$$

As $(z_2^H \bar{A}^\top z_1)^H = z_1^H \bar{A} z_2$, we have $\mathrm{Re}(-z_2^H \bar{A}^\top z_1 + z_1^H \bar{A} z_2) = 0$, where $\mathrm{Re}(\cdot)$ denotes the real part. So

$$\mathrm{Re}(\lambda) = -z_1^H C z_1 - \eta z_2^H I z_2 \leq 0.$$

Because $\lambda \neq 0$, we have $\mathrm{Re}(\lambda) < 0$. Assumption A.19 then holds. Invoking Corollary A.4 yields

$$\lim_{k\to\infty} \kappa_k = -\bar{G}^{-1}\bar{h} \quad \text{a.s.} \quad .$$

It can be easily seen when $\bar{A}$ is invertible, even if $\eta = 0$, $\det(\bar{G}) \neq 0$ still holds. Consequently, the convergence of $\{\kappa_k\}$ remains. Let w_η^* be the lower half of $-\bar{G}^{-1}\bar{h}$, we have by matrix multiplication that

$$w_\eta^* \doteq -(\eta I + \bar{A}^\top C^{-1}\bar{A})^{-1}\bar{A}^\top C^{-1}\bar{b}.$$

From (8.2), we can rewrite $L(w)$ as

$$L(w) = \|\bar{A}w + \bar{b}\|_{C^{-1}}^2 + \eta\|w\|^2.$$

It is easy to verify (e.g., using the first order optimality condition of $L(w)$) that w_η^* is the unique minimizer of $L(w)$.

We can also rewrite the update to $\{\hat{r}_k\}$ in Algorithm 8 as

$$\hat{r}_{k+1} \doteq \hat{r}_k + \beta_k\Big(\frac{1}{2}\sum_{i=1}^{2}(r_{k,i} + x_{k,i}'^\top w_\eta^* - x_{k,i}^\top w_\eta^*) - \hat{r}_k + o(1)\Big).$$

Similar to the convergence proof of $\{\kappa_k\}$, we can obtain

$$\lim_{k\to\infty} \hat{r}_k = d_\mu^\top(r + P_\pi X w_\eta^* - X w_\eta^*).$$

Assumption 8.1 implies there exists w such that,

$$\bar{A}w + \bar{b} = 0,$$

or equivalently,

$$C^{-\frac{1}{2}}\bar{A}w + C^{-\frac{1}{2}}\bar{b} = 0$$

has unique or infinite manly solutions. From standard results of system of linear equations, this is equivalent to

$$C^{-\frac{1}{2}}\bar{A}(C^{-\frac{1}{2}}\bar{A})^\dagger C^{-\frac{1}{2}}\bar{b} = C^{-\frac{1}{2}}\bar{b},$$

where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse, which always exists for any matrix. By the property of the Moore-Penrose pseudoinverse, it is easy to see

$$w_0^* \doteq \lim_{\eta\to 0} w_\eta^* = -(C^{-\frac{1}{2}}\bar{A})^\dagger C^{-\frac{1}{2}}\bar{b}.$$

Consequently, we have

$$C^{-\frac{1}{2}}(\bar{A}w_0^* + \bar{b}) = -C^{-\frac{1}{2}}\bar{A}(C^{-\frac{1}{2}}\bar{A})^\dagger C^{-\frac{1}{2}}\bar{b} + C^{-\frac{1}{2}}\bar{b} = 0,$$

implying

$$\bar{A}w_0^* + \bar{b} = 0.$$

Applying SVD to $C^{-\frac{1}{2}}\bar{A}$ and using σ to denote its minimum nonzero singular value, it is easy to see

$$\|w_\eta^* - w_0^*\| \leq \frac{\eta}{\sigma^3}\left\|C^{-\frac{1}{2}}\bar{b}\right\|, \tag{B.19}$$

which completes the proof. □

B.20 Proof of Proposition 8.2

Proof. Plugging $w^*, \hat{r}^*$ into (7.2) and (7.3) yields

$$\begin{aligned}\hat{r}^* - d_\mu^\top (r + P_\pi X w^* - X w^*) = 0,\\ X^\top D (r - \hat{r}^* 1 + P_\pi X w^* - X w^*) = 0.\end{aligned}$$

So we have

$$\left\| X^\top D (r - \hat{r}^* 1 + P_\pi X w^* - X w^*) \right\|_{C^{-1}}^2 = 0,$$

implying

$$\left\| \Pi_{d_\mu} (r - \hat{r}^* 1 + P_\pi X w^* - X w^*) \right\|_{d_\mu}^2 = 0.$$

Using the Schur complement, Assumption 8.2 implies (see Kolter (2011) for more details)

$$\left\| \Pi_{d_\mu} P_\pi X w \right\|_{d_\mu} \leq \xi \| X w \|_{d_\mu}$$

holds for any $w \in \mathbb{R}^K$. We then have

$$\begin{aligned}
&\| X w^* - q_\pi^c \|_{d_\mu} \\
\leq &\left\| X w^* - \Pi_{d_\mu} q_\pi^c \right\|_{d_\mu} + \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu} \\
= &\left\| \Pi_{d_\mu} (r + P_\pi X w^* - \hat{r}^* 1) - \Pi_{d_\mu} (r + P_\pi q_\pi^c - r_\pi 1) \right\|_{d_\mu} + \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu} \\
\leq &\left\| \Pi_{d_\mu} P_\pi X w^* - \Pi_{d_\mu} P_\pi q_\pi^c \right\|_{d_\mu} + \left\| \Pi_{d_\mu} (\hat{r}^* 1 - r_\pi 1) \right\|_{d_\mu} + \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu} \\
= &\left\| \Pi_{d_\mu} P_\pi X w^* - \Pi_{d_\mu} P_\pi q_\pi^c \right\|_{d_\mu} + \left\| X (X^\top D_\mu X)^{-1} (X^\top D_\mu 1) (\hat{r}^* - r_\pi) \right\|_{d_\mu} \\
&+ \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu} \\
= &\left\| \Pi_{d_\mu} P_\pi X w^* - \Pi_{d_\mu} P_\pi q_\pi^c \right\|_{d_\mu} + \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu} \quad (\text{Using } X^\top d_\mu = 0) \\
\leq &\left\| \Pi_{d_\mu} P_\pi X w^* - \Pi_{d_\mu} P_\pi \Pi_{d_\mu} q_\pi^c \right\|_{d_\mu} + \left\| \Pi_{d_\mu} P_\pi \Pi_{d_\mu} q_\pi^c - \Pi_{d_\mu} P_\pi q_\pi^c \right\|_{d_\mu} + \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu} \\
\leq &\xi \left\| X w^* - \Pi_{d_\mu} q_\pi^c \right\|_{d_\mu} + \| P_\pi \|_{d_\mu} \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu} + \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu} \\
= &\xi \| X w^* - q_\pi^c \|_{d_\mu} + (\| P_\pi \|_{d_\mu} + 1) \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu}.
\end{aligned}$$

From the above derivation we have

$$\| X w^* - q_\pi^c \|_{d_\mu} \leq \frac{\| P_\pi \|_{d_\mu} + 1}{1 - \xi} \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu}.$$

Taking the infimum then yields the desired bound:

$$\inf_{c \in \mathbb{R}} \| X w^* - q_\pi^c \|_{d_\mu} \leq \inf_{c \in \mathbb{R}} \frac{\| P_\pi \|_{d_\mu} + 1}{1 - \xi} \left\| \Pi_{d_\mu} q_\pi^c - q_\pi^c \right\|_{d_\mu}.$$

For the average reward, we have, for all $c \in \mathbb{R}$,

$$\begin{aligned}
|r_\pi - \hat{r}^*| &= \left|d_\mu^\top (P_\pi - I)(Xw^* - q_\pi^c)\right| \\
&= \left|d_\mu^\top (P_\pi - I) D_\mu^{-\frac{1}{2}} D_\mu^{\frac{1}{2}} (Xw^* - q_\pi^c)\right| \\
&\leq \left\|d_\mu^\top (P_\pi - I)\right\|_{D_\mu^{-1}} \|Xw^* - q_\pi^c\|_{d_\mu},
\end{aligned}$$

where the inequality is due to the Cauchy-Schwarz inequality. This completes the proof.

□

B.21 Proof of Theorem 9.1

Proof. The proof is similar to the proof of Theorem 3.3 in Section B.3 but is more involving. Consider the stochastic process $Y_t \doteq (S_t, A_t, S_{t+1})$ involving in the space

$$\mathcal{Y} \doteq \{(s, a, s') \mid s \in \mathcal{S}, a \in \mathcal{A}, s' \in \mathcal{S}, p(s'|s, a) > 0\}.$$

We define

$$\begin{aligned}
h_\theta(s, a, s') &\doteq \big(r(s, a) + \gamma \max_{a'} x(s', a')^\top \bar{\theta}\big) x(s, a), \\
G_\theta(s, a, s') &\doteq x(s, a) x(s, a)^\top + \eta I,
\end{aligned}$$

where $\bar{\theta} \doteq \Gamma_{B_1}(\theta)$ is shorthand. As $\theta_t \in B_1$ holds for all t, we can rewrite the update of w_t in Algorithm 9 as

$$w_{t+1} = w_t + \alpha_t (h_{\theta_t}(Y_t) - G_{\theta_t}(Y_t) w_t).$$

The expected update given θ is then controlled by

$$\begin{aligned}
\bar{h}(\theta) &\doteq \mathbb{E}_{(s,a)\sim d_{\mu_\theta}(\cdot), s'\sim p(\cdot|s,a)}[h_\theta(s, a, s')] \\
&= X^\top D_{\mu_\theta} r + \gamma X^\top D_{\mu_\theta} P_{\pi_{\bar{\theta}}} X \bar{\theta}, \\
\bar{G}(\theta) &\doteq \mathbb{E}_{(s,a)\sim d_{\mu_\theta}(\cdot), s'\sim p(\cdot|s,a)}[G_\theta(s, a, s')] \\
&= X^\top D_{\mu_\theta} X + \eta I,
\end{aligned}$$

where Assumption 9.1 ensures the existence of d_{μ_θ} and π_θ is the target policy, i.e. a greedy policy with random tie breaking defined as follows. Let $\mathcal{A}_{s,\theta}^{\max} \doteq \arg\max_a x(s, a)^\top \theta$ be the set of maximizing actions for state s, we define

$$\pi_\theta(a|s) \doteq \begin{cases} \frac{1}{|\mathcal{A}_{s,\theta}^{\max}|}, & a \in \mathcal{A}_{s,\theta}^{\max} \\ 0, & \text{otherwise} \end{cases}.$$

Similar to the proof in Section B.3, we define

$$w^*(\theta) \doteq \bar{G}(\theta)^{-1}\bar{h}(\theta) = (X^\top D_{\mu_\theta} X + \eta I)^{-1} X^\top D_{\mu_\theta}(r + \gamma P_{\pi_{\bar{\theta}}} X\bar{\theta}) \tag{B.20}$$

and proceed to verifying Assumptions 3.1 - 3.3 to invoke Theorem 3.1.

For Assumption 3.1, we resort to Corollary A.2. Assumption A.1 follows immediately from the update rule of Algorithm 9. Assumption A.2 is identical to Assumption 2.4. Assumption A.3 holds thanks to Lemma 3.2 and Assumption 2.6. Thanks to Assumption 2.3, we have $\forall w, \theta$,

$$w^\top \bar{G}(\theta) w \geq w^\top X^\top D_{\mu_\theta} X w + \eta \|w\|^2 \geq \eta \|w\|^2.$$

Assumption A.9 is then verified. Assumption A.10 follows immediately from Assumption 2.8. Assumption A.11 holds thanks to the projection in the definition of $\bar{h}$. Assumption A.12 holds thanks to Assumption 9.2. Invoking Corollary A.2 then verifies Assumption 3.1.

For Assumption 3.3, Lemma C.7 shows that $w^*(\theta)$ is Lipschitz continuous in θ with

$$\begin{aligned} C_w \doteq & \eta^{-1}\|X\|\|r\|L_D + \eta^{-2}\|X\|^3\|r\|L_D + \gamma\eta^{-1}L_0\|X\|^2 \\ & + \gamma U_P \|X\| R_{B_1}(\eta^{-1}\|X\|L_D + \eta^{-2}\|X\|^3 L_D) \end{aligned}$$

being a Lipschitz constant. Here L_D, L_0, and U_P are positive constants detailed in the proof of Lemma C.7. Assuming

$$\|X\| \leq 1 \quad \text{and} \quad \gamma U_P \|X\| R_{B_1} \leq 1, \tag{B.21}$$

we have

$$C_w \leq \eta^{-2}\|X\|(\eta\|r\|L_D + \|r\|L_D + \gamma\eta L_0 + \eta L_D + L_D).$$

Take any $\xi \in (0, 1)$, assuming

$$\|X\| \leq (1-\xi)\eta^2(\eta\|r\|L_D + \|r\|L_D + \gamma\eta L_0 + \eta L_D + L_D)^{-1}, \tag{B.22}$$

it then follows that $C_w \leq 1 - \xi$. Assumptions 3.3, therefore, holds.

We now select proper R_{B_1} and R_{B_2} to fulfill Assumption 3.2. Similar to Lemma C.7 (see, e.g., the last three rows of Table C.1 in the proof of Lemma C.7), we can easily get

$$\|w^*(\theta)\| \leq \eta^{-1}\|X\|\|r\| + \gamma\eta^{-1}\|X\|U_P\|X\|R_{B_1}.$$

Using (B.21) yields

$$\|w^*(\theta)\| \leq \eta^{-1}\|X\|(\|r\|+1).$$

For sufficiently large R_{B_1}, e.g.,

$$R_{B_1} > \eta^{-1}\|X\|(\|r\|+1)+\xi, \tag{B.23}$$

we have $\sup_\theta \|w^*(\theta)\| < R_{B_1}-\xi$. Taking $R_{B_2} \in (R_{B_1}-\xi, R_{B_1})$ then fulfills Assumption 3.2.

Invoking Theorem 3.1, we then get that there exists a unique θ_∞ such that

$$w^*(\theta_\infty) = \theta_\infty \quad \text{and} \quad \lim_{t\to\infty}\theta_t = \lim_{t\to\infty} w_t = \theta_\infty.$$

We now show what θ_∞ is. We define

$$f(\theta) \doteq (X^\top D_{\mu_\theta} X + \eta I)^{-1} X^\top D_{\mu_\theta}(r + \gamma P_{\pi_\theta} X\theta)$$

and consider a ball $B_0 \doteq \left\{\theta \in \mathbb{R}^K \mid \|\theta\| \leq R_{B_0}\right\}$ with R_{B_0} to be tuned (for the Brouwer fixed-point theorem). We have

$$\|f(\theta)\| \leq \eta^{-1}\|X\|\|r\| + \gamma\eta^{-1}\|X\|^2 U_P\|\theta\|$$

Assuming

$$\gamma\eta^{-1}\|X\|^2 U_P < 1-\xi, \tag{B.24}$$

we have

$$\begin{aligned}\|f(\theta)\| &\leq \eta^{-1}\|X\|\|r\| + (1-\xi)\|\theta\| \\ &= R_{B_0} - (R_{B_0} - (1-\xi)\|\theta\| - \eta^{-1}\|X\|\|r\|).\end{aligned}$$

Then for sufficiently large R_{B_0}, e.g.,

$$R_{B_0} \geq \frac{\|X\|\|r\|}{\eta\xi},$$

we have

$$\theta \in B_0 \implies f(\theta) \in B_0.$$

The Brouwer fixed-point theorem then asserts that there exists a $w_\eta^* \in B_0$ such that $f(w_\eta^*) = w_\eta^*$. For sufficiently large R_{B_1}, e.g.,

$$R_{B_1} > R_{B_0}, \tag{B.25}$$

we have $\Gamma_{B_1}(w_\eta^*) = w_\eta^*$, i.e., w_η^* is also a fixed point of $w^*(\cdot)$. The contraction of $w^*(\cdot)$ then implies $\theta_\infty = w_\eta^*$. Rewriting $f(w_\eta^*) = w_\eta^*$ yields

$$A_{\pi_{w_\eta^*},\mu_{w_\eta^*}} w_\eta^* - \eta w_\eta^* + b_{\mu_{w_\eta^*}} = 0.$$

In other words, w_η^* is the unique solution of $(A_{\pi_w,\mu_w} - \eta I)w + b_{\mu_w} = 0$ inside B_1 (due to the contraction of $w^*(\cdot)$). Combining (B.21) (B.22) (B.24) (B.23) (B.25), the desired constant is

$$C_0 \doteq \min\{1, \frac{1}{\gamma U_P R_{B_1}}, \frac{\eta(R_{B_1} - \xi)}{\|r\| + 1}, \sqrt{\frac{\eta(1-\xi)}{\gamma U_P}}, \frac{R_{B_1}\eta\xi}{\|r\|}, \\ \frac{(1-\xi)\eta^2}{\eta\|r\|L_D + \|r\|L_D + \gamma\eta L_0 + \eta L_D + L_D}\},$$

which completes the proof. As R_{B_1} is usually large, in general C_0 is $\mathcal{O}(R_{B_1}^{-1})$. Though C_0 is potentially small, we can use small η as well. So a small C_0 (i.e., $\|X\|$) does not necessarily implies a large regularization bias. □

B.22 Proof of Theorem 9.2

Proof. Let

$$\kappa_t \doteq \begin{bmatrix} u_t \\ w_t \end{bmatrix}.$$

Algorithms 10 implies that

$$\kappa_{t+1} = \kappa_t + \alpha_t(h_{\theta_t}(Y_t) - G_{\theta_t}(Y_t)\kappa_t),$$

where

$$G_\theta(s,a,s') \\ \doteq \begin{bmatrix} x(s,a)x(s,a)^\top & -x(s,a)(\gamma \sum_{a'} \pi_\theta(a'|s')x(s,a) - x(s,a))^\top \\ (\gamma \sum_{a'} \pi_\theta(a'|s')x(s,a) - x(s,a))x(s,a)^\top & \eta I \end{bmatrix},$$

$$h_\theta(s,a,s') \doteq \begin{bmatrix} x(s,a)r(s,a) \\ 0 \end{bmatrix}.$$

We define

$$\begin{aligned} \bar{G}(\theta) &\doteq \mathbb{E}_{(s,a)\sim d_{\mu_\theta}(\cdot), s'\sim p(\cdot|s,a)}[G_\theta(s,a,s')] = \begin{bmatrix} C_{\mu_\theta} & -A_{\pi_\theta,\mu_\theta} \\ A_{\pi_\theta,\mu_\theta}^\top & \eta I \end{bmatrix}, \\ \bar{h}(\theta) &\doteq \mathbb{E}_{(s,a)\sim d_{\mu_\theta}(\cdot), s'\sim p(\cdot|s,a)}[h_\theta(s,a,s')] = \begin{bmatrix} X^\top D_{\mu_\theta} r \\ 0 \end{bmatrix}, \\ w^*(\theta) &\doteq [\bar{G}(\theta)^{-1}\bar{h}(\theta)]_{K+1:2K} \\ &= -(\eta I + A_{\pi_\theta,\mu_\theta}^\top C_{\mu_\theta}^{-1} A_{\pi_\theta,\mu_\theta})^{-1} A_{\pi_\theta,\mu_\theta}^\top C_{\mu_\theta}^{-1} X^\top D_{\mu_\theta} r, \end{aligned} \tag{B.26}$$

where $[\cdot]_{K+1:2K}$ is the subvector indexed from $K+1$ to $2K$.

We proceed to verifying Assumptions 3.1, 3.2, and 3.3 thus invoke Theorem 3.1.

Assumption 3.1 can be verified via Corollary A.2 similarly to the proof in Section B.21. In particular, for Assumption A.9, consider

$$\kappa \doteq \begin{bmatrix} u \\ w \end{bmatrix},$$

we have

$$\begin{aligned} \kappa^\top \bar{G}(\theta)\kappa =& u^\top C_{\mu_\theta} u + \eta \|w\|^2 \\ \geq& \lambda_{min}(C_{\mu_\theta})\|u\|^2 + \eta\|w\|^2 \\ \geq& \min\left\{\inf_\theta \lambda_{min}(C_{\mu_\theta}), \eta\right\}\|\kappa\|^2. \end{aligned}$$

Here $\lambda_{min}(\cdot)$ denotes the minimum eigenvalue. Assumption 9.1, together with the extreme value theorem, the continuity of eigenvalues, and Lemma C.3, ensures that $\inf_\theta \lambda_{min}(C_{\mu_\theta})$ is always strictly positive.

Lemma C.8 shows that if $\|X\| \leq 1$, then there exist a constant $L_w > 0$, which depends on X through only $\frac{X}{\|X\|}$, such that

$$\|w^*(\theta_1) - w^*(\theta_2)\| \leq L_w \|X\| \|\theta_1 - \theta_2\|.$$

As $w^*(\cdot)$ is independent of R_{B_1}, so does L_w. So as long as

$$\|X\| \leq C_0 \doteq \min\left\{1, \frac{1-\xi}{L_w}\right\}, \tag{B.27}$$

$w^*(\cdot)$ is contractive and Assumption 3.3 is satisfied. Since L_w depends on X only through $\frac{X}{\|X\|}$, there are indeed X satisfying (B.27). For example, if some X' does not satisfy (B.27), we can simply scale X' down by some scalar. In the proof of Lemma C.8, we show $\sup_\theta \left\|C_{\mu_\theta}^{-1}\right\| < \infty$. It is then easy to see $\sup_\theta \|w^*(\theta)\| < \infty$. Consequently, we can choose sufficiently large R_{B_1} and R_{B_2} such that Assumption 3.2 holds.

Invoking Theorem 3.1 then yields that there exists a unique w_η^* such that

$$w^*(w_\eta^*) = w_\eta^* \quad \text{and} \quad \lim_{t\to\infty} \theta_t = \lim_{t\to\infty} w_t = w_\eta^*.$$

Expanding $w^*(w_\eta^*) = w_\eta^*$ yields

$$w_\eta^* = -(A_{\pi_{w_\eta^*},\mu_{w_\eta^*}}^\top C_{\mu_{w_\eta^*}}^{-1} A_{\pi_{w_\eta^*},\mu_{w_\eta^*}} + \eta I)^{-1} A_{\pi_{w_\eta^*},\mu_{w_\eta^*}}^\top C_{\mu_{w_\eta^*}}^{-1} b_{\mu_{w_\eta^*}},$$

which completes the proof. □

B.23 Proof of Theorem 9.3

Proof. The proof is combination of the proofs of Theorem 7.1 and Theorem 9.1. To avoid verbatim repetition, in this proof, we show only the existence of the constants C_0 and C_1 without showing the exact expressions. We define

$$\begin{aligned}
\theta \doteq \begin{bmatrix}\theta^r \\ \theta^w\end{bmatrix}, u \doteq \begin{bmatrix}\hat{r} \\ w\end{bmatrix}, \begin{bmatrix}\bar{\theta}^r \\ \bar{\theta}^w\end{bmatrix} &\doteq \Gamma_{B_1}\left(\begin{bmatrix}\theta^r \\ \theta^w\end{bmatrix}\right), \\
h_\theta(s,a,s') &\doteq \begin{bmatrix} r(s,a) \\ x(s,a)r(s,a)\end{bmatrix} \\
&\quad + \begin{bmatrix} 0 & \sum_{a'} \pi_{\bar{\theta}^w}(a'|s')x(s',a')^\top \bar{\theta}^w - x(s,a)^\top \bar{\theta}^w \\ -x(s,a)\bar{\theta}^r & x(s,a)\sum_{a'} \pi_{\bar{\theta}^w}(a'|s')x(s',a')^\top \bar{\theta}^w \end{bmatrix}, \\
G_\theta(s,a,s') &\doteq \begin{bmatrix} 1 & 0^\top \\ 0 & x(s,a)x(s,a)^\top + \eta I\end{bmatrix}.
\end{aligned}$$

We can then rewrite the update of $\hat{r}$ and w in Algorithm 11 as

$$u_{t+1} = u_t + \alpha_t(h_{\theta_t}(Y_t) - G_{\theta_t}(Y_t)u_t).$$

In the rest of this proof, we write μ_θ and π_θ as shorthand for μ_{θ^w} and π_{θ^w}. We define

$$\begin{aligned}
\bar{h}(\theta) &\doteq \mathbb{E}_{(s,a)\sim d_{\mu_\theta}(\cdot), s'\sim p(\cdot|s,a)}[h_\theta(s,a,s')] = \bar{h}_1(\theta) + \bar{H}_2(\theta), \\
\bar{h}_1(\theta) &\doteq \begin{bmatrix} d_{\mu_\theta}^\top r \\ X^\top D_{\mu_\theta} r\end{bmatrix}, \bar{H}_2(\theta) \doteq \begin{bmatrix} 0 & d_{\mu_\theta}^\top (P_{\pi_{\bar{\theta}^w}} - I)X\bar{\theta}^w \\ -(X^\top d_{\mu_\theta})\bar{\theta}^r & X^\top D_{\mu_\theta} P_{\pi_{\bar{\theta}^w}} X\bar{\theta}^w \end{bmatrix}, \\
\bar{G}(\theta) &\doteq \mathbb{E}_{(s,a)\sim d_{\mu_\theta}(\cdot), s'\sim p(\cdot|s,a)}[G_\theta(s,a,s')] = \begin{bmatrix} 1 & 0^\top \\ 0 & X^\top D_{\mu_\theta} X + \eta I\end{bmatrix}, \\
u^*(\theta) &\doteq \bar{G}(\theta)^{-1}\bar{h}(\theta).
\end{aligned} \tag{B.28}$$

We proceed to verifying Assumptions 3.1, 3.2, and 3.3 to invoke Theorem 3.1.

Assumption 3.1 can be verified via Corollary A.2 similarly to the proof in Section B.21.

For Assumption 3.3, Lemma C.9 suggests that assuming $\|X\| \leq 1$, $L_\mu \leq 1$, then

$$C_u = \max\left\{1, \eta^{-1}\right\}(\mathcal{O}(\|X\|) + \mathcal{O}(L_\mu)) + \max\left\{1, \eta^{-2}\right\}\mathcal{O}(\|X\|)$$

is a Lipschitz constant of $u^*(\theta)$. Take any $\xi \in (0,1)$, it is easy to see there exists positive constants C_2 and C_3 such that

$$\|X\| \leq C_2, L_\mu \leq C_3 \implies C_u \leq 1 - \xi. \tag{B.29}$$

Assumption 3.3, therefore, holds.

We now select proper R_{B_1} and R_{B_2} to fulfill Assumption 3.2. Using (B.29), it is easy to see

$$\|u^*(\theta)\| \leq C_4 + (1-\xi)R_{B_1}$$

for some positive constant C_4. For sufficiently large R_{B_1}, e.g.,

$$R_{B_1} \geq \max \frac{C_4}{\xi} + 1, \tag{B.30}$$

we have $\sup_\theta \|u^*(\theta)\| \leq R_{B_1} - \xi$. Selecting $R_{B_2} \in (R_{B_1} - \xi, R_{B_1})$ then fulfills Assumption 3.2.

Invoking Theorem 3.1 then yields that there exists a unique θ_∞ such that

$$u^*(\theta_\infty) = \theta_\infty \quad \text{and} \quad \lim_{t\to\infty} \theta_t = \lim_{t\to\infty} u_t = \theta_\infty.$$

We now show what θ_∞ is. We define

$$f(\theta) \doteq \bar{G}(\theta)^{-1}(\bar{h}_1(\theta) + \begin{bmatrix} 0 & d_{\mu_\theta}^\top (P_{\pi_\theta} - I)X \\ -X^\top d_{\mu_\theta} & X^\top D_{\mu_\theta} P_{\pi_\theta} X \end{bmatrix} \theta).$$

Similar to the proof of Theorem 9.1 in Section B.21, we can use the Brouwer fixed point theorem to find a $u_\eta^* \in \Gamma_{B_1}$ such that $f(u_\eta^*) = u_\eta^*$ if

$$R_{B_1} \geq C_5 \tag{B.31}$$

for some constant C_5. Then it is easy to see u_η^* is also the fixed point of $u^*(\cdot)$, implying $\theta_\infty = u_\eta^*$. Rearranging terms of $u_\eta^* = f(u_\eta^*)$ then completes the proof. In particular, the desired constants C_0 and C_1 can be deduced from (B.29), (B.30), and (B.31). □

B.24 Proof of Lemma 11.1

Proof. If (11.1) holds, then Lemma 5.6 implies that for any $u \in \Lambda_\mu$ and $\pi \in \Lambda_\pi$, $\Pi_{f_{\pi,\mu,n}} \mathcal{T}_\pi$ is a $\sqrt{\gamma}$-contraction in $\|\cdot\|_{f_{\pi,\mu,n}}$. We use $X w_{\pi,\mu,n}$ to denote its unique fixed point. Lemma 5.3 ensures that $X^\top D_{f_{\pi,\mu,n}}(I - \gamma P_\pi)X$ is p.d.. Similar to (B.14), it is easy to verify that

$$w_{\pi,\mu,n} = (X^\top D_{f_{\pi,\mu,n}}(I - \gamma P_\pi)X)^{-1} X^\top D_{f_{\pi,\mu,n}} r_\pi,$$

from which it is easy to see $w_{\pi,\mu,n}$ is continuous in μ and π since the invariant distribution d_μ is continuous in μ.

Similar to De Farias and Van Roy (2000), we first define several helper functions. For any policy $\mu \in \Lambda_\mu$, $\pi \in \Lambda_\Pi$, and $\eta > 0$, let

$$\begin{aligned}
g_{\mu,\pi}(w) &\doteq X^\top D_{f_{\pi,\mu,n}}(\mathcal{T}_\pi X w - X w) \\
&= X^\top D_{f_{\pi,\mu,n}} X (X^\top D_{f_{\pi,\mu,n}} X)^{-1} X^\top D_{f_{\pi,\mu,n}}(\mathcal{T}_\pi X w - X w) \\
&= X^\top D_{f_{\pi,\mu,n}}(\Pi_{f_{\pi,\mu,n}} \mathcal{T}_\pi X w - X w), \\
g(w) &\doteq X^\top D_{f_{\pi_w,\mu_w,n}}(\mathcal{T}_{\pi_w} X w - X w) \\
&= X^\top D_{f_{\pi_w,\mu_w,n}}(\Pi_{f_{\pi_w,\mu_w,n}} \mathcal{T}_{\pi_w} X w - X w), \\
z^\eta_{\mu,\pi}(w) &\doteq w + \eta g_{\mu,\pi}(w), \\
z^\eta(w) &\doteq w + \eta g(w).
\end{aligned}$$

We have

$$\begin{aligned}
& z^\eta_{\mu,\pi}(w) = w \\
\iff\ & g_{\mu,\pi}(w) = 0 \\
\iff\ & X^\top D_{f_{\pi,\mu,n}} \mathcal{T}_\pi X w = X^\top D_{f_{\pi,\mu,n}} X w \\
\iff\ & X(X^\top D_{f_{\pi,\mu,n}} X)^{-1} X^\top D_{f_{\pi,\mu,n}} \mathcal{T}_\pi X w = X(X^\top D_{f_{\pi,\mu,n}} X)^{-1} X^\top D_{f_{\pi,\mu,n}} X w \\
\iff\ & \Pi_{f_{\pi,\mu,n}} \mathcal{T}_\pi X w = X w,
\end{aligned}$$

i.e., Xw is a fixed point of $\Pi_{f_{\pi,\mu,n}} \mathcal{T}_\pi$ if and only if w is a fixed point of $z^\eta_{\mu,\pi}$. With the same procedure, we can also show

$$z^\eta(w) = w \iff \Pi_{f_{\pi_w,\mu_w,n}} \mathcal{T}_{\pi_w}(Xw) = Xw.$$

This suggests that to study the fixed points of emphatic approximate value iteration is to study the fixed points of z^η with any $\eta > 0$. To this end, we first study $z^\eta_{\mu,\pi}$ with the following lemma, which is analogous to Lemma 5.4 of De Farias and Van Roy (2000).

Lemma B.1. *There exists an $\eta_0 > 0$ such that for all $\eta \in (0, \eta_0)$, there exists a constant $\beta_\eta \in (0,1)$ such that for all $\mu \in \Lambda_\mu, \pi \in \Lambda_\Pi$,*

$$\left\| z^\eta_{\mu,\pi}(w) - w_{\pi,\mu,n} \right\| \leq \beta_\eta \| w - w_{\pi,\mu,n} \|.$$

Proof. By the contraction property of $\Pi_{f_{\pi,\mu,n}} \mathcal{T}_\pi$,

$$\left\| \Pi_{f_{\pi,\mu,n}} \mathcal{T}_\pi X w - X w_{\pi,\mu,n} \right\|_{f_{\pi,\mu,n}} \leq \sqrt{\gamma} \| X w - X w_{\pi,\mu,n} \|_{f_{\pi,\mu,n}}.$$

Consequently,

$$
\begin{aligned}
&(w - w_{\pi,\mu,n})^\top g_{\mu,\pi}(w) \\
=&(Xw - Xw_{\pi,\mu,n})^\top D_{f_{\pi,\mu,n}}(\Pi_{f_{\pi,\mu,n}}\mathcal{T}_\pi Xw - Xw) \\
=&(Xw - Xw_{\pi,\mu,n})^\top D_{f_{\pi,\mu,n}}\left(\Pi_{f_{\pi,\mu,n}}\mathcal{T}_\pi Xw - Xw_{\pi,\mu,n} + Xw_{\pi,\mu,n} - Xw\right) \\
\leq&\|Xw - Xw_{\pi,\mu,n}\|_{f_{\pi,\mu,n}}\left\|\Pi_{f_{\pi,\mu,n}}\mathcal{T}_\pi Xw - Xw_{\pi,\mu,n}\right\|_{f_{\pi,\mu,n}} - \|Xw - Xw_{\pi,\mu,n}\|^2_{f_{\pi,\mu,n}}
\end{aligned}
$$

(Cauchy-Schwarz inequality)

$$
\begin{aligned}
\leq&(\sqrt{\gamma} - 1)\|Xw - Xw_{\pi,\mu,n}\|^2_{f_{\pi,\mu,n}} \quad \text{(Property of contraction)} \\
=&(\sqrt{\gamma} - 1)(w - w_{\pi,\mu,n})^\top (X^\top D_{f_{\pi,\mu,n}} X)(w - w_{\pi,\mu,n}).
\end{aligned}
$$

Since $X^\top D_{f_{\pi,\mu,n}} X$ is symmetric and p.d., eigenvalues are continuous in the elements of the matrix, Λ_μ and Λ_Π are compact, by the extreme value theorem, there exists a constant $C_1 > 0$ (the infimum over the smallest eigenvalues of all $X^\top D_{f_{\pi,\mu,n}} X$), independent of μ and π, such that for all y,

$$
y^\top X^\top D_{f_{\pi,\mu,n}} Xy \geq C_1\|y\|^2.
$$

Consequently,

$$
(w - w_{\pi,\mu,n})^\top g_{\mu,\pi}(w) \leq (\sqrt{\gamma} - 1)C_1\|w - w_{\pi,\mu,n}\|^2. \tag{B.32}
$$

Moreover, let x_i be the i-th column X, we have

$$
\begin{aligned}
\|g_{\mu,\pi}(w)\|^2 &= \sum_{i=1}^K \left(x_i^\top D_{f_{\pi,\mu,n}}(\Pi_{f_{\pi,\mu,n}}\mathcal{T}_\pi Xw - Xw)\right)^2 \\
&\leq \sum_{i=1}^K \|x_i\|^2_{f_{\pi,\mu,n}}\left\|\Pi_{f_{\pi,\mu,n}}\mathcal{T}_\pi Xw - Xw\right\|^2_{f_{\pi,\mu,n}} \quad \text{(Cauchy-Schwarz inequality)} \\
&\leq \sum_{i=1}^K \|x_i\|^2_{f_{\pi,\mu,n}}\left(\left\|\Pi_{f_{\pi,\mu,n}}\mathcal{T}_\pi Xw - Xw_{\pi,\mu,n}\right\|_{f_{\pi,\mu,n}} + \|Xw_{\pi,\mu,n} - Xw\|_{f_{\pi,\mu,n}}\right)^2 \\
&\leq (\sqrt{\gamma} + 1)^2 \sum_{i=1}^K \|x_i\|^2_{f_{\pi,\mu,n}}\|Xw_{\pi,\mu,n} - Xw\|^2_{f_{\pi,\mu,n}} \quad (\sqrt{\gamma}\text{-contraction}) \\
&\leq (\sqrt{\gamma} + 1)^2 \left(\sum_{i=1}^K \|x_i\|^2_{f_{\pi,\mu,n}}\right)\left\|X^\top D_{f_{\pi,\mu,n}} X\right\|\|w - w_{\pi,\mu,n}\|^2.
\end{aligned}
$$

By the extreme value theorem,

$$\sup_{\mu\in\Lambda_\mu,\pi\in\Lambda_\pi}\left(\sum_{i=1}^{K}\|x_i\|^2_{f_{\pi,\mu,n}}\right)\left\|X^\top D_{f_{\pi,\mu,n}}X\right\|<\infty.$$

Consequently, there exists a constant $C_2>0$, independent of μ and π, such that

$$\|g_{\mu,\pi}(w)\|^2\le C_2\|w-w_{\pi,\mu,n}\|^2. \tag{B.33}$$

Combining (B.32) and (B.33) yields

$$\begin{aligned}\left\|z^\eta_{\mu,\pi}(w)-w_{\pi,\mu,n}\right\|^2&=\|w+\eta g_{\mu,\pi}(w)-w_{\pi,\mu,n}\|^2\\&=\|w-w_{\pi,\mu,n}\|^2+2\eta(w-w_{\pi,\mu,n})^\top g_{\mu,\pi}(w)+\eta^2\|g_{\mu,\pi}(w)\|^2\\&\le\left(1-2\eta(1-\sqrt{\gamma})\,C_1+\eta^2C_2\right)\|w-w_{\pi,\mu,n}\|^2\end{aligned}$$

Then for all $\eta<\eta_0\doteq 2C_1(1-\sqrt{\gamma})/C_2$, we have

$$\beta_\eta\doteq\sqrt{1-2\eta(1-\sqrt{\gamma})C_1+\eta^2C_2}<1.$$

□

We are now ready to study z^η with the previous lemma, analogously to Theorem 5.2 of De Farias and Van Roy (2000). Note $\mathcal{W}\doteq\{w_{\pi,\mu,n}\mid\mu\in\Lambda_\mu,\pi\in\Lambda_\Pi\}$ is a compact set by the continuity of $w_{\pi,\mu,n}$ in μ and π. Let $C_3\doteq\sup_{w\in W}\|w\|$ and take some η in $(0,\eta_0)$, we have for any w

$$\|z^\eta(w)\|\le\|z^\eta(w)-w_{\pi_w,\mu_w,n}\|+\|w_{\pi_w,\mu_w,n}\|$$

($w_{\pi_w,\mu_w,n}$ denotes the fixed point of $\Pi_{f_{\pi,\mu,n}}\mathcal{T}_\pi$ with μ being μ_w and π being π_w.)

$$\begin{aligned}&=\left\|z^\eta_{\mu_w,\pi_w}(w)-w_{\pi_w,\mu_w,n}\right\|+\|w_{\pi_w,\mu_w,n}\|\\&\le\beta_\eta\|w-w_{\pi_w,\mu_w,n}\|+C_3\\&\le\beta_\eta\|w\|+(1+\beta_\eta)C_3.\end{aligned}$$

Since $\beta_\eta<1$, we define

$$\mathcal{W}_2\doteq\left\{w\in\mathbb{R}^K\mid\|w\|<\frac{1+\beta_\eta}{1-\beta_\eta}C_3\right\}.$$

It is easy to verify that

$$w\in\mathcal{W}_2\implies z^\eta(w)\in\mathcal{W}_2.$$

The Brouwer fixed point theorem then asserts that $z^\eta(w)$ adopts at least one fixed point in $\mathcal{W}_2$, which completes the proof. □

B.25 Proof of Lemma 11.3

Proof. Recall

$$A_w = X^\top D_{f_{\pi_w,\mu_w,n}}(\gamma P_{\pi_w} - I)X,$$
$$f_{\pi_w,\mu_w,n} = \sum_{j=0}^{n} \gamma^j (P_{\pi_w}^\top)^j D_{\mu_w} i.$$

According to Lemma C.3, the invariant distribution d_μ is Lipschitz continuous w.r.t. μ in Λ_μ under Assumption 11.2. Consequently, Assumption 11.3 implies that D_{μ_w} is Lipschitz continuous in w. It is then easy to see from Lemma C.1 that $f_{\pi_w,\mu_w,n}$ is Lipschitz continuous in w. The Lipschitz continuity of A_w then follows easily, so does that of b_w. □

B.26 Proof of Theorem 11.4

Proof. Let $y_t \doteq (s_{t-n}, a_{t-n}, \ldots, s_t, a_t, s_{t+1})$. Define

$$\delta_w(s,a,s') \doteq r(s,a) + \gamma \sum_{a'} \pi_w(a'|s') x(s',a')^\top w - x(s,a)^\top w,$$
$$\bar{H}(w, y_t) \doteq \left(\sum_{j=0}^{n} \gamma^j \left(\prod_{k=t-j+1}^{t} \frac{\pi_w(a_k|s_k)}{\mu_w(a_k|s_k)} \right) i(s_{t-j}, a_{t-j}) \right) \delta_w(s_t, a_t, s_{t+1}) x(s_t, a_t).$$

Note here y_t is just a placeholder for defining the function g.

Let $Y_t \doteq (S_{t-n}, A_{t-n}, \ldots, S_t, A_t, S_{t+1})$ be a sequence of random variables generated by Algorithm 12. Then the update of w in Algorithm 12 can be expressed as

$$w_{t+1} = w_t + \alpha_t \bar{H}(w_t, Y_t).$$

We now prove Theorem 11.4 by verifying Assumptions A.30 - A.33 thus invoking Corollary A.7. Assumption A.30 is identical to Assumption 2.4.

Assumption A.31 is verified by the sampling procedure $A_{t+1} \sim \mu_{w_t}(\cdot|S_{t+1})$ in Algorithm 12 and Assumption 11.2. Similar to the proof of Theorem 5.4, it is easy to compute that the $h(w)$ of Assumption A.31 in our setting is

$$h(w) = A_w w + b_w.$$

For Assumption A.32, the Lipschitz continuity of the transition function is fulfilled by Assumption 11.3. By Assumption 11.2, there exists a constant $C_0 > 0$ such that

$\mu_w(a|s) \geq C_0 > 0$ holds for any w, a, s. Then it is easy to see $\bar{H}(w, y_t)$ is Lipschitz continuous on any compact set $Q \subset \mathbb{R}^K$.

We now verify Assumption A.33. For any $w_* \in \mathcal{W}_*$, let

$$U(w) \doteq \frac{1}{2}\|w - w_*\|^2.$$

Then Assumption A.33 (i) - (iii) trivially holds. To verify Assumption A.33 (iv), let $\tilde{w} \doteq w - w_*$. We have

$$\begin{aligned}
&\left\langle \frac{\mathrm{d}U(w)}{\mathrm{d}w}, h(w) \right\rangle \\
=&\langle w - w_*, h(w) - h(w_*)\rangle \quad (\text{Using } h(w_*) = 0) \\
=&\langle w - w_*, A_w w + b_w - A_w w_* + A_w w_* - A_{w_*} w_* - b_{w_*}\rangle \\
=&\tilde{w}^\top A_w \tilde{w} + \tilde{w}^\top (A_w - A_{w_*}) w_* + \tilde{w}^\top (b_w - b_{w_*}) \\
\leq&\tilde{w}^\top A_w \tilde{w} + \|\tilde{w}\|^2 (C_1 L_\mu + C_2 L_\pi) R + \|\tilde{w}\|^2 (C_3 L_\mu + C_4 L_\pi) \\
=&\frac{1}{2}\tilde{w}(A_w + A_w^\top)\tilde{w} + \|\tilde{w}\|^2 (C_1 L_\mu + C_2 L_\pi) R + \|\tilde{w}\|^2 (C_3 L_\mu + C_4 L_\pi) \\
=&-\tilde{w}\left(M(w) - ((C_1 L_\mu + C_2 L_\pi) R + (C_3 L_\mu + C_4 L_\pi))\, I\right)\tilde{w} \\
\leq&-\lambda'_{min}\|w - w_*\|^2,
\end{aligned}$$

where the last inequality results from the positive definiteness of the matrix

$$M(w) - ((C_1 L_\mu + C_2 L_\pi) R + (C_3 L_\mu + C_4 L_\pi))\, I$$

under Assumption 11.4. Assumption A.33 (iv) then follows immediately.

With Assumptions A.30 - A.33 fulfilled, (11.3) follows immediately from Corollary A.7. If there is a $w'_* \in \mathcal{W}_*$ and $w'_* \neq w_*$, repeating the previous procedure yields

$$\Pr\Big(\lim_{t\to\infty} w_t = w'_* \mid w_0 = w\Big) \geq 1 - C_{\mathcal{W}} \sum_{t=0}^{\infty} \alpha_t^2.$$

Using small enough $\{\alpha_t\}$ such that

$$1 - C_{\mathcal{W}} \sum_{t=0}^{\infty} \alpha_t^2 > 0.5$$

yields

$$\Pr\Big(\lim_{t\to\infty} w_t = w'_* \mid w_0 = w\Big) + \Pr\Big(\lim_{t\to\infty} w_t = w_* \mid w_0 = w\Big) > 1,$$

which is a contraction. Consequently, under the conditions of this theorem, $\mathcal{W}_*$ contains only one element, which completes the proof. □

B.27 Proof of Theorem 11.5

Proof. Readers familiar with Zou et al. (2019) should find this proof straightforward. We mainly follow the framework of Zou et al. (2019) except for some additional error terms introduced by the truncated followon traces. We include the proof here mainly for completeness. We, however, remark that it is the use of the truncated followon trace and Lemma 11.2 that make this straightforwardness possible in our off-policy setting.

Let $y_t \doteq (s_{t-n}, a_{t-n}, \dots, s_t, a_t, s_{t+1})$. For a sequence of weight vectors $(z_{t-n}, \dots, z_t)$ in $\mathbb{R}^K$, define

$$\delta_z(s,a,s') \doteq r(s,a) + \gamma \sum_{a'} \pi_z(a'|s') x(s',a')^\top z - x(s,a)^\top z,$$

$$g(z_{t-n}, \dots, z_t, y_t) \doteq \left(\sum_{j=0}^{n} \gamma^j \left(\prod_{k=t-j+1}^{t} \frac{\pi_{z_k}(a_k|s_k)}{\mu_{z_k}(a_k|s_k)} \right) i(s_{t-j}, a_{t-j}) \right) \delta_{z_t}(s_t, a_t, s_{t+1}) x(s_t, a_t).$$

Note here both y_t and $z_{t-n}, \dots, z_t$ are just placeholders for defining the function g, and we adopt the convention that $\prod_{k=i}^{j}(\cdot) = 1$ if $j < i$. Let $Y_t \doteq (S_{t-n}, A_{t-n}, \dots, S_t, A_t, S_{t+1})$ be a sequence of random variables generated by Algorithm 13. Then the update to w in Algorithm 13 can be expressed as

$$w_{t+1} = \Pi_R \left(w_t + \alpha_t g(w_{t-n}, \dots, w_t, Y_t) \right).$$

For the ease of presentation, we define

$$g(z, y_t) \doteq g(z, z, \dots, z, y_t),$$
$$\bar{g}(z) \doteq \mathbb{E}_{y_t \sim \mu_z(\cdot)}[g(z, y_t)]$$

as shorthand. By $y_t \sim \mu_z(\cdot)$, we mean

$$s_{t-n} \sim d_{\mu_z}(\cdot), a_{t-n} \sim \mu_z(\cdot|s_{t-n}), s_{t-n+1} \sim p(\cdot|s_{t-n}, a_{t-n}), \dots,$$
$$a_t \sim \mu_z(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t).$$

It can be easily computed that

$$\bar{g}(z) = X^\top D_{f_{\pi_z,\mu_z,n}} (\gamma P_{\pi_z} - I) X z + X^\top D_{f_{\pi_z,\mu_z,n}} r.$$

Consider a w_* in $\mathcal{W}_*$, we have

$$\bar{g}(w_*) \doteq A_{w_*} w_* + b_{w_*} = 0.$$

For any $\tau > 0$, we have

$$
\begin{aligned}
&\|w_{t+1} - w_*\|^2 \\
\leq &\|w_t + \alpha_t g(w_{t-n}, \ldots, w_t, Y_t) - w_*\|^2 \quad (\Pi_R \text{ is nonexpansive}) \\
= &\|w_t - w_*\|^2 + \alpha_t^2 \|g(w_{t-n}, \ldots, w_t, Y_t)\|^2 + 2\alpha_t \langle w_t - w_*, g(w_{t-n}, \ldots, w_t, Y_t)\rangle \\
= &\left\|w_t - w_*^2\right\| \\
&+ \alpha_t^2 \|g(w_{t-n}, \ldots, w_t, Y_t)\|^2 \qquad \text{(B.34)} \\
&+ 2\alpha_t \underbrace{\left(\langle w_t - w_*, g(w_{t-n}, \ldots, w_t, Y_t)\rangle - \langle w_{t-n-\tau} - w_*, \bar{g}(w_{t-n-\tau})\rangle\right)}_{err_t} \qquad \text{(B.35)} \\
&+ 2\alpha_t \langle w_{t-n-\tau} - w_*, \bar{g}(w_{t-n-\tau}) - \bar{g}(w_*)\rangle, \qquad \text{(B.36)}
\end{aligned}
$$

where we adopt the convention that $w_{t-n-\tau} \equiv w_0$ if $t - n - \tau < 0$. Using Lemmas B.2 and B.3 to bound (B.34) and (B.36) yields

$$
\mathbb{E}\left[\|w_{t+1} - w_*\|^2\right] \leq \mathbb{E}\left[\|w_t - w_*\|^2\right] + \alpha_t^2 U_g^2 + 2\alpha_t \mathbb{E}\left[err_t\right] - 2\alpha_t \alpha_\lambda \mathbb{E}\left[\|w_t - w_*\|^2\right].
$$

Dividing by $2\alpha_t$ in both sides yields

$$
\frac{1}{2\alpha_t}\mathbb{E}\left[\|w_{t+1} - w_*\|^2\right] \leq \frac{1}{2\alpha_t}\mathbb{E}\left[\|w_t - w_*\|^2\right] + \frac{1}{2}\alpha_t U_g^2 + \mathbb{E}\left[err_t\right] - \alpha_\lambda \mathbb{E}\left[\|w_t - w_*\|^2\right]. \tag{B.37}
$$

Using the definition of α_t in (11.4) yields

$$
\alpha_\lambda (t+1)\mathbb{E}\left[\|w_{t+1} - w_*\|^2\right] \leq \alpha_\lambda t \mathbb{E}\left[\|w_t - w_*\|^2\right] + \frac{1}{2}\alpha_t U_g^2 + \mathbb{E}\left[err_t\right].
$$

Lemma 1 of Zhang et al. (2021a) asserts that there are constants $C_0 > 0$ and $\kappa \in (0, 1)$ such that for any $w \in \mathbb{R}^K$ and $k \geq 0$, the chain $\{S_k\}_{k=0,1,\ldots}$ induced by the policy μ_w satisfies

$$
\sum_{s \in \mathcal{S}} |\Pr(S_k = s) - d_{\mu_w}(s)| \leq C_0 \kappa^k. \tag{B.38}
$$

For some fixed T, let $\tau_0 \doteq \min\{\tau : C_0 \kappa^\tau < \alpha_T\}$. Using the definition of α_t in (11.4), it can be easily computed that

$$
\tau_0 = \left\lceil \frac{\ln\left(2\alpha_\lambda (T+1) C_0\right)}{\ln \kappa^{-1}} \right\rceil = \mathcal{O}(\ln T)
$$

where $\lceil \cdot \rceil$ is the ceiling function. Here we assume T is large enough such that

$$
\tau_0 < T - n.
$$

Telescoping (B.37) for $t = 0, \ldots, T$ with $\tau = \tau_0$ yields

$$\begin{aligned}
&\alpha_\lambda T \mathbb{E}\left[\|w_T - w_*\|^2\right] \\
\leq &\sum_{t=0}^{T-1} \frac{1}{2}\alpha_t U_g^2 + \sum_{t=0}^{T-1} \mathbb{E}[err_t] \\
= &\sum_{t=0}^{T-1} \frac{1}{2}\frac{1}{2\alpha_\lambda(t+1)} U_g^2 + \sum_{t=0}^{\tau_0+n} \mathbb{E}[err_t] + \sum_{t=n+\tau_0+1}^{T-1} \mathbb{E}[err_t] \\
\leq &\frac{U_g^2}{4\alpha_\lambda} \ln T + (n+\tau_0+1)4RU_g + \sum_{t=n+\tau_0+1}^{T-1} \mathbb{E}[err_t], \qquad \text{(B.39)}
\end{aligned}$$

where the last inequality results from

$$\sum_{t=0}^{T-1} \frac{1}{t+1} \leq \ln T$$

and the first part of Lemma B.4. Using the second part of Lemma B.4 with $\tau = \tau_0$

to bound the last term of (B.39) yields

$$
\begin{aligned}
&\sum_{t=n+\tau_0+1}^{T-1} \mathbb{E}[err_t] \\
\leq &\sum_{t=n+\tau_0+1}^{T-1} \left(C_5 \sum_{j=t-n-\tau_0}^{t-1} \alpha_j + C_6 \sum_{k=t-n-\tau_0}^{t-2} \sum_{j=t-n-\tau_0}^{k} \alpha_j + C_7 C_0 \kappa^{\tau_0 - 1} \right) \\
= &\frac{1}{2\alpha_\lambda} \sum_{t=n+\tau_0+1}^{T-1} \left(C_5 \sum_{j=t-n-\tau_0}^{t-1} \frac{1}{j+1} + C_6 \sum_{k=t-n-\tau_0}^{t-2} \sum_{j=t-n-\tau_0}^{k} \frac{1}{j+1} + C_7 C_0 \kappa^{\tau_0 - 1} \right) \\
\leq &\frac{1}{2\alpha_\lambda} \sum_{t=n+\tau_0+1}^{T-1} \left(C_5 \ln \frac{t}{t-n-\tau_0} + C_6 \sum_{k=t-n-\tau_0}^{t-2} \ln \frac{k+1}{t-n-\tau_0} + C_7 C_0 \kappa^{\tau_0 - 1} \right) \\
\leq &\frac{1}{2\alpha_\lambda} \sum_{t=n+\tau_0+1}^{T-1} \left((C_5 + C_6(n+\tau_0)) \ln \frac{t}{t-n-\tau_0} + C_7 C_0 \kappa^{\tau_0 - 1} \right) \\
\leq &\frac{1}{2\alpha_\lambda} \sum_{t=n+\tau_0+1}^{T-1} \left((C_5 + C_6(n+\tau_0)) \ln \frac{t}{t-n-\tau_0} + \frac{C_7}{\kappa} \alpha_T \right) \\
= &\frac{1}{2\alpha_\lambda} \sum_{t=n+\tau_0+1}^{T-1} \left((C_5 + C_6(n+\tau_0)) \ln \frac{t}{t-n-\tau_0} + \frac{C_7}{2\alpha_\lambda \kappa} \frac{1}{T+1} \right) \\
\leq &\frac{1}{2\alpha_\lambda} (C_5 + C_6(n+\tau_0)) \ln \prod_{t=n+\tau_0+1}^{T-1} \frac{t}{t-n-\tau_0} + \frac{C_7}{2\alpha_\lambda \kappa} \\
\leq &\frac{1}{2\alpha_\lambda} (C_5 + C_6(n+\tau_0)) \ln \frac{(T-1)\cdots(T-1-n-\tau_0)}{(n+\tau_0)\cdots 1} + \frac{C_7}{2\alpha_\lambda \kappa} \\
\leq &\frac{1}{2\alpha_\lambda} (C_5 + C_6(n+\tau_0))(n+\tau_0) \ln T + \frac{C_7}{2\alpha_\lambda \kappa}.
\end{aligned}
$$

Plugging the above inequality back into (B.39) yields

$$
\begin{aligned}
&\mathbb{E}\left[\|w_T - w_*\|^2\right] \\
\leq &\frac{U_g^2}{4\alpha_\lambda^2} \frac{\ln T}{T} + \frac{4RU_g}{\alpha_\lambda} \frac{(n+\tau_0+1)}{T} + \frac{1}{2\alpha_\lambda^2} (C_5 + C_6(n+\tau_0))(n+\tau_0) \frac{\ln T}{T} + \frac{C_7}{2\alpha_\lambda^2 \kappa T} \\
= &\mathcal{O}\left(\frac{\ln^3 T}{T} \right).
\end{aligned}
$$

If there is also a $w_*' \in \mathcal{W}_*$, repeating the above procedure yields

$$
\mathbb{E}\left[\|w_T - w_*'\|^2\right] = \mathcal{O}\left(\frac{\ln^3 T}{T} \right).
$$

Consequently,

$$
\|w_* - w_*'\| = \mathbb{E}\left[\|w_* - w_*'\|\right] \leq \mathbb{E}\left[\|w_T - w_*'\|\right] + \mathbb{E}\left[\|w_T - w_*\|\right] = \mathcal{O}\left(\sqrt{\frac{\ln^3 T}{T}} \right).
$$

Letting T approaches infinity yields $w_* = w'_*$, i.e., $\mathcal{W}_*$ contains only one element under the condition of this theorem, which completes the proof. □

Lemma B.2. *(Bound of (B.34)) There exists a constant U_g such that*

$$\|g(w_{t-n}, \ldots, w_t, Y_t)\|^2 \leq U_g^2$$

Proof. Due to the projection Π_R, we have $\|w_t\| \leq R$ holds for all t. By the definition of g, it is easy to compute that

$$\|g(w_{t-n}, \ldots, w_t, Y_t)\| \leq \underbrace{(n+1)\rho_{max}^n i_{max}(r_{max} + (1+\gamma)Rx_{max})x_{max}}_{U_g},$$

where $i_{max} \doteq \max_{s,a} i(s,a), r_{max} \doteq \max_{s,a} |r(s,a)|, x_{max} \doteq \max_{s,a} \|x(s,a)\|$,

$$\rho_{\max} \doteq \sup_{\mu \in \Lambda_\mu, \pi \in \Lambda_\pi, s, a} \frac{\pi(s,a)}{\mu(s,a)}.$$

Assumption 11.2 and the extreme value theorem ensures that $\rho_{max} < \infty$. □

Lemma B.3. *(Bound of (B.36))*

$$\langle w_{t-n-\tau} - w_*, \bar{g}(w_{t-n-\tau}) - \bar{g}(w_*) \rangle \leq -\alpha_\lambda \|w_t - w_*\|^2$$

Proof. Let $\tilde{w} \doteq w_{t-n-\tau} - w_*$, we have

$$\begin{aligned}
&\langle w_{t-n-\tau} - w_*, \bar{g}(w_{t-n-\tau}) - \bar{g}(w_*) \rangle \\
=&\left\langle \tilde{w}, A_{w_{t-n-\tau}} w_{t-n-\tau} + b_{w_{t-n-\tau}} - A_{w_*} w_* - b_{w_*} \right\rangle \\
=&\left\langle \tilde{w}, A_{w_{t-n-\tau}} w_{t-n-\tau} - A_{w_*} w_{t-n-\tau} + A_{w_*} w_{t-n-\tau} - A_{w_*} w_* + b_{w_{t-n-\tau}} - b_{w_*} \right\rangle \\
=&\tilde{w}^\top A_{w_*} \tilde{w} + \tilde{w}(A_{w_{t-n-\tau}} - A_{w_*}) w_{t-n-\tau} + \tilde{w}^\top (b_{w_{t-n-\tau}} - b_{w_*}) \\
\leq&\tilde{w}^\top A_{w_*} \tilde{w} + \|\tilde{w}\|^2 (C_1 L_\mu + C_2 L_\pi) R + \|\tilde{w}\|^2 (C_3 L_\mu + C_4 L_\pi) \\
\leq& -\tilde{w}^\top \left(M(w_*) - ((C_1 L_\mu + C_2 L_\pi) R + (C_3 L_\mu + C_4 L_\pi)) I \right) \tilde{w} \\
\leq& -\lambda''_{min} \|\tilde{w}\|^2 \\
\leq& -\alpha_\lambda \|\tilde{w}\|^2.
\end{aligned}$$

□

Lemma B.4. *(Bound of (B.35)) Let*

$$err_t \doteq \langle w_t - w_*, g(w_{t-n}, \dots, w_t, Y_t)\rangle - \langle w_{t-n-\tau} - w_*, \bar{g}(w_{t-n-\tau})\rangle.$$

Then for any t and τ,

$$\|err_t\| \le 4RU_g.$$

If $t - n - \tau > 0$, there exist positive constants C_5, C_6, independent of t, such that

$$\mathbb{E}\left[err_t\right] \le C_5 \sum_{j=t-n-\tau}^{t-1} \alpha_j + C_6 \sum_{k=t-n-\tau}^{t-2} \sum_{j=t-n-\tau}^{k} \alpha_j + C_7 C_0 \kappa^{\tau-1}.$$

Proof. If $t - n - \tau < 0$,

$$\|err_t\| \le \|w_t - w_*\|\|g(w_{t-n}, \dots, w_t, Y_t)\| + \|w_{t-n-\tau} - w_*\|\|\bar{g}(w_{t-n-\tau})\| \le 4RU_g.$$

When $t-n-\tau > 0$, similar to Zou et al. (2019), we define an auxiliary Markov chain $\left\{\tilde{S}_t, \tilde{A}_t\right\}$ as

$$\left\{\tilde{S}_t, \tilde{A}_t\right\} : \cdots \underbrace{\to}_{\mu_{w_{t-n-\tau}}} S_{t-n-\tau+2} \underbrace{\to}_{\mu_{w_{t-n-\tau}}} \tilde{S}_{t-n-\tau+3} \underbrace{\to}_{\mu_{w_{t-n-\tau}}} \tilde{S}_{t-n-\tau+4} \to \cdots,$$

$$\left(\{S_t, A_t\} : \cdots \underbrace{\to}_{\mu_{w_{t-n-\tau}}} S_{t-n-\tau+2} \underbrace{\to}_{\mu_{w_{t-n-\tau+1}}} S_{t-n-\tau+3} \underbrace{\to}_{\mu_{w_{t-n-\tau+2}}} S_{t-n-\tau+4} \to \cdots\right)$$

i.e., the new chain is the same as the chain generated by Algorithm 13 (i.e., the chain (S_t, A_t)) before $S_{t-n-\tau+2}$, after which the new chain is generated by following a fixed behavior policy $\mu_{w_{t-n-\tau}}$ instead of the changing behavior policies $\mu_{w_{t-n-\tau+1}}, \mu_{w_{t-n-\tau+2}}, \cdots$ as the original chain. Let

$$\tilde{Y}_t \doteq (\tilde{S}_{t-n}, \tilde{A}_{t-n}, \dots, \tilde{S}_t, \tilde{A}_t, \tilde{S}_{t+1}),$$

we have

$$\begin{aligned}
err_t =& \langle w_t - w_*, g(w_{t-n}, \dots, w_t, Y_t)\rangle - \langle w_{t-n-\tau} - w_*, \bar{g}(w_{t-n-\tau}\rangle \\
=& \langle w_t - w_*, g(w_{t-n}, \dots, w_t, Y_t) - g(w_t, Y_t)\rangle && \text{(B.40)} \\
&+ \langle w_t - w_*, g(w_t, Y_t)\rangle - \langle w_{t-n-\tau} - w_*, g(w_{t-n-\tau}, Y_t)\rangle && \text{(B.41)} \\
&+ \left\langle w_{t-n-\tau} - w_*, g(w_{t-n-\tau}, Y_t) - g(w_{t-n-\tau}, \tilde{Y}_t)\right\rangle && \text{(B.42)} \\
&+ \left\langle w_{t-n-\tau} - w_*, g(w_{t-n-\tau}, \tilde{Y}_t) - \bar{g}(w_{t-n-\tau})\right\rangle. && \text{(B.43)}
\end{aligned}$$

Using Lemmas B.5, B.6, B.7, and B.9 to bound (B.40), (B.41), (B.42), and (B.43) yields

$$\begin{aligned}
\mathbb{E}\left[err_t\right] \leq & 2nRL_g \sum_{j=t-n}^{t-1} \alpha_j + (2RL_g + U_g)U_g \sum_{j=t-n-\tau}^{t-1} \alpha_j \\
& + 2R|\mathcal{A}|L_\mu U_g^2 \sum_{k=t-n-\tau}^{t-2} \sum_{j=t-n-\tau}^{k} \alpha_j + 2RU_g C_0 \kappa^{\tau-1} \\
\leq & \underbrace{(2nRL_g + (2RL_g + U_g)U_g)}_{C_5} \sum_{j=t-n-\tau}^{t-1} \alpha_j \\
& + \underbrace{2R|\mathcal{A}|L_\mu U_g^2}_{C_6} \sum_{k=t-n-\tau}^{t-2} \sum_{j=t-n-\tau}^{k} \alpha_j + \underbrace{2RU_g}_{C_7} C_0 \kappa^{\tau-1}
\end{aligned}$$

□

Lemma B.5. *(Bound of (B.40)) There exists a positive constant L_g such that*

$$\langle w_t - w_*, g(w_{t-n}, \ldots, w_t, Y_t) - g(w_t, Y_t) \rangle \leq 2nRL_g \sum_{j=t-n}^{t-1} \alpha_j.$$

Proof. First, for any $t' > t$, we have

$$\|w_{t'} - w_t\| \leq U_g \sum_{j=t}^{t'-1} \alpha_j$$

by using triangle inequalities with $w_{t+1}, w_{t+2}, \ldots, w_{t'-1}$ and Lemma B.2. It is then easy to show that $g(w_{t-n}, \ldots, w_t, Y_t)$ is Lipschitz in its first argument:

$$\begin{aligned}
& \|g(w_{t-n}, w_{t-n+1}, w_{t-n+2}, \ldots, w_t, Y_t) - g(w_t, w_{t-n+1}, w_{t-n+2}, \ldots, w_t, Y_t)\| \\
\leq & \underbrace{\frac{(n+1)(L_\mu + L_\pi)\rho_{max}^n (r_{max} + 2x_{max}R)x_{max}}{\mu_{min}^2}}_{L_g} \|w_{t-n} - w_t\| \leq L_g U_g \sum_{j=t-n}^{t-1} \alpha_j,
\end{aligned}$$

where $\mu_{min} \doteq \inf_{s,a} \mu_w(a|s)$. By the extreme value theorem, Assumption 11.2 implies that $\mu_{min} > 0$. Similarly, g is also Lipschitz continuous in its second argument:

$$\|g(w_t, w_{t-n+1}, w_{t-n+2} \ldots, w_t, Y_t) - g(w_t, w_t, w_{t-n+2}, \ldots, w_t, Y_t)\| \leq L_g U_g \sum_{j=t-n+1}^{t-1} \alpha_j.$$

Repeating this procedure for the third to n-th argument $(w_{t-n+2}, \ldots, w_{t-1})$ and putting them together with the triangle inequality yields

$$\|g(w_{t-n}, \ldots, w_t, Y_t) - g(w_t, Y_t)\| \leq nL_g U_g \sum_{j=t-n}^{t-1} \alpha_j.$$

Consequently,

$$\langle w_t - w_*, g(w_{t-n}, \ldots, w_t, Y_t) - g(w_t, Y_t)\rangle \leq 2nRL_g U_g \sum_{j=t-n}^{t-1} \alpha_j.$$

□

Lemma B.6. *(Bound of (B.41))*

$$\langle w_t - w_*, g(w_t, Y_t)\rangle - \langle w_{t-n-\tau} - w_*, g(w_{t-n-\tau}, Y_t)\rangle \leq (2RL_g + U_g)U_g \sum_{j=t-n-\tau}^{t-1} \alpha_j$$

Proof.

$$\begin{aligned}
&\langle w_t - w_*, g(w_t, Y_t)\rangle - \langle w_{t-n-\tau} - w_*, g(w_{t-n-\tau}, Y_t)\rangle \\
=&\langle w_t - w_*, g(w_t, Y_t) - g(w_{t-n-\tau}, Y_t)\rangle - \langle w_{t-n-\tau} - w_*, g(w_{t-n-\tau}, Y_t)\rangle \\
&+ \langle w_t - w_*, g(w_{t-n-\tau}, Y_t)\rangle \\
=&\langle w_t - w_*, g(w_t, Y_t) - g(w_{t-n-\tau}, Y_t)\rangle + \langle w_t - w_{t-n-\tau}, g(w_{t-n-\tau}, Y_t)\rangle \\
\leq&2RL_g\|w_t - w_{t-n-\tau}\| + U_g\|w_t - w_{t-n-\tau}\| \\
\leq&(2RL_g + U_g)U_g \sum_{j=t-n-\tau}^{t-1} \alpha_j
\end{aligned}$$

□

Lemma B.7. *(Bound of (B.42))*

$$\mathbb{E}\left[\left\langle w_{t-n-\tau} - w_*, g(w_{t-n-\tau}, Y_t) - g(w_{t-n-\tau}, \tilde{Y}_t)\right\rangle\right] \leq 2R|\mathcal{A}|L_\mu U_g^2 \sum_{k=t-n-\tau}^{t-2} \sum_{j=t-n-\tau}^{k} \alpha_j$$

Proof. Let $\Sigma_{t-n-\tau} \doteq (w_0, w_1, \ldots, w_{t-n-\tau}, S_0, A_0, \ldots, S_{t-n-\tau+1}, A_{t-n-\tau+1})$. We have

$$\begin{aligned}
&\mathbb{E}\left[\left\langle w_{t-n-\tau} - w_*, g(w_{t-n-\tau}, Y_t) - g(w_{t-n-\tau}, \tilde{Y}_t)\right\rangle\right] \\
=&\mathbb{E}\left[\mathbb{E}\left[\left\langle w_{t-n-\tau} - w_*, g(w_{t-n-\tau}, Y_t) - g(w_{t-n-\tau}, \tilde{Y}_t)\right\rangle \mid \Sigma_{t-n-\tau}\right]\right]
\end{aligned}$$

(Law of total expectation)

$$=\mathbb{E}\left[\left\langle w_{t-n-\tau}-w_*, \mathbb{E}\left[g(w_{t-n-\tau},Y_t)-g(w_{t-n-\tau},\tilde{Y}_t)\mid \Sigma_{t-n-\tau}\right]\right\rangle\right]$$

(Conditional independence)

$$\begin{aligned}
\leq&\mathbb{E}\left[\|w_{t-n-\tau}-w_*\|\left\|\mathbb{E}\left[g(w_{t-n-\tau},Y_t)-g(w_{t-n-\tau},\tilde{Y}_t)\mid \Sigma_{t-n-\tau}\right]\right\|\right]\\
\leq&2R\mathbb{E}\left[\left\|\mathbb{E}\left[g(w_{t-n-\tau},Y_t)-g(w_{t-n-\tau},\tilde{Y}_t)\mid \Sigma_{t-n-\tau}\right]\right\|\right]\\
\leq&2R|\mathcal{A}|L_\mu U_g^2\sum_{k=t-n-\tau}^{t-2}\sum_{j=t-n-\tau}^{k}\alpha_j,
\end{aligned}$$

where the last inequality comes from Lemma B.8. □

Lemma B.8.

$$\left\|\mathbb{E}\left[g(w_{t-n-\tau},Y_t)-g(w_{t-n-\tau},\tilde{Y}_t)\mid \Sigma_{t-n-\tau}\right]\right\|\leq|\mathcal{A}|L_\mu U_g^2\sum_{k=t-n-\tau}^{t-2}\sum_{j=t-n-\tau}^{k}\alpha_j$$

Proof. In the proof of this lemma, all expectations ($\mathbb{E}$) and probabilities (Pr) are conditioned on $\Sigma_{t-n-\tau}$. We suppress this condition in the presentation for improving readability. Given $t,n,\tau,\Sigma_{t-n-\tau}$, for any time step j such that $t-n-\tau+1\leq j\leq t$, we use $\mathcal{W}_j\subset\mathbb{R}^K$ to denote the set of all possible values of w_j. It is easy to see that $\mathcal{W}_j$ is always a *finite* set depending on $t,n,\tau,\Sigma_{t-n-\tau}$. This allows us to use summation instead of integral to further improve readability. We have

$$\begin{aligned}
&\left\|\mathbb{E}\left[g(w_{t-n-\tau},\tilde{Y}_t)-g(w_{t-n-\tau},Y_t)\right]\right\|\\
=&\left\|\sum_{y_t}\left(\Pr\left(\tilde{Y}_t=y_t\right)-\Pr(Y_t=y_t)\right)g(w_{t-n-\tau},y_t)\right\|
\end{aligned}$$

(Conditional independence of Y_t and $\tilde{Y}_t$ given $\Sigma_{t-n-\tau}$)

$$\leq U_g\sum_{y_t}\left|\Pr\left(\tilde{Y}_t=y_t\right)-\Pr(Y_t=y_t)\right|.$$

In the rest of this proof we bound $\left|\Pr\left(\tilde{Y}_t=y_t\right)-\Pr(Y_t=y_t)\right|$. To start,

$$\begin{aligned}
&\Pr(Y_t=y_t)\\
=&\sum_{z_{t-1}\in\mathcal{W}_{t-1}}\Pr(w_{t-1}=z_{t-1},A_t=a_t,S_{t+1}=s_{t+1},S_{t-n}=s_{t-n},\ldots,S_t=s_t)
\end{aligned}$$

(Law of total probability)

$$= \sum_{z_{t-1}} \Pr\left(A_t = a_t, S_{t+1} = s_{t+1} \mid \begin{smallmatrix} S_{t-n}=s_{t-n} \\ \cdots \\ S_t=s_t \\ w_{t-1}=z_{t-1} \end{smallmatrix}\right) \Pr\left(w_{t-1} = z_{t-1} \mid \begin{smallmatrix} S_{t-n}=s_{t-n} \\ \cdots \\ S_t=s_t \end{smallmatrix}\right)$$
$$\times \Pr(S_{t-n} = s_{t-n}, \ldots, S_t = s_t)$$

(Chain rule of joint distribution)

$$= \sum_{z_{t-1}} \mu_{z_{t-1}}(a_t|s_t) p(s_{t+1}|s_t, a_t) \Pr\left(w_{t-1} = z_{t-1} \mid \begin{smallmatrix} S_{t-n}=s_{t-n} \\ \cdots \\ S_t=s_t \end{smallmatrix}\right) \Pr(S_{t-n} = s_{t-n}, \ldots, S_t = s_t).$$

Further,

$$\Pr\left(\tilde{Y}_t = y_t\right)$$
$$= \mu_{w_{t-n-\tau}}(a_t|s_t) p(s_{t+1}|s_t, a_t) \Pr\left(\tilde{S}_{t-n} = s_{t-n}, \ldots, \tilde{S}_t = s_t\right)$$

(Definition of the auxiliary chain)

$$= \mu_{w_{t-n-\tau}}(a_t|s_t) p(s_{t+1}|s_t, a_t) \Pr\left(\tilde{S}_{t-n} = s_{t-n}, \ldots, \tilde{S}_t = s_t\right) \sum_{z_{t-1}} \Pr\left(w_{t-1} = z_{t-1} \mid \begin{smallmatrix} S_{t-n}=s_{t-n} \\ \cdots \\ S_t=s_t \end{smallmatrix}\right).$$

Consequently,

$$
\begin{aligned}
&\sum_{y_t} \Big|\Pr(Y_t = y_t) - \Pr\Big(\tilde{Y}_t = y_t\Big)\Big| \\
\leq &\sum_{s_{t-n},\ldots,s_t,a_t,z_{t-1}} \Pr\Big(w_{t-1} = z_{t-1} \mid \substack{S_{t-n}=s_{t-n} \\ \cdots \\ S_t=s_t}\Big) \times \\
&\Big|\mu_{z_{t-1}}(a_t|s_t)\Pr(S_{t-n},\ldots,S_t = s_t) - \mu_{w_{t-n-\tau}}(a_t|s_t)\Pr\Big(\tilde{S}_{t-n} = s_{t-n},\ldots,\tilde{S}_t = s_t\Big)\Big| \\
\leq &\sum_{s_{t-n},\ldots,s_t,a_t,z_{t-1}} \Pr\Big(w_{t-1} = z_{t-1} \mid \substack{S_{t-n}=s_{t-n} \\ \cdots \\ S_t=s_t}\Big) \times \\
&\Bigg(\Big|\mu_{z_{t-1}}(a_t|s_t)\Pr(S_{t-n},\ldots,S_t = s_t) - \mu_{w_{t-n-\tau}}(a_t|s_t)\Pr(S_{t-n} = s_{t-n},\ldots,S_t = s_t)\Big| + \\
&\Big|\mu_{w_{t-n-\tau}}(a_t|s_t)\Pr(S_{t-n},\ldots,S_t = s_t) - \mu_{w_{t-n-\tau}}(a_t|s_t)\Pr\Big(\tilde{S}_{t-n} = s_{t-n},\ldots,\tilde{S}_t = s_t\Big)\Big|\Bigg) \\
\leq &\sum_{s_{t-n},\ldots,s_t,a_t,z_{t-1}} \Pr\Big(w_{t-1} = z_{t-1} \mid \substack{S_{t-n}=s_{t-n} \\ \cdots \\ S_t=s_t}\Big) \times \\
&\Bigg(\Big|\mu_{z_{t-1}}(a_t|s_t) - \mu_{w_{t-n-\tau}}(a_t|s_t)\Big|\Pr(S_{t-n} = s_{t-n},\ldots,S_t = s_t) + \\
&\mu_{w_{t-n-\tau}}(a_t|s_t)\Big|\Pr(S_{t-n},\ldots,S_t = s_t) - \Pr\Big(\tilde{S}_{t-n} = s_{t-n},\ldots,\tilde{S}_t = s_t\Big)\Big|\Bigg) \qquad \text{(B.44)}
\end{aligned}
$$

Since $z_{t-1} \in \mathcal{W}_{t-1}$, we have

$$
|\mu_{z_{t-1}}(a_t|s) - \mu_{w_{t-n-\tau}}(a_t|s)| \leq L_\mu \|z_{t-1} - w_{t-n-\tau}\| \leq L_\mu U_g \sum_{j=t-n-\tau}^{t-2} \alpha_j.
$$

Plugging the above inequality back to (B.44) yields

$$
\begin{aligned}
&\sum_{y_t} \Big|\Pr(Y_t = y_t) - \Pr\Big(\tilde{Y}_t = y_t\Big)\Big| \\
\leq &\sum_{s_{t-n},\ldots,s_t,a_t,z_{t-1}} \Pr\Big(w_{t-1} = z_{t-1} \mid \substack{S_{t-n}=s_{t-n} \\ \cdots \\ S_t = s_t}\Big) \times \\
&\Bigg(\Pr(S_{t-n} = s_{t-n}, \ldots, S_t = s_t) L_\mu U_g \sum_{j=t-n-\tau}^{t-2} \alpha_j + \\
&\mu_{w_{t-n-\tau}}(a_t|s_t)\Big|\Pr(S_{t-n}, \ldots, S_t = s_t) - \Pr\Big(\tilde{S}_{t-n} = s_{t-n}, \ldots, \tilde{S}_t = s_t\Big)\Big|\Bigg) \\
= &|\mathcal{A}| L_\mu U_g \sum_{j=t-n-\tau}^{t-2} \alpha_j + \sum_{s_{t-n},\ldots,s_t} \Big|\Pr(S_{t-n}, \ldots, S_t = s_t) - \Pr\Big(\tilde{S}_{t-n} = s_{t-n}, \ldots, \tilde{S}_t = s_t\Big)\Big|.
\end{aligned}
$$

Recursively using the above inequality $n+1$ times yields

$$
\begin{aligned}
&\sum_{y_t} \Big|\Pr\Big(\tilde{Y}_t = y_t\Big) - \Pr(Y_t = y_t)\Big| \qquad \text{(B.45)} \\
\leq &|\mathcal{A}| L_\mu U_g \left(\sum_{j=t-n-\tau}^{t-2} \alpha_j + \cdots + \sum_{j=t-n-\tau}^{t-n-2} \alpha_j\right) + \sum_{s_{t-n}} \Big|\Pr(S_{t-n} = s_{t-n}) - \Pr\Big(\tilde{S}_{t-n} = s_{t-n}\Big)\Big|
\end{aligned}
$$

We now bound the last term in the above equation. We have

$$
\begin{aligned}
&\Pr(S_{t-n} = s_{t-n}) \\
= &\sum_s \Pr(S_{t-n-1} = s, S_{t-n} = s_{t-n}) \\
= &\sum_s \Pr(S_{t-n-1} = s) \Pr(S_{t-n} = s_{t-n} | S_{t-n-1} = s) \\
= &\sum_{s,a} \Pr(S_{t-n-1} = s) \Pr(S_{t-n} = s_{t-n}, A_{t-n-1} = a | S_{t-n-1} = s) \\
= &\sum_{s,a} \Pr(S_{t-n-1} = s) \mathbb{E}_{w_{t-n-2}} \left[\Pr(S_{t-n} = s_{t-n}, A_{t-n-1} = a | S_{t-n-1} = s, w_{t-n-2})\right] \\
= &\sum_{s,a} \Pr(S_{t-n-1} = s) \mathbb{E}_{w_{t-n-2}} \left[\mu_{w_{t-n-2}}(a|s) p(s_{t-n}|s,a)\right]
\end{aligned}
$$

Similarly,

$$
\Pr\Big(\tilde{S}_{t-n} = s_{t-n}\Big) = \sum_{s,a} \Pr\Big(\tilde{S}_{t-n-1} = s\Big) \mu_{w_{t-n-\tau}}(a|s) p(s_{t-n}|s,a).
$$

Consequently,

$$
\begin{aligned}
&\sum_{s_{t-n}} \left|\Pr(S_{t-n}=s_{t-n}) - \Pr\Big(\tilde{S}_{t-n}=s_{t-n}\Big)\right| \\
=&\sum_{s,a} \left|\Pr(S_{t-n-1}=s)\mathbb{E}_{w_{t-n-2}}\left[\mu_{w_{t-n-2}}(a|s)\right] - \Pr\Big(\tilde{S}_{t-n-1}=s\Big)\mu_{w_{t-n-\tau}}(a|s)\right| \\
\leq&\sum_{s,a} \left|\Pr(S_{t-n-1}=s)\mathbb{E}_{w_{t-n-2}}\left[\mu_{w_{t-n-2}}(a|s)\right] - \Pr\Big(\tilde{S}_{t-n-1}=s\Big)\mathbb{E}_{w_{t-n-2}}\left[\mu_{w_{t-n-2}}(a|s)\right]\right| + \\
&\sum_{s,a} \left|\Pr\Big(\tilde{S}_{t-n-1}=s\Big)\mathbb{E}_{w_{t-n-2}}\left[\mu_{w_{t-n-2}}(a|s)\right] - \Pr\Big(\tilde{S}_{t-n-1}=s\Big)\mu_{w_{t-n-\tau}}(a|s)\right| \\
=&\sum_{s} \left|\Pr(S_{t-n-1}=s) - \Pr\Big(\tilde{S}_{t-n-1}=s\Big)\right| + \\
&\sum_{s,a} \Pr\Big(\tilde{S}_{t-n-1}=s\Big)\left|\mathbb{E}_{w_{t-n-2}}\left[\mu_{w_{t-n-2}}(a|s)\right] - \mu_{w_{t-n-\tau}}(a|s)\right| \\
\leq&\sum_{s} \left|\Pr(S_{t-n-1}=s) - \Pr\Big(\tilde{S}_{t-n-1}=s\Big)\right| + \\
&\sum_{s,a} \Pr\Big(\tilde{S}_{t-n-1}=s\Big)\max_{s}\left|\mathbb{E}_{w_{t-n-2}}\left[\mu_{w_{t-n-2}}(a|s)\right] - \mu_{w_{t-n-\tau}}(a|s)\right|
\end{aligned}
$$

Since

$$
\begin{aligned}
&\left|\mathbb{E}_{w_{t-n-2}}\left[\mu_{w_{t-n-2}}(a|s)\right] - \mu_{w_{t-n-\tau}}(a|s)\right| \\
=&\left|\mathbb{E}_{w_{t-n-2}}\left[\mu_{w_{t-n-2}}(a|s) - \mu_{w_{t-n-\tau}}(a|s)\right]\right| \\
\leq&\mathbb{E}_{w_{t-n-2}}\left[\left|\mu_{w_{t-n-2}}(a|s) - \mu_{w_{t-n-\tau}}(a|s)\right|\right] \\
\leq&U_g L_\mu \sum_{j=t-n-\tau}^{t-n-3} \alpha_j,
\end{aligned}
$$

we have

$$
\begin{aligned}
&\sum_{s} \left|\Pr(S_{t-n}=s) - \Pr\Big(\tilde{S}_{t-n}=s\Big)\right| \\
\leq&\sum_{s} \left|\Pr(S_{t-n-1}=s) - \Pr\Big(\tilde{S}_{t-n-1}=s\Big)\right| + |\mathcal{A}|U_g L_\mu \sum_{j=t-n-\tau}^{t-n-3} \alpha_j.
\end{aligned}
$$

Applying the above inequality recursively yields

$$
\begin{aligned}
&\sum_{s} \left|\Pr(S_{t-n}=s) - \Pr\Big(\tilde{S}_{t-n}=s\Big)\right| \\
\leq&|\mathcal{A}|U_g L_\mu \left(\sum_{j=t-n-\tau}^{t-n-3} \alpha_j + \cdots + \sum_{j=t-n-\tau}^{t-n-\tau} \alpha_j\right)
\end{aligned}
\tag{B.46}
$$

as

$$\Pr(S_{t-n-\tau+2}=s)=\Pr\Big(\tilde{S}_{t-n-\tau+2}=s\Big)$$

by the construction of the auxiliary chain. Plugging (B.46) back to (B.45) yields

$$\sum_{y_t}\Big|\Pr\Big(\tilde{Y}_t=y_t\Big)-\Pr(Y_t=y_t)\Big|\leq|\mathcal{A}|L_\mu U_g\sum_{k=t-n-\tau}^{t-2}\sum_{j=t-n-\tau}^{k}\alpha_j,$$

which completes the proof. □

Lemma B.9. *(Bound of (B.43))*

$$\mathbb{E}\left[\left\langle w_{t-n-\tau}-w_*,g(w_{t-n-\tau},\tilde{Y}_t)-\bar{g}(w_{t-n-\tau})\right\rangle\right]\leq 2RU_gC_0\kappa^{\tau-1}$$

Proof.

$$\begin{aligned}
&\mathbb{E}\left[\left\langle w_{t-n-\tau}-w_*,g(w_{t-n-\tau},\tilde{Y}_t)-\bar{g}(w_{t-n-\tau})\right\rangle\right]\\
=&\mathbb{E}\left[\mathbb{E}\left[\left\langle w_{t-n-\tau}-w_*,g(w_{t-n-\tau},\tilde{Y}_t)-\bar{g}(w_{t-n-\tau})\right\rangle\mid\Sigma_{t-n-\tau}\right]\right]\\
=&\mathbb{E}\left[\left\langle w_{t-n-\tau}-w_*,\mathbb{E}\left[g(w_{t-n-\tau},\tilde{Y}_t)-\bar{g}(w_{t-n-\tau})\mid\Sigma_{t-n-\tau}\right]\right\rangle\right]\\
\leq&\mathbb{E}\left[\|w_{t-n-\tau}-w_*\|\left\|\mathbb{E}\left[g(w_{t-n-\tau},\tilde{Y}_t)-\bar{g}(w_{t-n-\tau})\mid\Sigma_{t-n-\tau}\right]\right\|\right]\\
\leq&2R\left\|\mathbb{E}\left[g(w_{t-n-\tau},\tilde{Y}_t)-\bar{g}(w_{t-n-\tau})\mid\Sigma_{t-n-\tau}\right]\right\|
\end{aligned}$$

We now bound $\left\|\mathbb{E}\left[g(w_{t-n-\tau},\tilde{Y}_t)-\bar{g}(w_{t-n-\tau})\mid\Sigma_{t-n-\tau}\right]\right\|$. In the rest of the proof, all expectations ($\mathbb{E}$) and probabilities (Pr) are conditioned on $\Sigma_{t-n-\tau}$. We suppress the condition in the presentation for improving readability. Let $\bar{Y}_t\doteq(\bar{S}_{t-n},\bar{A}_{t-n},\ldots,\bar{S}_t,\bar{A}_t,\bar{S}_{t+1})$ be a sequence of random variables such that

$$\bar{S}_{t-n}\sim d_{\mu_{w_{t-n-\tau}}}(\cdot),\bar{A}_{t-n}\sim\mu_{w_{t-n-\tau}}(\cdot|\bar{S}_{t-n}),\ldots,\bar{A}_t\sim\mu_{w_{t-n-\tau}}(\cdot|\bar{S}_t),\bar{S}_{t+1}\sim p(\cdot|S_t,A_t).$$

Then

$$
\begin{aligned}
&\left\|\mathbb{E}\left[g(w_{t-n-\tau}, \tilde{Y}_t) - \bar{g}(w_{t-n-\tau})\right]\right\| \\
=&\left\|\mathbb{E}\left[g(w_{t-n-\tau}, \tilde{Y}_t) - g(w_{t-n-\tau}, \bar{Y}_t)\right]\right\| \\
=&\left\|\sum_{y_t}\left(\Pr\left(\tilde{Y}_t = y_t\right) - \Pr\left(\bar{Y}_t = y_t\right)\right) g(w_{t-n-\tau}, y_t)\right\| \\
\leq& U_g \sum_{y_t}\left|\Pr\left(\tilde{Y}_t = y_t\right) - \Pr\left(\bar{Y}_t = y_t\right)\right| \\
=& U_g \sum_{y_t}\left|\Pr\left(\tilde{S}_{t-n} = s_{t-n}\right) - \Pr\left(\bar{S}_{t-n} = s_{t-n}\right)\right| \mu_{w_{t-n-\tau}}(a_{t-n}|s_{t-n}) p(s_{t-n+1}|s_{t-n}, a_{t-n}) \\
&\cdots \mu_{w_{t-n-\tau}}(a_t|s_t) p(s_{t+1}|s_t, a_t) \\
=& U_g \sum_{s_{t-n}}\left|\Pr\left(\tilde{S}_{t-n} = s_{t-n}\right) - \Pr\left(\bar{S}_{t-n} = s_{t-n}\right)\right| \\
\leq& U_g C_0 \kappa^{\tau-1} \quad \text{(Using (B.38) and the construction of the auxiliary chain)} \quad ,
\end{aligned}
$$

which completes the proof. □

B.28 Proof of Theorem 12.1

Proof. Consider the stochastic process $Y_t \doteq (S_t, A_t, S_{t+1})$ involving in the space

$$\mathcal{Y} \doteq \{(s, a, s') \mid s \in \mathcal{S}, a \in \mathcal{A}, s' \in \mathcal{S}, p(s'|s, a) > 0\}.$$

Assumption 2.8 ensures that the chain $\{Y_t\}$ is ergodic. We then use $d_{\mathcal{Y}}$ to denote its stationary distribution, i.e.,

$$d_{\mathcal{Y}}(s, a, s') = d_\mu(s)\mu(a|s)p(s'|s, a).$$

Let

$$d_t \doteq \begin{bmatrix} \kappa_t \\ w_t \end{bmatrix}.$$

Then the updates of $\{\kappa_t\}, \{w_t\}$ in Algorithm 14 can be expressed as

$$d_{t+1} = d_t + \alpha_t(h_{\theta_t}(Y_t) - G_{\theta_t}(Y_t)d_t),$$

where

$$
\begin{aligned}
h_\theta(s, a, s') &\doteq \begin{bmatrix} i(s')x(s') \\ 0 \end{bmatrix}, \\
G_\theta(s, a, s') &\doteq \begin{bmatrix} x(s')x(s')^\top & x(s')\left(x(s') - \gamma\rho_\theta(s, a)x(s)\right)^\top \\ -\left(x(s') - \gamma\rho_\theta(s, a)x(s)\right)x(s')^\top & \eta I \end{bmatrix}.
\end{aligned}
$$

Define

$$\bar{h}_\theta \doteq \mathbb{E}_{(s,a,s')\sim d_{\mathcal{Y}}(\cdot)}\left[h_\theta(s,a,s')\right] = \begin{bmatrix} X^\top D_\mu i \\ 0 \end{bmatrix},$$
$$\bar{G}_\theta \doteq \mathbb{E}_{(s,a,s')\sim d_{\mathcal{Y}}(\cdot)}\left[G_\theta(s,a,s')\right] = \begin{bmatrix} C_\mu & -A_{\pi_\theta,\mu}^\top \\ A_{\pi_\theta,\mu} & \eta I \end{bmatrix}.$$

We now prove the desired convergence via invoking Corollary A.2. Assumption A.1 follows immediately from the sampling rules in Algorithm 14. Assumption A.2 is identical to Assumption 2.4. Assumption A.3 holds thanks to Assumption 2.6, the boundedness of $\nabla\pi_\theta(a|s)$ in Assumption 12.2, and the two adaptive learning rates Γ_1 and Γ_2. To see Assumption A.9, consider any

$$d \doteq \begin{bmatrix} \kappa \\ w \end{bmatrix}.$$

It is easy to verify that

$$\begin{aligned} d^\top \bar{G}(\theta) d =& \kappa^\top C \kappa + \eta w^\top w \\ \geq& \lambda_{min}(C)\|\kappa\|^2 + \eta\|w\|^2 \\ \geq& \min\left\{\lambda_{min}(C), \eta\right\}\|d\|^2, \end{aligned}$$

where $\lambda_{min}(\cdot)$ denotes the minimum eigenvalue and Assumption 2.3 ensures that $\lambda_{min}(C)$ is strictly positive. Assumption A.9 is then verified. Assumption A.10 follows from Assumption 2.8. Assumption A.11 is obvious because we consider a finite MDP. Assumption A.12 follows immediately from Assumption 12.2. Invoking Corollary A.4 then completes the proof. □

B.29 Proof of Theorem 12.2

Proof. The proof is the same as the proof of Theorem 12.1 up to a change of notations and is thus omitted to avoid verbatim repetition. □

B.30 Proof of Theorem 12.3

Proof. This proof is inspired by Konda (2002). We first define the following shorthand:

$$\begin{aligned}
\psi_t \doteq& \rho_t \nabla \log \pi(A_t|S_t),\\
w_t^* \doteq& w_{\theta_t,\eta}^*,\\
u_t^* \doteq& u_{\theta_t,\eta}^*,\\
J(\theta) \doteq& J_{\pi_\theta,\mu},\\
\Delta_t \doteq& \rho_t (w_t^\top x_t)(u_t^\top \tilde{x}_t) \nabla \log \pi_\theta(A_t|S_t),\\
\hat{g}(\theta) \doteq& \sum_{s,a} \tilde{d}_\mu(s,a) \left(x(s)^\top w_{\theta,\eta}^*\right) \left(\tilde{x}(s,a)^\top u_{\theta,\eta}^*\right) \rho_\theta(s,a) \nabla \log \pi_\theta(a|s).
\end{aligned}$$

We recall that in (12.2), we have

$$b(\theta) = \nabla J(\theta) - \hat{g}(\theta).$$

We first decompose the incremental update to θ_t in Algorithm 14 as

$$\Gamma_1(w_t)(w_t^\top x_t)\Gamma_2(u_t)(u_t^\top \tilde{x}_t)\psi_t = e_t^{(1)} + e_t^{(2)} + g_t^*,$$

where

$$\begin{aligned}
e_t^{(1)} \doteq& (\Gamma_1(w_t) w_t - \Gamma_1(w_t^*) w_t^*)^\top x_t \Gamma_2(u_t) u_t^\top \tilde{x}_t \psi_t,\\
e_t^{(2)} \doteq& \Gamma_1(w_t^*) w_t^{*\top} x_t (\Gamma_2(u_t) u_t^\top - \Gamma_2(u_t^*) u_t^*)^\top \tilde{x}_t \psi_t,\\
g_t^* \doteq& \Gamma_1(w_t^*)(x_t^\top w_t^*)\Gamma_2(u_t^*)(\tilde{x}_t^\top u_t^*)\psi_t.
\end{aligned}$$

Then we have

$$\theta_{t+1} = \theta_t + \beta_t e_t^{(1)} + \beta_t e_t^{(2)} + \beta_t g_t^* + \beta_t \Gamma_1(w_t^*)\Gamma_2(u_t^*) \left(\nabla J(\theta_t) - \hat{g}(\theta_t) - b(\theta_t)\right)$$

Using the second order Taylor expansion and Cauchy-Schwarz inequality, we have

$$\begin{aligned}
J(\theta_{t+1}) \geq& J(\theta_t) + \beta_t \Gamma_1(w_t^*)\Gamma_2(u_t^*) \|\nabla J(\theta_t)\| \left(\|\nabla J(\theta_t)\| - \|b(\theta_t)\|\right) \qquad \text{(B.47)}\\
&+ \beta_t \nabla J(\theta_t)^\top (g_t^* - \Gamma_1(w_t^*)\Gamma_2(u_t^*)\hat{g}(\theta_t))\\
&+ \beta_t \nabla J(\theta_t)^\top e_t^{(1)}\\
&+ \beta_t \nabla J(\theta_t)^\top e_t^{(2)}\\
&- \frac{1}{2} C_0 \|\beta_t \Gamma_1(w_t)\Gamma_2(u_t)\Delta_t\|^2,
\end{aligned}$$

where C_0 reflects the bound of the Hessian of $J(\theta)$ (Lemma C.10). We will prove in following subsections that all noise terms in (B.47) are negligible. Namely,

Lemma B.10. $\lim_{t\to\infty} e_t^{(i)} = 0 \quad a.s. \quad (i = 1, 2)$

Lemma B.11. $\sum_t \|\beta_t \Gamma_1(w_t)\Gamma_2(u_t)\Delta_t\|^2$ *converges a.s.*

Lemma B.12. $\sum_t \beta_t \nabla J(\theta_t)^\top (g_t^* - \Gamma_1(w_t^*)\Gamma_2(u_t^*)\hat{g}(\theta_t))$ *converges a.s.*

Same as the section "Proof of Theorem 5.5" in Konda (2002), we now consider a sequence $\{k_i\}$ such that

$$k_0 = 0, \quad k_{i+1} = \min\{k \geq k_i \mid \sum_{l=k_i}^{k} \beta_l \geq T\}$$

for some constant $T > 0$. Telescoping (B.47) yields

$$\begin{aligned} J(\theta_{k_{i+1}}) \geq & J(\theta_{k_i}) + \delta_i \\ & + \sum_{t=k_i}^{k_{i+1}-1} \beta_t \Gamma_1(w_t^*)\Gamma_2(u_t^*)\|\nabla J(\theta_t)\| \left(\|\nabla J(\theta_t)\| - \|b(\theta_t)\|\right), \end{aligned} \tag{B.48}$$

where

$$\begin{aligned} \delta_i \doteq \sum_{t=k_i}^{k_{i+1}-1} \Big[& \beta_t \nabla J(\theta_t)^\top (g_t^* - \Gamma_1(w_t^*)\Gamma_2(u_t^*)\hat{g}(\theta_t)) + \beta_t \nabla J(\theta_t)^\top (e_t^{(1)} + e_t^{(2)}) \\ & - \frac{1}{2} C_0 \|\beta_t \Gamma_1(w_t)\Gamma_2(u_t)\Delta_t\|^2 \Big]. \end{aligned}$$

Lemmas C.10, B.10, B.11, and B.12 and the selection of $\{k_i\}$ imply

$$\lim_{i\to\infty} \delta_i = 0 \quad \text{a.s.} \quad .$$

Theorems 12.1 and 12.2 imply that

$$\sup_t \max\{\|w_t^*\|, \|u_t^*\|\} < \infty.$$

Assumption 12.1 then implies that there exists some constant $C_1 > 0$ such that

$$\inf_t \min\{\Gamma_1(w_t^*), \Gamma_2(w_t^*)\} \geq C_1.$$

We now proceed by contradiction. If

$$\liminf_t \left[\|\nabla J(\theta_t)\| - \|b(\theta_t)\|\right] \leq 0 \tag{B.49}$$

does not hold, there must exist t_0 and $\epsilon > 0$ such that $\forall t > t_0$,

$$\|\nabla J(\theta_t)\| - \|b(\theta_t)\| > \epsilon.$$

Then (B.48) implies that

$$J(\theta_{k_{i+1}}) \geq J(\theta_{k_i}) + \delta_i + TC_1^2\epsilon^2.$$

Telescoping the above inequality yields that

$$\lim_{i\to\infty} J(\theta_{k_i}) = \infty.$$

This is impossible because as $\gamma < 1$ and r is bounded. We, therefore, conclude that (B.49) must hold, which completes the proof. □

Proof of Lemma B.10

Proof. We show only $\lim_{t\to\infty} e_t^{(1)} = 0$. The convergence of $e_t^{(2)}$ is the same up to change of notations. Theorem 12.1 ensures that there exists a compact set $\mathcal{W} \in \mathbb{R}^{K_1}$ such that almost surely, $\forall t$,

$$w_t \in \mathcal{W},\ w_t^* \in \mathcal{W}, u_t \in \mathcal{W}.$$

It is easy to verify that the function $w \to \Gamma_1(w)w$ is Lipschitz continuous on $\mathcal{W}$. Consequently, the desired convergence follows immediately from Theorem 12.1, Assumption 12.1, and Lemma C.10. □

Proof of Lemma B.11

Proof. Using Lemma C.10, Theorems 12.1 and 12.2, and Assumption 12.1, it is easy to see that there exists a constant $C_0 > 0$ such that

$$\sup_t \|\Gamma_1(w_t)\Gamma_2(u_t)\Delta_t\| \leq C_0.$$

Consequently,

$$\sum_t \|\beta_t\Gamma_1(w_t)\Gamma_2(u_t)\Delta_t\|^2 \leq C_0^2 \sum_t \beta_t^2 < \infty.$$

□

Proof of Lemma B.12

Proof. In this subsection we write $w_{\theta,\eta}^*$ as w_θ^* for simplifying notation and define

$$g_\theta^*(s,a) \doteq \Gamma_1(w_\theta^*)\Gamma_2(u_\theta^*)(x(s)^\top w_\theta^*)(\tilde{x}(s)^\top u_\theta^*)\rho_\theta(s,a)\nabla \log \pi_\theta(a|s). \tag{B.50}$$

So $g_t^* \doteq g_{\theta_t}^*(S_t, A_t)$. From Lemma C.10, it is easy to see that

$$\sup_\theta \|g_\theta^*\| < \infty.$$

We first make a transformation of the original noise using the differential Bellman equation as in the proof of Corollary A.2. Define

$$\begin{aligned}\mathcal{Y} \doteq& \mathcal{S} \times \mathcal{A},\\ Y_t \doteq& (S_t, A_t),\\ y \doteq& (s, a).\end{aligned}$$

For every integer i in $[1, K_3]$, we consider the MRP with the reward function $g_{\theta,i}^* : \mathcal{Y} \to \mathbb{R}$, where $g_{\theta,i}^*(y)$ is the i-the element of $g_\theta^*(y)$. The average reward is therefore

$$\bar{g}(\theta)_i \doteq \sum_{s,a} d_\mu(s)\mu(a|s) g_{\theta,i}^*(s, a).$$

Here we have defined a vector $\bar{g}(\theta) \in \mathbb{R}^{K_3}$ by defining its each element $\bar{g}(\theta)_i$. It is easy to verify that

$$\bar{g}(\theta) \doteq \Gamma_1(w_\theta^*)\Gamma_2(u_\theta^*)\hat{g}(\theta).$$

The differential value function of this MRP is then

$$\hat{v}_{\theta,i} \doteq (I - \tilde{P}_\mu + 1\tilde{d}_\mu^\top)^{-1}(I - 1\tilde{d}_\mu^\top) g_{\theta,i}^*. \tag{B.51}$$

See, e.g., (8.2.2) in Puterman (2014) for a proof. These differential value functions define a vector-valued function $\hat{v}_\theta : \mathcal{Y} \to \mathbb{R}^K$. Namely, the i-the element of $\hat{v}_\theta(y)$ is $\hat{v}_{\theta,i}$. It is then easy to see that

$$\sup_\theta \|\hat{v}_\theta\| < \infty.$$

According to the differential Bellman equation, we have

$$\hat{v}_\theta(s, a) = g_\theta^*(s, a) - \bar{g}(\theta) + \sum_{s',a'} \tilde{P}_\mu((s, a), (s', a'))\hat{v}_\theta(s', a').$$

Now we are ready to decompose the noise $\nabla J(\theta_t)^\top (g_t^* - \Gamma_1(w_t^*)\Gamma_2(u_t^*)\hat{g}(\theta_t))$ as

$$\begin{aligned}&\nabla J(\theta_t)^\top (g_t^* - \Gamma_1(w_t^*)\Gamma_2(u_t^*)\hat{g}(\theta_t))\\ =&\nabla J(\theta_t)^\top (g_{\theta_t}^*(S_t, A_t) - \bar{g}(\theta_t)) \quad \text{(Definition of } g_{\theta_t}^* \text{ and } \bar{g}(\theta_t))\\ =&\nabla J(\theta_t)^\top \left(\hat{v}_{\theta_t}(S_t, A_t) - \sum_{s',a'} p(s'|S_t, A_t)\mu(a'|s')\hat{v}_{\theta_t}(s', a')\right) \quad \text{(Using (B.51))}\\ =&\sum_{i=1}^4 \epsilon_t^{(i)},\end{aligned}$$

where

$$\epsilon_t^{(1)} \doteq \nabla J(\theta_t)^\top \Big(\hat{v}_{\theta_t}(S_{t+1}, A_{t+1}) - \sum_{s',a'} p(s'|S_t, A_t)\mu(a'|s')\hat{v}_{\theta_t}(s', a') \Big),$$

$$\epsilon_t^{(2)} \doteq \frac{\beta_{t-1}\nabla J(\theta_{t-1})^\top \hat{v}_{\theta_{t-1}}(S_t, A_t) - \beta_t \nabla J(\theta_t)^\top \hat{v}_{\theta_t}(S_{t+1}, A_{t+1})}{\beta_t},$$

$$\epsilon_t^{(3)} \doteq \frac{\beta_t - \beta_{t-1}}{\beta_t} \nabla J(\theta_{t-1})^\top \hat{v}_{\theta_{t-1}}(S_t, A_t),$$

$$\epsilon_t^{(4)} \doteq \nabla J(\theta_t)^\top \hat{v}_{\theta_t}(S_t, A_t) - \nabla J(\theta_{t-1})^\top \hat{v}_{\theta_{t-1}}(S_t, A_t).$$

We now show $\sum_t \beta_t \epsilon_t^{(i)}$ converges almost surely for $i = 1, 2, 3, 4$.

(1) We proceed via a Martingale convergence theorem (Lemma C.5). Let

$$\mathcal{F}_l \doteq \sigma(S_0, A_0, \theta_0, \ldots, S_l, A_l, \theta_l, S_{l+1}, A_{l+1})$$

be a σ-algebra and $M_l \doteq \sum_{t=0}^{l} \beta_t \epsilon_t^{(1)}$. It is easy to see that M_l is adapted to $\mathcal{F}_l$. Due to Lemma C.10 and boundedness of $\hat{v}_\theta$, we have

$$\sup_t \left| \epsilon_t^{(1)} \right| < \infty,$$

implying $\mathbb{E}\left[|M_l|\right] < \infty$ holds for any fixed $l < \infty$. Moreover,

$$\begin{aligned}
\mathbb{E}[M_{l+1}|\mathcal{F}_l] &= M_l + \mathbb{E}_{\theta_{l+1}, S_{l+2}, A_{l+2}}[\beta_{l+1}\epsilon_{l+1}^{(1)}|\mathcal{F}_l] \\
&= M_l + \beta_{l+1}\mathbb{E}_{\theta_{l+1}}\left[\mathbb{E}_{S_{l+2}, A_{l+2}}[\epsilon_{l+1}^{(1)}|\theta_{l+1}, \mathcal{F}_l]\right] \\
&= M_l + \beta_{l+1}\mathbb{E}_{\theta_{l+1}}[0] = M_l
\end{aligned}$$

$\{M_l\}$ is, therefore, a Martingale sequence. We now verify that M_l has bounded second moments, then $\{M_l\}$ converges according to Lemma C.5. For any $t_1 < t_2$, we have

$$\mathbb{E}[\epsilon_{t_1}^{(1)}\epsilon_{t_2}^{(1)}] = \mathbb{E}\left[\mathbb{E}[\epsilon_{t_1}^{(1)}\epsilon_{t_2}^{(1)}|\mathcal{F}_{t_2-1}]\right] = \mathbb{E}\left[\epsilon_{t_1}^{(1)}\mathbb{E}[\epsilon_{t_2}^{(1)}|\mathcal{F}_{t_2-1}]\right] = \mathbb{E}\left[\epsilon_{t_1}^{(1)}0\right] = 0.$$

Consequently,

$$\forall l, \quad \mathbb{E}[|M_l|^2] = \mathbb{E}[\sum_{t=0}^{l} \beta_t^2 \left(\epsilon_t^{(1)}\right)^2] = \mathcal{O}\left(\sum_{t=0}^{\infty} \beta_t^2\right).$$

Therefore, $\{M_l\}$ and $\sum_t \beta_t \epsilon_t^{(1)}$ converges a.s..

(2) $\sum_{t=1}^{l} \beta_t \epsilon_t^{(2)} = \beta_0 \nabla J(\theta_0)^\top \hat{v}_{\theta_0}(S_1, A_1) - \beta_l \nabla J(\theta_l)^\top \hat{v}_{\theta_l}(S_{l+1}, A_{l+1})$. The rest follows from the boundedness of $\nabla J(\theta)$ and $\hat{v}_\theta(s, a)$ and $\lim_{l\to\infty} \beta_l = 0$.

(3) Notice that

$$\sum_{t=1}^{l}\left|\beta_t\epsilon_t^{(3)}\right| \leq \sum_{t=1}^{l}|\beta_t-\beta_{t-1}|\left|\nabla J(\theta_{t-1})^\top \hat{v}_{\theta_{t-1}}(S_t,A_t)\right|$$
$$=\mathcal{O}\left(\sum_{t=1}^{l}\beta_{t-1}-\beta_t\right)=\mathcal{O}\left(\beta_0\right).$$

It follows easily that $\sum_t \beta_t\epsilon_t^{(3)}$ converges absolutely, thus converges.

(4) Lemma C.10 implies $\nabla J(\theta)$ is bounded and Lipschitz continuous in θ, if we are able to show $\forall(s,a,t)$, there exists a constant C_0 such that

$$\left\|\hat{v}_{\theta_t}(S_t,A_t)-\hat{v}_{\theta_{t-1}}(S_t,A_t)\right\| \leq C_0\|\theta_t-\theta_{t-1}\|, \tag{B.52}$$

we will have for some constants C_1 and C_2,

$$\left|\epsilon_t^{(4)}\right| \leq C_1\|\theta_t-\theta_{t-1}\| = C_1\|\beta_t\Gamma_1(w_t)\Gamma_2(u_t)\Delta_t\| \leq \beta_t C_2.$$

Consequently,

$$\sum_{t=1}^{l}\left|\beta_t\epsilon_t^{(4)}\right| \leq C_2\sum_{t=1}^{l}\beta_t^2 < C_2\sum_{t=1}^{\infty}\beta_t^2 \quad a.s.$$

Thus $\sum_t \beta_t\epsilon_t^{(4)}$ converges. We now proceed to show (B.52) does hold. According to (B.51), it suffices to show $\forall(s,a,\theta,\theta')$, there exists a constant C_0 such that

$$\|g_\theta^*(s,a)-g_{\theta'}^*(s,a)\| \leq C_0\left\|\theta-\bar{\theta}\right\|.$$

According to Assumption 12.2, $\rho_\theta(s,a)\nabla\log\pi_\theta(a|s)$ is bounded and Lipschitz continuous in θ. It is easy to verify the function $d\to\Gamma_i(d)d$ is bounded and Lipschitz continuous in any compact set. According to the definition of $g_\theta^*(s,a)$ in (B.50) and the boundedness of w_θ^* and u_θ^*, it then suffices to show w_θ^* and u_θ^* are Lipschitz continuous in θ. This Lipschitz continuity follows easily from Lemma C.2, which then completes the proof. □

B.31 Proof of Proposition 12.4

Proof. We first decompose $b(\theta)$ as $b(\theta)=b_1(\theta)+b_2(\theta)$, where

$$\psi_\theta \doteq \rho_\theta(s,a)\nabla\log\pi_\theta(a|s),$$
$$b_1(\theta) \doteq \sum_s d_\mu(s)\left(m_{\pi_\theta,\mu}(s)-x(s)^\top w_{\theta,\eta}^*\right)\sum_a \mu(a|s)\psi_\theta(s,a)\left(\tilde{x}(s,a)^\top u_{\theta,\eta}^*\right)$$
$$b_2(\theta) \doteq \sum_s d_\mu(s)m_{\pi_\theta,\mu}(s)\sum_a \mu(a|s)\psi_\theta(s,a)\left(q_{\pi_\theta}(s,a)-\tilde{x}(s,a)^\top u_{\theta,\eta}^*\right)$$

The boundedness of $m_{\pi_\theta,\mu}(s)$ and $\psi_\theta(s,a)$ implies

$$\begin{aligned}\|b_2(\theta)\|_{d_\mu} =& \mathcal{O}\left(\left\|q_{\pi_\theta} - \tilde{X}u^*_{\theta,\eta}\right\|_{\tilde{d}_\mu}\right)\\ =& \mathcal{O}\left(\left\|q_{\pi_\theta} - \tilde{X}u^*_{\theta,0}\right\|_{\tilde{d}_\mu}\right) + \mathcal{O}\left(\left\|\tilde{X}u^*_{\theta,\eta} - \tilde{X}u^*_{\theta,0}\right\|_{\tilde{d}_\mu}\right).\end{aligned}$$

Theorem 2 in Kolter (2011) states

$$\left\|q_{\pi_\theta} - \tilde{X}u^*_{\theta,0}\right\|_{\tilde{d}_\mu} \leq \frac{1+\gamma\left\|\tilde{P}_{\pi_\theta}\right\|_{\tilde{d}_\mu}}{1-\gamma}\left\|q_{\pi_\theta} - \tilde{\Pi}q_{\pi_\theta}\right\|_{\tilde{d}_\mu}.$$

Moreover, similarly to (B.19), we have

$$\left\|\tilde{X}u^*_{\theta,\eta} - \tilde{X}u^*_{\theta,0}\right\|_{\tilde{d}_\mu} = \mathcal{O}(\eta),$$

where have used the extreme value theorem for obtaining the constants hidden by $\mathcal{O}(\cdot)$. The term $b_1(\theta)$ can be similarly bounded (cf. Proposition 4.3), which will then completes the proof.

□

B.32 Proof of Theorem 12.5

Proof. By the definition of $J_{\hat{\gamma}}$, we have

$$\frac{\partial J_{\hat{\gamma}}}{\partial \theta_i} = \sum_s \left(\frac{\partial d_{\hat{\gamma}}(s)}{\partial \theta_i} i(s) v_\pi(s) + d_{\hat{\gamma}}(s) i(s) \frac{\partial v_\pi(s)}{\partial \theta_i}\right).$$

Using the Bellman equation yields

$$\begin{aligned}\frac{\partial v_\pi(s)}{\partial \theta_i} =& \frac{\partial \sum_a \pi(a|s) q_\pi(s,a)}{\partial \theta_i}\\ =& \sum_a \frac{\partial \pi(a|s)}{\partial \theta_i} q_\pi(s,a) + \gamma \pi(a|s) \sum_{s'} p(s'|s,a) \frac{\partial v_\pi(s')}{\partial \theta_i},\end{aligned}$$

or in a matrix form

$$\frac{\partial v_\pi}{\partial \theta_i} = \hat{r} + \gamma P_\pi \frac{\partial v_\pi}{\partial \theta_i},$$

where $\hat{r} \in \mathbb{R}^{|\mathcal{S}|}$ is defined as

$$\hat{r}(s) \doteq \sum_a \frac{\partial \pi(a|s)}{\partial \theta_i} q_\pi(s,a).$$

We then have

$$\frac{\partial v_\pi}{\partial \theta_i} = (I - \gamma P_\pi)^{-1} \hat{r}.$$

Consequently,

$$\begin{aligned} &\sum_s d_{\hat{\gamma}}(s) i(s) \frac{\partial v_\pi(s)}{\partial \theta_i} \\ =& i^\top diag(d_{\hat{\gamma}})(I - \gamma P_\pi)^{-1} \hat{r} \\ =& \hat{r}^\top (I - \gamma P_\pi^\top)^{-1} diag(d_{\hat{\gamma}}) i \\ =& \hat{r}^\top D_\mu h \\ =& \sum_s d_\mu(s) h(s) \sum_a \frac{\partial \pi(a|s)}{\partial \theta_i} q_\pi(s,a). \end{aligned} \tag{B.53}$$

For $\hat{\gamma} < 1$, we can rewrite (B.38) as

$$d_{\hat{\gamma}} = (1 - \hat{\gamma}) d_\mu + \hat{\gamma} P_\pi^\top d_{\hat{\gamma}}$$

Taking derivatives in both sides yields

$$\frac{\partial d_{\hat{\gamma}}}{\partial \theta_i} = \hat{\gamma} \frac{\partial P_\pi^\top}{\partial \theta_i} d_{\hat{\gamma}} + \hat{\gamma} P_\pi^\top \frac{\partial d_{\hat{\gamma}}}{\partial \theta_i},$$

i.e.,

$$\frac{\partial d_{\hat{\gamma}}}{\partial \theta_i} = \hat{\gamma} (I - \hat{\gamma} P_\pi^\top)^{-1} \frac{\partial P_\pi^\top}{\partial \theta_i} d_{\hat{\gamma}} = D_\mu g_i$$

Consequently, we have

$$\sum_s \frac{\partial d_{\hat{\gamma}}(s)}{\partial \theta_i} i(s) v_\pi(s) = \sum_s d_\mu(s) i(s) v_\pi(s) g_i(s). \tag{B.54}$$

Combining (B.53) and (B.54) then completes the proof. □

B.33 Proof of Proposition 12.6

Proof. Similar to the proof of Lemma 5.1 in Section B.10, we can show that

$$\begin{aligned} &\lim_{t \to \infty} \mathbb{E}\left[F_t^{(1)} \mid S_t = s \right] \\ =& \left(D_\mu^{-1} (I - \gamma P_\pi^\top) D_\mu D_\mu^{-1} diag(d_{\hat{\gamma}}) i \right)(s) = h(s). \end{aligned}$$

It can be computed that

$$
\begin{aligned}
&\lim_{t\to\infty} \mathbb{E}\left[\tau_{\hat{\gamma}}(S_{t-1})\rho_{t-1}\frac{\partial \log \pi(A_{t-1}|S_{t-1})}{\partial \theta_i} \mid S_t = s\right] \\
=&\lim_{t\to\infty} \sum_{\bar{s},\bar{a}} \Pr(S_{t-1}=\bar{s}, A_{t-1}=\bar{a} \mid S_t = s)\tau_{\hat{\gamma}}(\bar{s})\frac{\pi(\bar{a}|\bar{s})}{\mu(\bar{a}|\bar{s})}\frac{\partial \log \pi(\bar{a}|\bar{s})}{\partial \theta_i} \\
=&\sum_{\bar{s},\bar{a}} \frac{d_\mu(\bar{s})\mu(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})}{d_\mu(s)}\frac{d_{\hat{\gamma}}(\bar{s})}{d_\mu(\bar{s})}\frac{\pi(\bar{a}|\bar{s})}{\mu(\bar{a}|\bar{s})}\frac{\partial \log \pi(\bar{a}|\bar{s})}{\partial \theta_i} \\
=&\sum_{\bar{s}} \frac{d_{\hat{\gamma}}(\bar{s})}{d_\mu(s)} \sum_{\bar{a}} \frac{\partial \pi(\bar{a}|\bar{s})}{\partial \theta_i} p(s|\bar{s},\bar{a}) \\
=&\left(D_\mu^{-1}\frac{\partial P_\pi^\top}{\partial \theta_i} d_{\hat{\gamma}}\right)(s).
\end{aligned}
$$

Let $F_{t,i}^{(2)}$ denote the i-th element of $F_t^{(2)}$. Similar to the proof of Lemma 5.1 in Section B.10, we can show that

$$
\begin{aligned}
&\lim_{t\to\infty} \mathbb{E}\left[F_{t,i}^{(2)} \mid S_t = s\right] \\
=&\left(D_\mu^{-1}(I-\hat{\gamma}P_\pi^\top)^{-1}D_\mu D_\mu^{-1}\frac{\partial P_\pi^\top}{\partial \theta_i} d_{\hat{\gamma}}\right)(s) = g_i(s).
\end{aligned}
$$

Let $Z_{t,i}$ be the i-th element of Z_t. We then have

$$
\begin{aligned}
&\lim_{t\to\infty} \mathbb{E}\left[Z_{t,i}\right] \\
=&\lim_{t\to\infty} \sum_{s} \Pr(S_t = s, A_t = a)\mathbb{E}\left[Z_{t,i} \mid S_t = s, A_t = a\right] \\
=&\sum_{s,a} d_\mu(s)\mu(a|s) \lim_{t\to\infty} \mathbb{E}\left[Z_{t,i} \mid S_t = s, A_t = a\right] \\
=&\sum_{s,a} d_\mu(s)\mu(a|s)\frac{\pi(a|s)}{\mu(a|s)}h(s)q_\pi(s,a)\frac{\partial \log \pi(a|s)}{\partial \theta_i} \\
&+\sum_{s,a} d_\mu(s)\mu(a|s)\hat{\gamma}i(s)v_\pi(s)g_i(s) \\
=&\sum_{s} d_\mu(s)h(s)\sum_{a} q_\pi(s,a)\frac{\partial \pi(a|s)}{\partial \theta_i} + \hat{\gamma}\sum_{s} d_\mu(s)i(s)v_\pi(s)g_i(s),
\end{aligned}
$$

which completes the proof. □

Appendix C

Auxiliary Lemmas

Lemma C.1. *Let $f_1(x)$ and $f_2(x)$ be two Lipschitz continuous functions with Lipschitz constants C_1 and C_2. If they are also bounded by U_1 and U_2, then their product $f_1(x)f_2(x)$ is also Lipschitz continuous with $C_1U_2 + C_2U_1$ being a Lipschitz constant.*

Proof.

$$\begin{aligned}\|f_1(x)f_2(x) - f_1(y)f_2(y)\| &\leq \|f_1(x)(f_2(x) - f_2(y))\| + \|f_2(y)(f_1(x) - f_1(y))\| \\ &\leq (U_1C_2 + U_2C_1)\|x - y\|.\end{aligned}$$

□

Lemma C.2. $\left\|Y_1^{-1} - Y_2^{-1}\right\| \leq \left\|Y_1^{-1}\right\|\|Y_1 - Y_2\|\left\|Y_2^{-1}\right\|$.

Proof.

$$\left\|Y_1^{-1} - Y_2^{-1}\right\| = \left\|Y_1^{-1}(Y_1 - Y_2)Y_2^{-1}\right\| \leq \left\|Y_1^{-1}\right\|\|Y_1 - Y_2\|\left\|Y_2^{-1}\right\|.$$

□

Lemma C.3. *Let $\Lambda_P \doteq \left\{P \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}\right\}$ be a set of transition matrices. Assume Λ_P is compact. For any $P \in \Lambda_P$, assume the Markov chain in $\mathcal{Y}$ induced by P is ergodic and use $d_P \in \mathbb{R}^{|\mathcal{Y}|}$ to denote the corresponding stationary distribution. Then there exists a constant C_0 such that $\forall P, P' \in \Lambda_P$,*

$$\|d_P - d_{P'}\| \leq C_0\|P - P'\|.$$

Proof. For any $P \in \Lambda_P$, by the definition of stationary distribution, we have

$$L(P)d_P = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

where

$$L(P) \doteq \begin{bmatrix} P^\top - I \\ 1^\top \end{bmatrix}.$$

The ergodicity of P and the Perron-Frobenius theorem for nonnegative irreducible matrices (see, e.g., Horn and Johnson (2012)) ensure that $L(P)$ has full column rank. Consequently, we have

$$d_P = \left(L(P)^\top L(P)\right)^{-1} L(P)^\top \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

It is easy to see that $L(P)$ is Lipschitz continuous in P and we have by the extreme value theorem

$$\sup_{P \in \Lambda_P} \|L(P)\| < \infty,$$
$$\sup_{P \in \Lambda_P} \left\| \left(L(P)^\top L(P)\right)^{-1} \right\| < \infty.$$

Lemmas C.1 and C.2 then asserts that $\left(L(P)^\top L(P)\right)^{-1}$ is Lipschitz continuous on Λ_P. Using Lemma C.2 again confirms the Lipschitz continuity of d_P on Λ_P. □

Lemma C.4. *(Corollary 6.1 in page 150 of Bertsekas and Tsitsiklis (2015))*
If $Y \in \mathbb{R}^K$ is a square nonnegative matrix and $\rho(Y) < 1$, then there exists some vector $w \in \mathbb{R}^K$ satisfying $w_i > 0$ such that $\|Y\|_\infty^w < 1$. Here $\rho(\cdot)$ is the spectral radius, w_i is the i-th element of w. For a vector y, its w-weighted maximum norm is

$$\|y\|_\infty^w \doteq \max_i \left| \frac{y_i}{w_i} \right|.$$

For a matrix Y,

$$\|Y\|_\infty^w \doteq \max_{y \neq 0} \frac{\|Yy\|_\infty^w}{\|y\|_\infty^w}$$

.

Lemma C.5. *(Proposition 4.3 in Bertsekas and Tsitsiklis 1996) Assuming $\{M_l\}_{l=1,\ldots}$ is a Martingale sequence and there exists a constant $C_0 < \infty$ such that $\forall l$,*

$$\mathbb{E}\left[|M_l|^2\right] \leq C_0,$$

then $\{M_l\}$ converges almost surely.

Lemma C.6. *If A is invertible,*

$$\det\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right) = \det(A)\det\big(D - CA^{-1}B\big).$$

If D is invertible,

$$\det\left(\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right) = \det(D)\det\big(A - BD^{-1}C\big).$$

Proof. This is a standard result in linear algebra. □

Lemma C.7. *The $w^*(\theta)$ defined in (B.20) is Lipschitz continuous in θ.*

Proof. Recall

$$w^*(\theta) \doteq \bar{G}(\theta)^{-1}\bar{h}(\theta) = (X^\top D_{\mu_\theta} X + \eta I)^{-1} X^\top D_{\mu_\theta}(r + \gamma P_{\pi_{\bar{\theta}}} X\bar{\theta}).$$

We first show $P_{\pi_\theta} X\theta$ is Lipschitz continuous in θ. By definition of π_θ,

$$\begin{aligned}(P_{\pi_\theta} X\theta)(s,a) &= \sum_{s'} p(s'|s,a) \max_{a'} x(s',a')^\top \theta \\ &= \sum_{s'} p(s'|s,a) \|X_{s'}\theta\|_\infty,\end{aligned}$$

where each row of $X_{s'} \in \mathbb{R}^{|\mathcal{A}|\times K}$ is $x(s',a')^\top$. Then

$$\begin{aligned}&\Big|(P_{\pi_{\theta_1}} X\theta_1)(s,a) - (P_{\pi_{\theta_2}} X\theta_2)(s,a)\Big| \\ =&\left|\sum_{s'} p(s'|s,a)(\|X_{s'}\theta_1\|_\infty - \|X_{s'}\theta_2\|_\infty)\right| \\ \le&\sum_{s'} p(s'|s,a)\big|\|X_{s'}\theta_1\|_\infty - \|X_{s'}\theta_2\|_\infty\big| \\ \le&\sum_{s'} p(s'|s,a)\|X_{s'}\theta_1 - X_{s'}\theta_2\|_\infty \quad \text{(Triangle inequality)} \\ \le&\sum_{s'} p(s'|s,a)\|X_{s'}\|_\infty\|\theta_1 - \theta_2\|_\infty \\ \le&\|X\|_\infty\|\theta_1 - \theta_2\|_\infty.\end{aligned}$$

Consequently,

$$\Big\|P_{\pi_{\theta_1}} X\theta_1 - P_{\pi_{\theta_2}} X\theta_2\Big\|_\infty \le \|X\|_\infty\|\theta_1 - \theta_2\|_\infty.$$

The equivalence between norms then asserts that there exists a constant $L_0 > 0$ such that

$$\left\| P_{\pi_{\theta_1}} X\theta_1 - P_{\pi_{\theta_2}} X\theta_2 \right\| \leq L_0 \|X\| \|\theta_1 - \theta_2\|.$$

It is easy to see $L_0\|X\|$ is also a Lipschitz constant of $P_{\pi_{\bar{\theta}}} X\bar{\theta}$ by the property of projection.

Lemma C.3 ensures that D_{μ_θ} is Lipschitz continuous in θ and we use L_D to denote a Lipschitz constant. We remark that if we assume $\|X\| \leq 1$, we can indeed select an L_D that is independent of X. To see this, let L_μ be the Lipschitz constant in Assumption 9.2, then we have

$$\begin{aligned} |\mu_\theta(a|s) - \mu_{\theta'}(a|s)| \leq & L_\mu \|X_s\theta - X_s\theta'\|_\infty \leq L_\mu \|X_s\|_\infty \|\theta - \theta'\|_\infty \\ \leq & L_\mu \|X\|_\infty \|\theta - \theta'\|_\infty. \end{aligned}$$

Due to the equivalence between norms, there exists a constant $L'_\mu > 0$ such that

$$|\mu_\theta(a|s) - \mu_{\theta'}(a|s)| \leq L'_\mu \|X\| \|\theta - \theta'\| \leq L'_\mu \|\theta - \theta'\|. \tag{C.1}$$

We can now use Lemmas C.1 and C.2 to compute the bounds and Lipschitz constants for several terms of interest, which are detailed in Table C.1. From Table C.1 and Lemma C.1, a Lipschitz constant of $w^*(\theta)$ is

$$\begin{aligned} C_w \doteq & \left(\eta^{-1}\|X\|\|r\|L_D + \eta^{-2}\|X\|^3\|r\|L_D\right) + \eta^{-1}\|X\|\gamma L_0\|X\| \\ & + \gamma U_P \|X\| R_{B_1}(\eta^{-1}\|X\|L_D + \eta^{-2}\|X\|^3 L_D), \end{aligned}$$

which completes the proof.

	Bound	Lipschitz constant
D_{μ_θ}	1	L_D
$(X^\top D_{\mu_\theta} X + \eta I)^{-1}$	η^{-1}	$\eta^{-2}\|X\|^2 L_D$
$X^\top D_{\mu_\theta} r$	$\|X\|\|r\|$	$\|X\|\|r\|L_D$
$(X^\top D_{\mu_\theta} X + \eta I)^{-1} X^\top D_{\mu_\theta} r$	$\eta^{-1}\|X\|\|r\|$	$\eta^{-1}\|X\|\|r\|L_D + \eta^{-2}\|X\|^3\|r\|L_D$
$(X^\top D_{\mu_\theta} X + \eta I)^{-1} X^\top D_{\mu_\theta}$	$\eta^{-1}\|X\|$	$\eta^{-1}\|X\|L_D + \eta^{-2}\|X\|^3 L_D$
$\gamma P_{\pi_{\bar{\theta}}} X^\top \theta$	$\gamma U_P \|X\| R_{B_1}$	$\gamma L_0 \|X\|$

Table C.1: $U_P \doteq \sup_\theta \|P_{\pi_\theta}\|$.

□

Lemma C.8. *If $\|X\| \leq 1$, then $w^*(\theta)$ defined in (B.26) satisfies*

$$\|w^*(\theta_1) - w^*(\theta_2)\| \leq \|X\| L_w \|\theta_1 - \theta_2\|,$$

where L_w is a positive constant that depends on X through only $\frac{X}{\|X\|}$.

Proof. We first recall that if $\|X\| \leq 1$, the Lipschitz constants for both μ_θ and π_θ in θ can be selected to be independent of X (cf. (C.1)). Recall

$$w^*(\theta) = -(\eta I + A_{\pi_\theta,\mu_\theta}^\top C_{\mu_\theta}^{-1} A_{\pi_\theta,\mu_\theta})^{-1} A_{\pi_\theta,\mu_\theta}^\top C_{\mu_\theta}^{-1} X^\top D_{\mu_\theta} r.$$

Let

$$\tilde{X} \doteq \frac{X}{\|X\|}, \tilde{A}_{\pi_\theta,\mu_\theta} = \tilde{X}^\top D_{\mu_\theta}(\gamma P_{\pi_\theta} - I)\tilde{X}, \tilde{C}_{\mu_\theta} = \tilde{X}^\top D_{\mu_\theta} \tilde{X},$$

then

$$w^*(\theta) = -\|X\|(\eta I + \|X\|^2 \tilde{A}_{\pi_\theta,\mu_\theta}^\top \tilde{C}_{\mu_\theta}^{-1} \tilde{A}_{\pi_\theta,\mu_\theta})^{-1} \tilde{A}_{\pi_\theta,\mu_\theta}^\top \tilde{C}_{\mu_\theta}^{-1} \tilde{X} D_{\mu_\theta} r.$$

We now show $w^*(\theta)/\|X\|$ is Lipschitz continuous in θ by invoking Lemmas C.1 and C.2. Let $D_P \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ be a diagonal matrix whose diagonal entry is the stationary distribution of the chain induced by P. For any $P \in \Lambda_\mu$, Assumption 9.1 ensures that D_P is positive definite. Consequently, $\left\|(\tilde{X}^\top D_P \tilde{X})^{-1}\right\|$ is well defined and is continuous in P, implying it obtains its maximum in the compact set Λ_μ, say U_g. So $\left\|\tilde{C}_{\mu_\theta}^{-1}\right\| \leq U_g$ and importantly, U_g depends on X through only $\frac{X}{\|X\|}$. Using Lemma C.1, it is easy to see the bound and the Lipschitz constant of $\tilde{A}_{\pi_\theta,\mu_\theta}^\top \tilde{C}_{\mu_\theta}^{-1} \tilde{A}_{\pi_\theta,\mu_\theta}$ depend on X through only $\frac{X}{\|X\|}$. It is easy to see $\left\|(\eta I + \|X\|^2 \tilde{A}_{\pi_\theta,\mu_\theta}^\top \tilde{C}_{\mu_\theta}^{-1} \tilde{A}_{\pi_\theta,\mu_\theta})^{-1}\right\| \leq 1/\eta$. If we further assume $\|X\| \leq 1$, Lemma C.2 then implies that $(\eta I + \|X\|^2 \tilde{A}_{\pi_\theta,\mu_\theta}^\top \tilde{C}_{\mu_\theta}^{-1} \tilde{A}_{\pi_\theta,\mu_\theta})^{-1}$ has a Lipschitz constant that depends on X through only $\frac{X}{\|X\|}$. It is then easy to see there exists a constant $L_w > 0$, which depends on X only through $\frac{X}{\|X\|}$, such that

$$\|w^*(\theta_1) - w^*(\theta_2)\| \leq L_w \|X\| \|\theta_1 - \theta_2\|,$$

which completes the proof. □

Lemma C.9. *The $u^*(\theta)$ defined in (B.28) is Lipschitz continuous in θ.*

Proof. Recall

$$\begin{aligned}
\bar{h}(\theta) &\doteq \mathbb{E}_{(s,a)\sim d_{\mu_\theta}(\cdot), s'\sim p(\cdot|s,a)}[h_\theta(s,a,s')] = \bar{h}_1(\theta) + \bar{H}_2(\theta), \\
\bar{h}_1(\theta) &\doteq \begin{bmatrix} d_{\mu_\theta}^\top r \\ X^\top D_{\mu_\theta} r \end{bmatrix}, \bar{H}_2(\theta) \doteq \begin{bmatrix} 0 & d_{\mu_\theta}^\top (P_{\pi_{\bar{\theta}^w}} - I) X \bar{\theta}^w \\ -(X^\top d_{\mu_\theta})\bar{\theta}^r & X^\top D_{\mu_\theta} P_{\pi_{\bar{\theta}^w}} X \bar{\theta}^w \end{bmatrix}, \\
\bar{G}(\theta) &\doteq \mathbb{E}_{(s,a)\sim d_{\mu_\theta}(\cdot), s'\sim p(\cdot|s,a)}[G_\theta(s,a,s')] = \begin{bmatrix} 1 & 0^\top \\ 0 & X^\top D_{\mu_\theta} X + \eta I \end{bmatrix}, \\
u^*(\theta) &\doteq \bar{G}(\theta)^{-1} \bar{h}(\theta).
\end{aligned}$$

We now use Lemmas C.1 and C.2 to compute the bounds and Lipschitz constants for several terms of interest, which are detailed in Table C.2. From Table C.2 and Lemma C.1, a Lipschitz constant of $u^*(\theta)$ is

$$C_u = \max\left\{1, \eta^{-1}\right\}(\mathcal{O}(\|X\|) + \mathcal{O}(L_\mu)) + \max\left\{1, \eta^{-2}\right\}\mathcal{O}(\|X\|)$$

which completes the proof.

	Bound	Lipschitz constant
D_{μ_θ}	1	L_D
$\bar{G}(\theta)^{-1}$	$\max\{1, \eta^{-1}\}$	$\max\{1, \eta^{-2}\}\mathcal{O}(\|X\|^2)$
$h_1(\theta)$	$\mathcal{O}(\|X\|) + \mathcal{O}(1)$	$\mathcal{O}(L_\mu)$
$\bar{H}_2(\theta)$	$\mathcal{O}(\|X\|)$	$\mathcal{O}(\|X\|)$
$\bar{h}(\theta)$	$\mathcal{O}(\|X\|) + \mathcal{O}(1)$	$\mathcal{O}(\|X\|) + \mathcal{O}(L_\mu)$

Table C.2: Bounds and Lipschitz constants of several terms, assuming $\|X\| \leq 1, L_\mu \leq 1$.

□

Lemma C.10. *Let Assumptions 2.8 and 12.2 hold. Then there exists a constant $C_1 < \infty$ such that $\forall \theta, \theta'$,*

$$\begin{aligned}
\|\nabla J_{\pi_\theta,\mu}\| \leq & C_1, \\
\left\|\nabla J_{\pi_\theta,\mu} - \nabla J_{\pi_{\theta'},\mu}\right\| \leq & C_1\|\theta - \theta'\|, \\
\left|\frac{\partial^2 J_{\pi_\theta,\mu}}{\partial\theta_i \partial\theta_j}\right| \leq & C_1.
\end{aligned}$$

Proof. Recall

$$\begin{aligned}
\nabla_\theta J_{\pi_\theta,\mu} = & \sum_{s,a} d_\mu(s) m_{\pi_\theta,\mu}(s) q_{\pi_\theta}(s,a) \nabla_\theta \pi_\theta(a|s), \qquad \text{(C.2)} \\
m_{\pi_\theta,\mu} = & D_\mu^{-1}(I - \gamma P_{\pi_\theta}^\top)^{-1} D_\mu i, \\
q_{\pi_\theta} = & (I - \gamma \tilde{P}_{\pi_\theta})^{-1} r.
\end{aligned}$$

Since the space of all policies are compact, $\sup_\theta \|\nabla_\theta \pi(a|s)\| < \infty$ (Assumption 12.2), using the extreme value theorem yields

$$\sup_\theta \|\nabla J_{\pi_\theta,\mu}\| < \infty.$$

We now proceed to showing the boundedness of the Hessian of $J_{\pi_\theta,\mu}$. We first show $\left\|\frac{\partial q_\pi}{\partial \theta_i}\right\|$ is bounded. To see this, recall that

$$q_\pi = r + \gamma \tilde{P}_\pi q_\pi.$$

Taking derivatives in both sides yields

$$\frac{\partial q_\pi}{\partial \theta_i} = \gamma \frac{\partial \tilde{P}_\pi}{\partial \theta_i} q_\pi + \gamma \tilde{P}_\pi \frac{\partial q_\pi}{\partial \theta_i}.$$

Consequently, we have

$$\frac{\partial q_\pi}{\partial \theta_i} = \gamma (I - \gamma \tilde{P}_\pi)^{-1} \frac{\partial \tilde{P}_\pi}{\partial \theta_i} q_\pi.$$

The extreme value theorem and Assumption 12.2 then ensure that

$$\sup_\theta \left\| \frac{\partial q_\pi}{\partial \theta_i} \right\| < \infty.$$

We can similarly show that

$$\sup_\theta \left\| \frac{\partial m_{\pi,\mu}}{\partial \theta_i} \right\| < \infty.$$

Taking gradients w.r.t. θ in both sides of (C.2) and using the product rule of calculus then yields

$$\sup_\theta \left| \frac{\partial^2 J_{\pi,\mu}}{\partial \theta_i \partial \theta_j} \right| < \infty.$$

This bounded Hessian of $J_{\pi,\mu}$ immediately implies the Lipschitz continuity of $\nabla_\theta J_{\pi,\mu}$, which completes the proof. □

Lemma C.11. $\|P_\pi\|_{d_\mu} = \left\|D_\mu^{-1} P_\pi^\top D_\mu\right\|_{d_\mu}$

Proof. This proof is inspired by Kolter (2011). For any Y, we have

$$\begin{aligned}
\|Y\|_{d_\mu} &= \sup_{\|x\|_{d_\mu}=1} \|Yx\|_{d_\mu} = \sup_{\|x\|_{d_\mu}=1} \sqrt{x^\top Y^\top D_\mu Y x} \\
&= \sup_{\|y\|=1} \sqrt{y^\top D_\mu^{-\frac{1}{2}} Y^\top D_\mu Y D_\mu^{-\frac{1}{2}} y} = \left\| D_\mu^{\frac{1}{2}} Y D_\mu^{-\frac{1}{2}} \right\|.
\end{aligned}$$

Letting $Y = P_\pi$ yields

$$\|P_\pi\|_{d_\mu} = \left\| D_\mu^{\frac{1}{2}} P_\pi D_\mu^{-\frac{1}{2}} \right\|. \tag{C.3}$$

Letting $Y = D_\mu^{-1} P_\pi^\top D_\mu$ yields

$$\left\| D_\mu^{-1} P_\pi^\top D_\mu \right\| = \left\| D_\mu^{-\frac{1}{2}} P_\pi^\top D_\mu^{\frac{1}{2}} \right\|.$$

The rest follows from the well-known fact that ℓ_2 matrix norm is invariant under matrix transpose. □