Reinforcement Learning- Results in Approximate Dynamic Programming and Stochastic Approximation

A Thesis

Submitted for the Degree of

Doctor of Philosophy

in the **Faculty of Engineering**

by

Chandrashekar Lakshminarayanan

Computer Science and Automation
INDIAN INSTITUTE OF SCIENCE
BANGALORE – 560 012, INDIA

JUNE 2015

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Markov Decision Processes

Markov Decision Process (MDP) is an useful framework to cast problems involving optimal sequential decision making under uncertainity. Such problems often occur in science, engineering and economics, with planning by autonomous agents, control of large queuing systems and inventory control being some specific examples. Informally, the decision maker/planner wishes to maximize its objective, and in order to achieve this end, needs to perform an appropriate *action/control* at each decision epoch based on the *state* of the underlying *system*. The sequence of actions/control is also known as *policy* and the planner is interested in computing the *optimal* policy. In most cases, in order to compute the optimal policy, the planner also needs to *predict* the value of each state of any given control. Thus, with any given MDP, there are two natural sub-problems namely the problem of prediction and the problem of control.

Dynamic Programming (DP) is a general principle to solve MDPs, the core of which constitutes the *Bellman* equation (BE). The Bellman equation relates the optimal policy to its corresponding value known as the optimal *value function*, i.e., it relates the prediction and control problems of a given MDP. A host of analytical and iterative methods exists to solve the Bellman equation for MDPs. These methods are known as exact dynamic programming methods since they spell out the optimal control and its value

function exactly.

The three important exact DP methods are

- Value Iteration.
- Policy Iteration.
- Linear Programming.

Of these methods, value iteration and linear programming are termed as value function based DP methods since they compute the optimal value function first and the optimal policy is obtained by substituting it in the Bellman Equation. On the other hand, policy iteration predicts the value of a policy at each iterative step and then improves it to yeild a better policy in the next step eventually converging to the optimal policy. It is important to emphasize that these exact DP methods solve both the control and prediction problems.

1.2 Practical Issues

MDPs arising in practical problems are faced with two important issues. First issue is due to the *curse-of-dimensionality* (or simply the *curse*), a term which signifies the fact that the number of states of the MDP grows exponentially in the number of state variables. As the number of state variables increase, it becomes difficult to solve the MDP employing exact DP methods due to the computational overhead involved. Moreover, it also difficult to store and retrieve the exact value function and policy when the number of states are large.

The second issue is the lack of model information, wherein, the the model parameters of the MDP are not available explicitly. However, the the underlying system can be simulator or sample trajectories via direct interaction with the system. This scenario is known as the *reinforcement learning* setting since the model parameters have to be *learned* by using the feedback obtained via direct interaction with the environment.

A wide range of methods/algorithms exist in literature to address the issues of the curse and the lack of model information. In particular, the methods that tackle the curse

are broadly termed as approximate dynamic programming methods and methods that handle the case of lack of model information are called reinforcement learning algorithms.

1.3 Approximate Dynamic Programming

Approximate Dynamic Programming (ADP) is the name given to a class of approximate solution methods that tackle the curse. A major bottleneck the ADP methods face in the case of MDPs having a large number of states is that the value corresponding to each and every state cannot be stored, retrieved and computed. Most often ADP methods adopt an approximation architecture that enables compact representation of the value function in higher dimension so as to ease the storage and computation. Once the approximation architecture is fixed it is also important to devise a scheme that will choose the right candidate function that approximates the value function well. Thus central to any ADP method are

1. Approximation architecture.
2. Scheme to compute the right function within the chosen approximation architecture.

In most cases, ADP methods are obtained by combining exact DP methods with a given approximation architecture. A given ADP scheme can be said to belong to either of two approaches based on the way the prediction and the control problems are addressed. The two distinct approaches are:

1. The *value function* based approach, wherein a direct approximation to the optimal value function is obtained and a *greedy/sub-optimal* policy is computed by substituing the approximate value function in the Bellman equation.
2. The *approximate policy iteration* based approach, wherein, the critic evaluates the current policy, i.e., computes an approximation of the value function of the current policy. The actor on the other hand, makes use of the approximate value function to imporve the current policy.

Further, the performance of a given ADP method can be quantified by the following metrics namely

- Prediction Error, i.e., the difference between the exact value function and the approximate value function.
- Control Error, i.e., the loss in performance due to the sub-optimal policy as compared to the optimal policy.

It is a known result that if the prediction error is bounded in the max-norm (or L_∞-norm), the control error can be bounded as well. The necessacity of L_∞-norm arises due to the fact that the Bellman operator is only a contraction map in the L_∞ norm.

An ADP method is said to address both the prediction and control problems if it can offer guaranteed performance levels measured in terms of the aforementioned metrics. Given an ADP method, it a major theoretical challenge to come up with an analytical expression to bound the aforementioned performance metrics and it is in particlar difficult to bound the control error in many ADP schemes.

1.4 Linear Function Approximation

The first step in any ADP method is the choice of approximation architecture. In particular, a compact way of representing an approximation for the value function is necessary in most cases. A paramtererized function class is an useful approximation architecture since every function can be specified by the parameter it is enough to store only the parameters. Computing the right function that best approximates the exact value function then boils down to computations involving only the parameters and the curse can be tackled by choosing the number of parameters to be much lesser than the number of states. The most widely used parameterized class is the linear function approximator (LFA). Under a LFA, each function is written as a linear combination of pre-selected *basis* functions. The exact value function is then approximated by computing/learning the weights of the various basis functions in the linear combination.

The method used to compute the right weights of the linear combination affects the quality of the approximation. The projected Bellman equation and the approximate linear programming are two different ways of computing the weights. These two differing approaches have their advantages/disadvantages and provide interesting research problems.

1.4.1 Projected Bellman Equation

Once the basis functions of the LFA have been fixed, the right candidate function from the LFA has to be picked as the approximate value function. A candidate for the approximate value function could be that function which is at the least distance from the exact value function. Minimizing the distance between the linear combination of basis functions and the exact value function is similar to the idea of conventional *linear regression* wherein, the target function is *projected* onto the linear sub-space spanned by the basis functions. However, the idea of linear regression cannot be applied in a straightforward manner to compute the approximate value function because the exact value function (which is the target function in this case) is not known.

The Projected Bellman Equation (PBE) combines the idea of linear least squares projection and the Bellman equation. The central idea underlying the PBE is to find a fixed point in the linear sub-space spanned by the basis functions. However, existence of such a fixed point can be ensured only under a restricted setting. In particular, the Bellam operator is *non-linear* and is a contraction map in the max-norm (L_∞-norm). On the other hand, the linear least squares projection operator is non-expansive only in the L_2-norm. Hence the Bellman operator cannot be combined as such with the least squares projection operator. Neverthless, this issue can be side stepped by considering the Bellman operator restricted to a given policy which can be shown to be a contraction map in a generalized L_2-norm. The Bellman operator restricted to a given policy can be combined with the linear least squares projection operator to find a fixed point. However, such a fixed point can approximate only the value function of the particular policy (with respect to which the Bellman operator has been restricted) and not the optimal value

function. As a consequence, the PBE can only be used for approximate policy evaluation, i.e., it can be used to compute an approximation to the value function of a given policy. This means that the PBE based methods fall into the category of approximate policy iteration approach, i.e., the approximate policy evaluation should be followed by a policy improvement step. However, there is a 'norm-mismatch' between the projection operator that minimizes the error in L_2-norm and the max $/L_\infty$-norm required to guarantee policy improvement. As a result, the sub-optimal policy obtained from the approximate value function computed by the PBE is not guaranteed to be an improvement.

The major disadvantage of the PBE based methods is that they do not address the control problem completely. In fact, there are simple MDPs for which the PBE based solution methods are known to produce a sequence of policies that oscillate within a set of bad policies. This phenomenon is called as *policy-chattering* and is a direct consequence of the norm mismatch.

Problem 1: Projected Bellman Equation in the $(\min, +)$ linear basis

It is known that [**?**] the issue of norm-mismatch can be alleviated if the projection operator preserves *monotonicity*, i.e., if given two functions with one of them greater (component-wise) that the other, then the same holds for their projections as well. It is also known that the projection operator arising in the $(\min, +)$ linear algebra preserves this monotonicity property. The first part of the thesis deals with developing convergent ADP methods based on $(\min, +)$ linear algebra. In particular, the monotonicity property helps in

- Building convergent value function based ADP methods. Since the optimal value function is approximated directly, the issue of policy-chattering is absent.
- Providing performance guarantees for the greedy/sub-optimal policy. This is possible since there is no issue of 'norm-mismatch'.

1.4.2 Approximate Linear Programming

The approximate linear programming (ALP) formulation is obtained by introducing linear function approximation in the linear programming formulation (LP) of the given MDP. In

contrast to the PBE, the ALP does not rely on the linear least squares projection operator, but instead, computes the approximate value function by optimizing a linear objective over a set of linear inequality constraints. In particular, the ALP restricts its search to a space of linear functions that upper bound the optimal value function. The ALP is a value function method as it computes an approximation to the optimal value function. Since, the corresponding greedy policy can be derived in a straightforward manner, the ALP does not suffer from issues of policy-chattering. Further, the ALP also offers good performance guarantees for both the prediction as well as the control problems, and thus addresses both the problems.

A significant shortcoming of the ALP formulation is that the number of constraints is of the order of the state space. Most MDPs arising in practice have a large number of constraints and hence it is always not possible to solve the ALP with all the constraints. However, there are some special cases of factored MDPs with factored value function representations that enable elimination of variables to come up with a tractable number of constraints. A general approach is to handle the large number of constraints is constraint sampling, wherein a Reduced Linear Program (RLP) is formulated by sampling fewer number of constraints from the orginial constraints of the ALP. Though the RLP is known to perform well in experiments, the theoretical analysis is available only for a specific RLP formulated under idealized settings.

Problem 2: Framework to analyse constraint reduction in ALP:

The second part of the thesis deals with developing a novel theoretical framework to analyse constraint reduction/approximation in the ALP. The framework is built around studying a Generalized Reduced Linear Program (GRLP) whose constraints are obtained as positive linear combinations of the original constraints of the ALP. The salient features of the framework are

1. The analysis is based on two novel contraction maps and the error bounds are provided in a modified L_∞-norm. Both the prediction and control error are bounded, thus making the GRLP a complete ADP scheme.

2. Justification of linear function approximation of the Lagrange multipliers associated

with the constraints of the ALP. This is a desirable outcome, since both the primal and dual variables have linear function approximation.

1.5 Reinforcement Learning

Reinforcement Learning (RL) algorithms can be viewed as 'on-line'/sample trajectory based solution methods for solving MDPs. In the case of MDPs with a large number of states, RL algorithms are obtained as on-line versions of ADP methods. In order to handle the noise in the sample trajectory RL algorithms make use of stochastic approximation (SA). Typically, RL algorithm employing stochastic approximation are iterative schemes which take a small *step* towards the desired value at each iteration. By making the right choice of the *step-size* schedule effect of noise can be nullified and convergence to the desired value can be guaranteed.

Actor-Critic algorithms form an important sub-class of RL algorithms, wherein, the critic is responsible for policy evalutaion and the actor is responsible for policy improvement. In order to tackle the curse, the actor-critic schemes parameterize the policy and the value function. In an actor-critic algorithm, the critic forms the inner-loop and the actor constitutes the outer-loop. This is due to the fact that the actor has to needs the critic to evaluate a given policy before the actor updates the policy parameters. This effect of having two seperate loops can instead by achieved in practice by adopting different *step-size* schedules for the actor and the critic. Specifically, the step-sizes used by the actor updates have to be much smaller than those used in the critic updates.

Stochastic approximation schemes that use different step-size schedules for different sets of iterates are known as multi timescale stochastic approximation schemes. The conditions under which the iterates of a multi timescale SA schemes converge to the desired value have been studied in literature. One of the conditions required to ensure the convergence of the iterates of a multi timescale SA scheme is that the iterates need to be stable, i.e., they should be bounded. However, the conditions that *imply* the stability of the iterates in a multi timescale SA scheme have not been understood.

Problem 3: Stability Criterion for Two Timescale Stochastic Approximation Schemes:

The third part of the thesis deals with providing conditions under which the stability of iterates in a two timescale stochastic approximation scheme follows. Salient features of the contribution are as follows:

1. The analysis is based on the ODE approach to stochastic approximation.

2. Stability of iterates of important actor-critic algorithms follow for the result.

Chapter 2

Markov Decision Processes (MDPs)

2.1 Definition

An MDP is a 4-tuple $\langle S, A, P, g \rangle$, where S is the state space, A is the action space, P is the probability transition kernel and g is the reward function. The theoretical results developed in this thesis assume the setting as under.

Setting and Notation:

1. The MDP has finite number of states and actions.
2. Without the loss of generality, the state space is given by $S = \{1, 2, \ldots, n\}$ and the action space is given by $A = \{1, 2, \ldots, d\}$.
3. It is also assumed that all the actions are feasible in all the states.
4. The probability transition kernel P specifies the probability $p_a(s, s')$ of transitioning from state s to state s' under the action a.
5. The reward obtained for performing action $a \in A$ in state $s \in S$ is denoted by $g_a(s)$.

2.2 Inifinte Horizon Discounted Reward

By a policy, we mean a sequence $\pi = \{\pi_0, \ldots, \pi_n, \ldots\}$ of functions $\pi_i, i \geq 0$ that describe the manner in which an action is picked in a given state at time i. The most general class of policies is the class of *history-dependent randomized policies* denoted by π and is defined below.

Definition 1 *Let* $h_n = \{s_0, \ldots, s_n, \pi_0, \ldots, \pi_{n-1}\}$ *denote the* history. *A policy* π *is said to be a* history-dependent randomized *policy if* $\pi_n, n \geq 0$ *is such that for every* $s \in S$, $\pi_n(s, \cdot|h_n)$ *is a probability distribution over the set of actions.*

We denote the class of policies by Π. The infinite horizon discounted reward corresponding to state s under a policy π is denoted by $J_\pi(s)$ and is defined by

$$J_\pi(s) \stackrel{\Delta}{=} \mathbf{E}[\sum_{n=0}^{\infty} \alpha^n g_{a_n}(s_n)|\pi],$$

where $a_n \sim \pi_n(s, \cdot)$ and $\alpha \in (0, 1)$ is a given discount factor. Here $J_\pi(s)$ is known as the value of the state s under the policy π, and the vector quantity $J_\pi \stackrel{\Delta}{=} (J_\pi(s))_{s \in S} \in R^n$ is called the value-function corresponding to the policy π. The optimal value function J^* is defined as $J^*(s) \stackrel{def}{=} \max_{\pi \in \Pi} J_\pi(s)$.

A stationary deterministic policy (SDP) is one where $\pi_n \equiv u$ for all $n \geq 0$ for some $u \colon S \to A$. By abuse of notation we denote the SDP by u itself instead of π. In the setting that we consider, one can find an SDP that is optimal [**?** **?**], i.e., there exists an SDP u^* such that $J_{u^*}(s) = J^*(s), \forall s \in S$. In this paper, we restrict our focus to the class U of SDPs and without loss of generality we assume that there exists a unique SDP u^* that is optimal. Under a stationary policy u (or π), the MDP is a Markov chain and we denote its probability transition kernel by $P_u = (p_{u(i)}(i, j), i, j = 1, \ldots, n)$ (or $P_\pi = (p_{\pi(i)}(i, j), i, j = 1, \ldots, n)$, where $p_{\pi(i)}(i, j) = \sum_{a \in A} \pi(i, a) p_a(i, j)$ and $\pi(i) = (\pi(i, a), a \in A)$).

2.3 Bellman Equation

Given an MDP, our aim is to find the optimal value function J^* and the optimal policy SDP u^*. The optimal policy and value function obey the Bellman equation (BE): for all $s \in S$,

$$J^*(s) = \max_{a \in A} \big(g_a(s) + \alpha \sum_{s'} p_a(s, s') J^*(s')\big), \tag{2.1a}$$

$$u^*(s) = \arg\max_{a \in A} \big(g_a(s) + \alpha \sum_{s'} p_a(s, s') J^*(s')\big). \tag{2.1b}$$

If J^* is computed first, then u^* can be obtained by substituting J^* in (2.1b). The Bellman operator T is defined using the model parameters of the MDP as follows:

Definition 2 *The Bellman operator $T : \mathrm{R}^n \to \mathrm{R}^n$ is defined by*

$$(TJ)(s) = \max_{a \in A} \left(g_a(s) + \alpha \sum_{s'} p_a(s, s') J(s')\right), \tag{2.2}$$

where $J \in \mathrm{R}^n$ and $J_s = J(s)$, $s \in S = \{1, \ldots, n\}$. Similarly one can define the Bellman operator $T_u : \mathrm{R}^n \to \mathrm{R}^n$ restricted to an SDP u as follows:

$$(T_u J)(s) = g_{u(s)}(s) + \alpha \sum_{s'} p_{u(s)}(s, s') J(s')\,. \tag{2.3}$$

Given $J \in \mathrm{R}^n$, TJ is the 'one-step' greedy value function. It is also useful to define the notion of a one-step greedy policy as below:

Definition 3 *A policy $\tilde{u}$ is said to be greedy with respect to $\tilde{J}$ if*

$$\tilde{u}(s) = \arg\max_{a \in A} \big(g_a(s) + \alpha \sum_{s'} p_a(s, s') \tilde{J}(s')\big) \tag{2.4}$$

We also define the Bellman operator H for action values [**?**]:

Definition 4 *Let $H : \mathrm{R}^n \to \mathrm{R}^{nd}$ be defined as follows: For $J \in \mathrm{R}^n$,*

$$HJ = \begin{bmatrix} H_1 J \\ \vdots \\ H_d J \end{bmatrix} \in \mathrm{R}^{nd}, \textit{where}$$

$$(H_a J)(s) = g_a(s) + \alpha \sum_{s'} p_a(s, s') J(s'), \qquad s \in S, a \in A. \tag{2.5}$$

We now state without proof the important properties related to the Bellman operator. Though in these results, we make use of the Bellman operator T, the results also trivially hold for T_u as well.

2.4 Bellman Operator

Lemma 5 *T is a* max*-norm contraction operator, i.e., given $J_1, J_2 \in \mathbf{R}^n$,*

$$||TJ_1 - TJ_2||_\infty \leq \alpha ||J_1 - J_2||_\infty. \tag{2.6}$$

Corollary 1 *J^* is a unique fixed point of T, i.e., $J^* = TJ^*$.*

The Bellman operator T exhibits two more important properties presented in the following lemmas (see [**?**] for proofs):

Lemma 6 *T is a monotone map, i.e., given $J_1, J_2 \in \mathrm{R}^n$ such that $J_2 \geq J_1$, we have $TJ_2 \geq TJ_1$. Further, if $J \in \mathrm{R}^n$ is such that $J \geq TJ$, it follows that $J \geq J^*$.*

Lemma 7 *Given $J \in \mathrm{R}^n$, $t \in \mathrm{R}$ and $\mathbf{1} \in \mathrm{R}^n$, a vector with all entries 1, we have*

$$T(J + t\mathbf{1}) = TJ + \alpha t \mathbf{1}. \tag{2.7}$$

Note that Lemmas 5-13 also hold for the Q-Bellman operator H. Note that Lemmas 5-13 also hold for the Bellman operator H defined for the action values in Definition 4.

Solving an MDP involves handling two sub-problems namely the problem of *control* and the problem of *prediction*. The problem of *control* deals with coming up with a good

(and if possible the optimal) policy. Often, in order to solve the problem of *control*, one needs to solve the problem of *prediction*, which deals with computing the value function J_u of the policy u. The fact that the two problems are related is reflected in the Bellman equation in (2.1), where J^* from (2.1a) is used in (2.1b) to obtain the optimal policy u^*. Thus, the Bellman equation is at the heart of the solution methods to MDPs. Any solution method to MDP is said to be complete only if it satisfactorily (with provable performance guarantees) addresses both the prediction and the control problems.

The basic solution methods namely value iteration, policy iteration and linear programming (LP) formulation [**?**] solve both the control and prediction problems. Of the three basic methods, in this paper, we are interested in the LP formulation given by

$$\begin{aligned} &\min_{J \in \mathrm{R}^n} c^\top J \\ &\quad \text{s.t } J(s) \geq g_a(s) + \alpha \sum_{s'} p_a(s,s') J(s'), \qquad s \in S, a \in A, \end{aligned} \tag{2.8}$$

where $c \in \mathrm{R}^n_+$ is any vector whose components are all non-negative. One can show that J^* is the solution to the LP formulation (2.9) [**?**]. Also, of the three methods, value iteration and LP formulation are value function based methods, i.e., they compute J^* directly and then u^* is obtained by plugging J^* in (2.1b).

While the basic methods (i.e., VI, PI and LP) can be used to compute exact values J^* and u^* for MDPs with a small number of states, they are computationally expensive in the case of MDPs with a large number of states.

2.5 Basic Solution Methods

Algorithm 1 Value Iteration (VI)

1: Initialize J_0.
2: **for** $n = 0, 1, 2, \ldots$ **do**
3: $\quad J_{n+1} = TJ_n$.
4: **end for**

One can show that the iterates J_n in Algorithm 1 converge to J^*, i.e., $J_n \to J^*$ as $n \to \infty$.

Algorithm 2 Policy Iteration (PI)

1: Initialize a policy u_0.
2: **for** $n = 0, 1, 2, \ldots$ **do**
3: Policy Evaluation Step: Compute J_{u_n} such that $J_{u_n} = T_{u_n} J_{u_n}$.
4: Policy Improvement Step: $u_{n+1}(s) = \arg\max_{a \in A}(g_a(s) + \sum_{s'} p_a(s, s') J_{u_n}(s'))$.
5: **end for**

The policy iteration algorithm (Algorithm 2) has two important steps namely *evaluation/prediction* and *improvement/control*. PI is different from VI and LP in that it computes a sequence of policies u_n and their corresponding value functions J_{u_n}. One can show that in Algorithm 2 as $n \to \infty$ $u_n \to u^*$ and $J_{u_{n+1}} \geq J_{u_n}, \forall n$.
The linear programming (LP) formulation can be obtained by unfurling the max operator in the Bellman equation into a set of linear inequalities and is given by:

$$\begin{aligned} &\min_{J \in \mathrm{R}^n} c^\top J \\ &\text{s.t } J(s) \geq g_a(s) + \alpha \sum_{s'} p_a(s, s') J(s'), \ \forall s \in S, a \in A, \end{aligned} \tag{2.9}$$

where $c \in \mathrm{R}^n_+$ is any vector whose components are all non-negative. One can show that J^* is the solution to the LP formulation (2.9) [**?**]. Also, note that the value iteration and linear programming formulations are value function based methods, i.e., they compute J^* directly and then u^* is obtained by plugging J^* in (2.1b).
While the basic methods (i.e., VI, PI and LP) can be used to compute exact values J^* and u^* for MDPs with a small number of states, they are computationally expensive in the case of MDPs with a large number of states.

2.6 Curse of Dimentionality

Curse-of-Dimensionality is a term used to denote the fact that the number of states grows exponentially in the number of state variables. Most MDPs arising in practical applications suffer from the curse, i.e., have large number of states and it is difficult to compute $J^*/J_u \in \mathrm{R}^n$ exactly in such scenarios. Approximate dynamic programming (ADP) [**? ? ? ?**] methods compute an approximate value function $\tilde{J}$ instead of J^*/J_u. In order to tackle the curse, ADP methods resort to dimensionality reduction by parameterizing the value function and/or the policy. Value-function based ADP methods form an important subclass of ADP methods whose brief overview is given below.

2.7 Value Function Based Approximate Dynamic Programming

In the value-function based ADP schemes a parameterized family is chosen and the approximate value-function $\tilde{J}$ is picked from the chosen parameterized class. Once the approximate value function $\tilde{J}$ is computed, a sub-optimal policy $\tilde{u}$ can be obtained as the one-step greedy policy with respect to $\tilde{J}$ by making use of (2.4). The following lemma characterizes the degree of sub-optimality of the greedy policy $\tilde{u}$.

Lemma 8 *Let $\tilde{J}$ be the approximate value function and $\tilde{u}$ be as in* (2.4)*, then*

$$||J^* - J_{\tilde{u}}||_\infty \leq \frac{2}{1-\alpha}||J^* - \tilde{J}||_\infty. \tag{2.10}$$

Proof: We know that

$$(T_{\tilde{u}})\tilde{J}(s) = g(s) + \alpha \sum_{s'} p_{\tilde{u}(s)}(s, s')\tilde{J}(s'), \tag{2.11}$$

$$J_{\tilde{u}}(s) = g(s) + \alpha \sum_{s'} p_{\tilde{u}(s)}(s, s')J_{\tilde{u}}(s'). \tag{2.12}$$

Hence we can write by subtracting (2.11) from (2.12)

$$\begin{aligned} J_{\tilde{u}} - \tilde{J} &= T_{\tilde{u}}\tilde{J} - \tilde{J} + \alpha P_{\tilde{u}}(J_{\tilde{u}} - \tilde{J}), \text{ or,} \\ J_{\tilde{u}} - \tilde{J} &= (I - \alpha P_{\tilde{u}})^{-1}(T_{\tilde{u}}\tilde{J} - \tilde{J}), \text{ hence} \\ ||J_{\tilde{u}} - \tilde{J}||_\infty &\leq \frac{1}{1-\alpha}||T_{\tilde{u}}\tilde{J} - \tilde{J}||_\infty. \end{aligned}$$

We know from (2.4) that $T_{\tilde{u}}\tilde{J} = T\tilde{J}$. Also from the fact that $J^* = TJ^*$ and the contraction property of T, we know $||T\tilde{J} - J^*||_\infty \leq \alpha||\tilde{J} - J^*||_\infty$ and $||T_{\tilde{u}}\tilde{J} - \tilde{J}||_\infty \leq (1+\alpha)||\tilde{J} - J^*||_\infty$. Hence we have

$$\begin{aligned} ||J_{\tilde{u}} - J^*||_\infty &= ||J_{\tilde{u}} - J^* + \tilde{J} - \tilde{J}||_\infty \\ &\leq ||J_{\tilde{u}} - \tilde{J}||_\infty + ||\tilde{J} - J^*||_\infty \\ &\leq \frac{1}{1-\alpha}||T_{\tilde{u}}\tilde{J} - \tilde{J}||_\infty + ||J^* - \tilde{J}||_\infty \\ &\leq \frac{1+\alpha}{1-\alpha}||J^* - \tilde{J}||_\infty + ||J^* - \tilde{J}||_\infty \\ &\leq \frac{2}{1-\alpha}||J^* - \tilde{J}||_\infty. \end{aligned}$$

The quality of any ADP method depends on the approximation guarantees it offers for the quantities $||J^* - \tilde{J}||$ and $||J^* - J_{\tilde{u}}||$, where $||\cdot||$ is an appropriate norm. The term $||J^* - \tilde{J}||$ denotes the error in prediction and $||J^* - J_{\tilde{u}}||$ denotes the loss in performance resulting from the sub-optimal policy $\tilde{u}$ with respect to the optimal policy u^*. Of the two error terms, $||J^* - J_{\tilde{u}}||$ is more important because ultimately we are interested in coming up with a useful policy. In the context of ADP methods, the control and prediction problems are said to be addressed when the error terms $||J^* - \tilde{J}||$ and $||J^* - J_{\tilde{u}}||$ are bounded by "small" constants.

2.8 Linear Function Approximation

The most widely used parameterized class to approximate the value-function is the linear function approximator (LFA). Under LFA, the value-function is approximated as $\tilde{J} = \Phi r^*$, with $\Phi = [\phi_1 | \dots | \phi_k]$ being an $n \times k$ feature matrix and r^* is the parameter to be learnt. There are two important approaches to value function approximation. Both the approaches start out with a basic solution method and appropriately introduce function approximation in it. The two approaches are:

1. The Projected Bellman Equation (PBE) which combines the BE and the linear least squares projection operator to project high dimensional quantities onto the subspace of the LFA.

2. The approximate linear programming formulation which introduces the LFA in the linear programming formulation.

Chapter 3

Approximate Dynamic Programming in $(\min, +)$ linear basis

A host of ADP methods are based on the PBE and have been found to be useful in practice. The main application of the PBE is for approximate policy evaluation, i.e., to compute $\tilde{J}_u$, an approximation to the value function J_u of policy u. Due to the mismatch in the norms, i.e, the linear least squares projection operator based on the L_2-norm and the L_∞-norm of the Bellman operator T, one cannot use the PBE to obtain a direct approximation to J^*. Thus in order to solve the problem of control, $\tilde{J}_u$ is used to compute a one-step greedy policy. However, again due to the mismatch in the norms, i.e., L_2-norm of the linear least squares projection and the L_∞ norm required for policy improvement (Lemma 8), the one-step greedy policy need not necessarily be an improvement. This leads to a phenomenon called *policy-chattering* [**?**] where looping within of bad policies can occur. Further, such policy-chattering can be demonstrated in simple examples as well [**?**]. Thus, though the approximate value function obtained by solving the PBE offers guarantees for prediction it does not offer any guarantees for the control problem, a significant shortcoming of the PBE based methods.

3.1 Projected Bellman Equation

A large class of ADP methods compute r^* to be the solution to the projected Bellman equation (PBE) given below:

$$\Phi r^* = \Phi(\Phi^\top D_u \Phi)^{-1}\Phi^\top T_u \Phi r^*, \tag{3.1}$$

where T_u is the Bellman operator corresponding to an SDP u and D_u is a $n \times n$ diagonal matrix whose diagonal entries are the stationary probabilities of the states under the SDP u. The PBE can be written in short as

$$\Phi r^* = \Pi T_u \Phi r^*, \tag{3.2}$$

where $\Pi = \Phi(\Phi^\top D_u \Phi)^{-1}\Phi^\top$ is the projection operator. Notice that approximate value function $\tilde{J}_c = \Phi r^*$ obtained by solving the PBE (3.2) is an approximation only to J_u and not J^*. This scenario is unavoidable because T_u cannot be replaced by T in (3.2), since ΠT cannot be guaranteed to be a contraction map. As a result, an equation of the form $\Phi r = \Pi T \Phi r$ need not have a solution. Thus, the PBE can only be used for approximate policy evaluation, i.e., to compute $\tilde{J}_u \approx J_u$. Nevertheless, the solution to the PBE offers error bounds for the prediction problem and is given below:

$$||J_u - \Phi r^*||_D \leq \frac{1}{\sqrt{1-\alpha^2}}||J_u - \Pi J_u||_D. \tag{3.3}$$

In order to address the problem of control, one resorts to approximate policy improvement (API) wherein a one-step look ahead policy is derived. The API scheme is given in Algorithm 3

Algorithm 3 Approximate Policy Iteration (API)

1: Start with any policy u_0
2: **for** $i = 0, 1, \ldots, n$ **do**
3: Approximate policy evaluation $\tilde{J}_i = \Phi r_i^*$, where $\Phi r_i^* = \Pi T_{u_i} \Phi r_i^*$.
4: Improve policy $u_{i+1}(s) = \arg\max_a (g_a(s) + \alpha \sum_{s'} p_a(s, s') \tilde{J}_i(s'))$.
5: **end for**
6: **return** u_n

The performance guarantee of API can be stated as follows:

Lemma 9 *If at each step i one can guarantee that $||\tilde{J}_i - J_{u_i}||_\infty \leq \delta$, then one can show that $\lim_{n\to\infty} ||J_{u_i} - J^*|| \leq \frac{2\delta\alpha}{(1-\alpha)^2}$.*

Note that the error bound required by Lemma 9 is in the L_∞ norm, whereas (3.3) is only in the L_2 norm. So API cannot guarantee an approximate policy improvement at each step which is a shortcoming. Also, even-though each evaluation step (line 3 of Algorithm 3) converges, the sequence $u_n, n \geq 0$ is not guaranteed to converge. This is known as policy *chattering*. Thus the problem of *control* is only partially addressed by API, i.e, suffers in both convergence and performance guarantees.

In short, whilst the approximate value function obtained by the solving the PBE offers guarantees for prediction it does not offer any guarantees for the control problem.

In the next section, we discuss the approximate linear programming (ALP) formulation, the basic results and present prior results in literature as well as motivate the problem that we address in this paper.

3.2 $(\min, +)$ linear functions

The $\mathbf{R}_{\min}$ semiring is obtained by replacing multiplication ($\times$) by $+$, and addition ($+$) by min.

Definition 10

$$\begin{aligned} &\textit{Addition:} && x \oplus y = \min(x, y). \\ &\textit{Multiplication:} && x \otimes y = x + y. \end{aligned}$$

Henceforth we use, $(+, \times)$ and $(\oplus, \otimes)$ to respectively denote the conventional and $\mathbf{R}_{\min}$ addition and multiplication respectively. A Semimodule over a semiring can be defined in a similar manner to vector spaces over fields. In particular we are interested in the semimodule $\mathcal{M} = \mathbf{R}^n_{\min}$ (since $J^* \in \mathbf{R}^n_{\min}$). Given $u, v \in \mathbf{R}^n_{\min}$, and $\lambda \in \mathbf{R}_{\min}$, we define addition and scalar multiplication as follows:

Definition 11

$$\begin{aligned} (u \oplus v)(i) &= \min\{u(i), v(i)\} = u(i) \oplus v(i), \forall i = 1, 2, \ldots, n. \\ (u \otimes \lambda)(i) &= u(i) \otimes \lambda = u(i) + \lambda, \forall i = 1, 2, \ldots, n. \end{aligned}$$

Similarly one can define the $\mathrm{R}_{\max}$ semiring which has the operators max as addition and $+$ as multiplication.

It is a well known fact that deterministic optimal control problems with cost/reward criterion are $(\min, +)/(\max, +)$ linear ([**? ? ? ? ?**]). However, it is straightforward to show that the Bellman operator T (as well as H) in (**??**) corresponding to infinite horizon discounted reward MDPs is neither $(\min, +)$ linear nor $(\max, +)$ linear. Nevertheless, the motivation behind developing ADP methods based on $(\min, +)$ LFAs is to explore them as an alternative to the conventional basis representation.

Given a set of basis functions $\{\phi_i, i = 1, \ldots, k\}$, we define the $(\min, +)$ linear span to be $\mathcal{V} = \{v | v = \Phi \otimes r \stackrel{def}{=} \min(\phi_1 + r(1), \ldots, \phi_k + r(k)), r \in \mathbf{R}^k_{\min}\}$. Thus $\mathcal{V}$ is a subsemimodule. In the context of value function approximation, we would want to project quantities in $\mathbf{R}^n_{\min}$ onto $\mathcal{V}$. The $(\min, +)$ projection operator Π_M works as follows ([**? ? ? ?**]):

$$\Pi_M u = \min\{v | v \in \mathcal{V}, v \geq u\}, \forall u \in \mathcal{M}. \tag{3.4}$$

We can write the PBE in the $(\min, +)$ basis:

$$\begin{aligned} v &= \Pi_M T v, v \in \mathcal{V} \\ \Phi \otimes r^* &= \min\{\Phi \otimes r^* \in \mathcal{V} | \Phi \otimes r^* \geq T\Phi \otimes r^*\}. \end{aligned} \tag{3.5}$$

Thus by making use of $(\min, +)$ LFAs and the projection operator Π_M we aim to find the minimum upper bound to J^*/Q^*.

3.3 Fenchel Dual and Projection on Subsemimodules

In this section, we demonstrate the connections between the *Fenchel-Legendre* transform (FLT) and the $(\min, +)$ projection defined in (3.4). Given a function $f\colon \mathrm{R}^n \to \mathrm{R}$, its FLT is defined by $f^*\colon \mathrm{R}^n \to \mathrm{R}$, with

$$f^*(y) = \sup_{x \in \mathrm{R}^n} (y^\top x - f(x)), y \in \mathrm{R}^n. \tag{3.6}$$

If f is convex, then it can be recovered as $f = f^{*^*}$, i.e.,

$$f(x) = f^{*^*}(x) = \sup_{y \in \mathrm{R}^n} (x^\top y - f^*(y)), x \in \mathrm{R}^n. \tag{3.7}$$

We can rewrite (3.6) as below

$$f^*(y) = \sup_{x \in \mathrm{R}^n} (f_y(x) - f(x)), y \in \mathrm{R}^n, \text{ where } f_y(x) = y^\top x. \tag{3.8}$$

Now instead of considering functions $f_y(x)$ indexed by $y \in \mathrm{R}^n$, we consider the sequence $\{\phi_j\}, j \in \mathcal{J} = \{1, 2, \ldots, k\}, \phi_j\colon \mathrm{R}^n \to \mathrm{R}$. Then (3.8) can be modified as below:

$$f^*(j) = \sup_{x \in \mathrm{R}^n} (\phi_j(x) - f(x)), j \in \mathcal{J}. \tag{3.9}$$

We call (3.9), the sup-Transform or the max-Transform. It is easy to check that $\phi_j(x) - f^*(j) < f(x), \forall x \in \mathrm{R}^n, j \in \mathcal{J}$. Since our index set in (3.9) is finite (as opposed to R^n

as in (3.6)), it is not necessary that the original function f can be *reconstructed* from $f^*(j), j \in \mathcal{J}$. However, we can get an approximation $\tilde{f}$ as below:

$$f(x) \approx \tilde{f}(x) = \sup_{j \in \mathcal{J}} (\phi_j(x) - f^*(j)). \tag{3.10}$$

In the light of (3.9) and (3.10), the projection in (3.4) is nothing but the min-Transform (as opposed to the max-Transform (3.9)). It is more clear if we rewrite (3.4) for the case when $\mathcal{V} = Span\{\phi_j | \phi_j \in \mathbf{R}^n_{\min}, j = 1, \ldots, k)$. Let $\Pi_M u = \Phi \otimes r^u$, then one can see that

$$\Pi_M u = \{\min \Phi \otimes r | \Phi \otimes r \geq u, r \in \mathbf{R}^k_{\min}\}. \tag{3.11}$$

$$r^u(j) = \min_{i=1,2,\ldots,n} (\phi_j(i) - u(i)), \forall j = 1, 2, \ldots, k. \tag{3.12}$$

Note the similarity between $r^u(j)$ in (3.12) and $f^*(j)$ in (3.9). Then the approximation/projection of u onto $\mathcal{V}$ is given by $\tilde{u} = \Pi_M u = \Phi \otimes r^u$ with

$$\begin{aligned} \Pi_M u(i) &= - \min_{j=1,\ldots,k} (\phi_j(i) + r^u(j)) \\ &= \phi_1(i) \otimes r^u(1) \oplus \ldots \oplus \phi_k(i) \otimes r^u(k). \end{aligned} \tag{3.13}$$

Also, it is important to note that (3.9) deals with projecting a function, while (3.4) deals with projecting the elements of n-dimensional semimodule. Nevertheless, the spirit of the projection is similar in both cases. Also, $\phi_j(i) + r^u_j - u(i) > 0$, i.e., the min-Transform approximates the given element u by point-wise minimum of functions that upper bound u. We end this section with the following illustration.

Example 1 *Let* $f(x) = x^2$, *and let* $a = (a(j), j = 1, \ldots, 5) = (-0.8, -0.4, 0, 0.4, 0.8)$, *and* $\phi_j(x) = 2|x - a(j)|$. *Then* $(\min, +)$ *LFA of* $f(x)$ *via the* min-*Transform using the* $\{\phi_j(x), j = 1, \ldots, 5\}$ *as the* $(\min, +)$ *basis, is given in the Figure* **??**.

3.4 Approximate Q Iteration

We now present an ADP scheme based on solving the PBE in $(\min, +)$ basis to compute $\tilde{Q}(\approx Q^*)$. Our ADP scheme successfully addresses the two shortcomings of the ADP scheme in conventional basis. First, we establish contraction of the projected Bellman operator in the L_∞ norm. This enables us to show that our recursion to compute $\tilde{Q}$ converges. We compute a greedy policy $\tilde{u}(s) = \max_a \tilde{Q}(s,a)$, and an approximate value function $\tilde{J}(s) = \max_a \tilde{Q}(s,a)$. Secondly, we also bound $||\tilde{Q} - Q^*||$ in the max norm which along with Lemma 8 gives a guarantee for $J_{\tilde{u}}$.
We are interested in solving the following PBE:

$$\Phi \otimes r^* = \Pi_M H \Phi \otimes r^*. \tag{3.14}$$

Since we want to approximate Q^*, Φ is an $nd \times k$ feature matrix, and $Q^*(s,a) \approx \tilde{Q}(s,a) = \phi^{(s-1)\times d + a} \otimes r^*$, where ϕ^i is the i^{th} row of Φ. The Approximate Q Iteration (AQI) algorithm is given by

$$\Phi \otimes r_{n+1} = \Pi_M H \Phi \otimes r_n. \tag{3.15}$$

The following results help us to establish the fact that the projected Bellman operator $\Pi_M H \colon \mathbf{R}_{\min}^{n\times d} \to \mathbf{R}_{\min}^{n\times d}$ is a contraction map in the L_∞ norm. In the discussion that follows, $\mathbf{1} \in \mathrm{R}^n \times d$ is a vector with all components equal to 1.

Lemma 12 *For $Q_1, Q_2 \in \mathbf{R}_{\min}^{n\times d}$, such that $Q_1 \geq Q_2$, we have $\Pi_M Q_1 \geq \Pi_M Q_2$.*

Proof: Follows from the definition of Π_M in (3.4).

Lemma 13 *Let $Q \in \mathbf{R}_{\min}^{n\times d}$, $V_1 = \Pi_M Q$ be its projection onto $\mathcal{V}$. For $k \in \mathrm{R}$ the projection of $Q + k\mathbf{1}$ is $V_2 = \Pi_M Q + k\mathbf{1}$.*

Proof: We know that since $V_1 \geq Q$, $V_1 + k\mathbf{1} \geq Q + k\mathbf{1}$, and from the definition of the Π_M in (3.4) $V_2 \leq V_1 + k\mathbf{1}$. Similarly $V_2 - k\mathbf{1} \geq Q$, so $V_1 \leq V_2 - k\mathbf{1}$.

Theorem 14 *The projected Bellman operator $\Pi_M H$ is a contraction map in L_∞ norm with modulus α.*

Proof: Let $Q_1, Q_2 \in \mathbf{R}_{\min}^{n\times d}$, and $\epsilon \stackrel{def}{=} ||Q_1 - Q_2||_\infty$, then by Lemma 12 we have

$$\begin{aligned}\Pi_M H Q_1 - \Pi_M H Q_2 &\leq \Pi_M H(Q_2 + \epsilon \mathbf{1}) - \Pi_M H Q_2 \\ &= \Pi_M (H Q_2 + \alpha\epsilon \mathbf{1}) - \Pi_M H Q_2 \\ &= \alpha\epsilon \mathbf{1}. \end{aligned} \tag{3.16}$$

Here the equality in second line is due to Lemma **??**, and the final line follows from Lemma 13. Similarly $\Pi_M H Q_2 - \Pi_M H Q_1 \leq \alpha\epsilon\mathbf{1}$, so it follows that $||\Pi_M H Q_1 - \Pi_M H Q_2||_\infty \leq \alpha||Q_1 - Q_2||_\infty$.

Corollary 2 *AQI in* (3.15) *converges to a fixed point r^* such that $\Phi \otimes r^* = \Pi_M H \Phi \otimes r^*$.*

3.5 Variational Approximate Q Iteration

The projection operator Π_M used in (3.15) is exact. Let $v = \Pi_M u$, then for test vectors $w_i \in \mathbf{R}_{\min}^n, i = 1, \ldots, m$ it follows from the definition of Π_M that

$$w_i^\top v \geq w_i^\top u$$

where the dot product $x^\top y = \min_{i=1}^n (x(i) + y(i))$. Let W denote the $nd \times m$ test matrix whose columns are w_i. Now we shall define the approximate projection operator to be

$$\Pi_M^W u = \min\{v \in \mathcal{V} | W^\top v \geq W^\top u\}. \tag{3.17}$$

The superscript in Π_M^W denotes the test matrix W. The Variational Approximate Q Iteration (VAQI) is given by

$$\Phi \otimes r_{n+1} = \Pi_M^W H \Phi \otimes r_n. \tag{3.18}$$

Lemmas 12, 13, and Theorem 14 continue to hold if Π_M is replaced with Π_M^W. Thus by Corollary 2, we know that (3.18) converges to a unique fixed point r_W^* such that $\Phi \otimes r_W^* = \Pi_M^W H\Phi \otimes r_W^*$.

3.6 Error Analysis

Theorem 15 *Let $\tilde{r}$ be such that $\tilde{r} = \arg\min_r ||Q^* - \Phi \otimes r||_\infty$. Let r_W^* be the fixed point of the iterates in* (3.18)*, then*

$$||Q^* - \Phi \otimes r_W^*||_\infty \leq \frac{2}{1+\alpha}(||Q^* - \Phi \otimes \tilde{r}||_\infty + ||\Phi \otimes \tilde{r} - \Pi_M^W \Phi \otimes \tilde{r}||_\infty). \tag{3.19}$$

Proof: Let $\epsilon = ||Q^* - \Phi \otimes \tilde{r}||_\infty$. By contraction property of H (Lemma 5) we know that

$$||HQ^* - H\Phi \otimes \tilde{r}||_\infty \leq \alpha\epsilon.$$

So we have $||\Phi \otimes \tilde{r} - H\Phi \otimes \tilde{r}||_\infty \leq (1+\alpha)\epsilon$. Then

$$||\Phi \otimes \tilde{r} - \Pi_M^W H\Phi \otimes \tilde{r}||_\infty = ||\Phi \otimes \tilde{r} - \Pi_M^W \Phi \otimes \tilde{r}||_\infty + ||\Pi_M^W \Phi \otimes \tilde{r} - \Pi_M^W H\Phi \otimes \tilde{r}||_\infty.$$

$$\begin{aligned} \text{Now, } \Pi_M^W \Phi \otimes \tilde{r} - \Pi_M^W H\Phi \otimes \tilde{r} &\leq \Pi_M^W \Phi \otimes \tilde{r} \\ &\quad - \Pi_M^W(\Phi \otimes \tilde{r} - (1+\alpha)\epsilon) \\ &= (1+\alpha)\epsilon. \end{aligned}$$

Similarly $\Pi_M^W H\Phi \otimes \tilde{r} - \Pi_M^W \Phi \otimes \tilde{r} \leq (1+\alpha)\epsilon$, and hence

$$||\Phi \otimes \tilde{r} - \Pi_M^W H\Phi \otimes \tilde{r}||_\infty \leq (1+\alpha)\epsilon + \beta, \tag{3.20}$$

where $\beta = ||\Phi \otimes \tilde{r} - \Pi_M^W \Phi \otimes \tilde{r}||_\infty$. Now consider the iterative scheme in (3.18) with $r_0 = \tilde{r}$, and

$$\begin{aligned}
||Q^* - \Phi \otimes r_W^*||_\infty &= ||Q^* - \Phi \otimes r_0 + \Phi \otimes r_0 \\
&\quad - \Phi \otimes r_1 + \ldots - \Phi \otimes r_W^*||_\infty \\
&\leq ||Q^* - \Phi \otimes r_0||_\infty + \\
&\quad ||\Phi \otimes r_0 - \Phi \otimes r_1||_\infty \\
&\quad + ||\Phi \otimes r_1 - \Phi \otimes r_2||_\infty + \ldots \\
&\leq \epsilon + (1+\alpha)\epsilon + \beta + \alpha((1+\alpha)\epsilon + \beta) \\
&\quad + \ldots \\
&= \epsilon(\frac{1+\alpha}{1-\alpha} + 1) + \frac{\beta}{1-\alpha} \\
&= \frac{2\epsilon + \beta}{1-\alpha}.
\end{aligned}$$

The error bound for AQI is obtained by letting $\beta = 0$ in Theorem 15.

Corollary 3 *Let r^* and r_W^* be the fixed points of iterations* (3.15) *and* (3.18) *respectively, and let $\tilde{Q} = \Phi \otimes r^*$, $\tilde{Q}_W = \Phi \otimes r_W^*$. The greedy policies $\tilde{u}(s) = \arg\max_a \tilde{Q}(s,a)$ and $\tilde{u}_W(s) = \arg\max_a \tilde{Q}_W(s,a)$ obey*

$$||J_{\tilde{u}} - J^*|| \leq \frac{2(2\epsilon)}{(1-\alpha)^2}, ||J_{\tilde{u}_W} - J^*|| \leq \frac{2(2\epsilon+\beta)}{(1-\alpha)^2}, \tag{3.21}$$

where $\epsilon = \min_r ||Q^* - \Phi \otimes r||_\infty$.

Proof: Follows from Theorem 15, Lemma 8, and the observation that the term β is the error due to the usage of Π_M^W. Thus for solution to (3.14), $\beta = 0$.

3.6.1 Experiment in Grid World

We test our MPADP algorithm (Algorithm **??**) on a 10×10 grid world problem. There are a total of 100 states, i.e., $S = \{1, 2, \ldots, 100\}$, and a grid position with the co-ordinate (i, j) is encoded as the state $s = (i-1) \times 10 + j$. The reward matrix is as given in

Table 3.1, where each entry is an integer between 1 and 10. The grid world problem is used to model terrain exploration by autonomous decision making agents (robots). In each grid position, the agent has 8 actions corresponding to the 8 possible directions. In the corners, fewer directions are feasible, and the rest of the directions lead to the current grid position. So $A = \{1, 2, \ldots, 8\}$. Actions fail with probability of 0.1 when no movement is made and the same grid position is retained, i.e., $p_a(s, s) = 0.1,\ \forall\ a \in A,\ \forall\ s \in S$, and with probability 0.9 the agent reaches the intended grid position.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
y_1	2	5	9	5	8	3	6	10	7	3
y_2	10	10	7	1	4	4	3	8	4	4
y_3	1	2	4	10	3	9	8	5	9	5
y_4	8	3	6	10	5	1	2	5	6	3
y_5	9	2	5	5	1	1	7	5	4	9
y_6	9	2	1	5	2	2	2	4	10	2
y_7	1	9	3	4	10	7	4	6	9	3
y_8	4	6	2	10	10	8	7	6	6	2
y_9	3	6	2	4	6	7	8	9	7	3
y_{10}	9	2	3	2	1	5	1	8	6	5

Table 3.1: Grid world with rewards

Let $\{\phi_j, j = 1, \ldots, k\}, \phi_j \in \mathbf{R}^n_{\min}$ and $\{\phi^i, i = 1, \ldots, n\}, \phi^i \in \mathbf{R}^k_{\min}$ be the columns and rows respectively of the feature matrix Φ. Choice of features in the $(\min, +)$ basis is similar to the way features are designed in the conventional basis, wherein it is desirable to have the features ϕ^s and $\phi^{s'}$ of states s and s' to be orthogonal if the states s and s' are unrelated or dissimilar. This translates to the following condition in the case of $(\min, +)$ basis:

$$< \phi^s, \phi^{s'} >= +\infty. \tag{3.22}$$

The condition in (3.22) arises from the fact that $+\infty$ is the additive identity in the $(\min, +)$

basis. The exact choice of the basis however changes from problem to problem. We now present the basis used for the grid problem.

We partition the state space into k aggregate states based on the immediate reward. Let $g_{\min} = \min_s g(s), s \in S$, $g_{\max} = \max_s g(s), s \in S$ and $L = g_{\max} - g_{\min}$, then we select the features as follows:

$$\phi^s(i) = \begin{cases} 0 & : g(s) \in [g_{\min} + \frac{(i-1)L}{k}, g_{\min} + \frac{(i)L}{k}] \\ +\infty & : g(s) \notin [g_{\min} + \frac{(i-1)L}{k}, g_{\min} + \frac{(i)L}{k}], \end{cases}$$
$$\forall i = 1, \ldots, k. \tag{3.23}$$

The basis definition in (3.23) generates features such that the features corresponding to different partitions are orthogonal. In the experiments, we use 1000 in place of $+\infty$ and set $\epsilon = 0$ (see Algorithm **??**). The error values are given in Table 3.2 for discount factors 0.9 and 0.99 respectively. In Table 3.2 r_{opt} denotes the result returned by the MPADP in Algorithm **??**, and $\tilde{u}$ is the greedy policy given by

$$\tilde{u} = \arg\max_{a \in A} \left(g(s) + \alpha \sum p_a(s, s') \tilde{J}(s') \right), \tag{3.24}$$
$$\text{where } \tilde{J} = \Phi \otimes r_{opt}.$$

The results are plotted in Fig. **??**. Note that $\tilde{J} \geq J^*$. Also the error values seen in the table obey the error bounds. We also note that the algorithm finds the optimal actions for about 75 states.

Error Term	Error for $\alpha = 0.9$	Error for $\alpha = 0.99$
$\|\|J^* - \Phi \otimes r_{opt}\|\|_\infty$	9.2768	18.657
$\|\|J^* - J_{\tilde{u}}\|\|_\infty$	9.3248	99.149

Table 3.2: Error Table

The result in Theorem **??** justifies the use of WMPADP algorithm to MDPs with large state space. We demonstrate the usefulness of WMPADP algorithm by applying it to solve the 'mountain-car' problem with a continuous state space.

3.6.2 The Mountain Car Experiment

The problem is to make an under-powered car climb a one-dimensional hill (Fig. 3.2), whose position x lies in the interval $[-1.2, 0.5]$. There are 3 actions available to the car, i.e., $A = \{0, 1, 2\}$. Here $a = 0$ and $a = 2$ correspond to accelerating to the left and the right respectively. Further, $a = 1$ corresponds to no acceleration. The velocity y is limited to $[-0.07, 0.07]$. The dynamics is given by

$$y_{t+1} = y_t + 0.001(a_t - 1) - 0.0025cos(3x_t), \tag{3.25}$$

$$x_{t+1} = x_t + y_t. \tag{3.26}$$

The state space is continuous with $S = [-1.2, 0.5] \times [-0.07, 0.07]$ and the state is given by $s = (x, y), x \in [-1.2, 0.5], y \in [-0.07, 0.07]$. The goal-state is reached once the car crosses the position $x \geq 0.5$. The reward in the goal-state is 100 and the reward is 0 in all the other states. The feature vector for state $s = (x, y)$ is given by

$$\phi^s = (\left|\beta(\frac{x + 1.2}{1.7} - x_i)\right|^\gamma + \left|\beta(\frac{y + 0.07}{0.14} - y_j)\right|^\gamma, \\ 1 \leq i, j \leq k) \tag{3.27}$$

where $\beta > 0$ is a scaling factor, $\gamma > 1$ is the order and $(x_i, y_j), i = 1, \ldots, k, j = 1, \ldots, k$ are the $k \times k$ centers. We let,

$$\begin{aligned} &x_1 = -1.2 < x_2 < \ldots < x_k = 0.5, \\ &\quad \text{with } x_{n+1} - x_n = \frac{1.7}{k-1},\ 1 < n < k. \\ &y_1 = -0.07 < y_2 < \ldots < y_k = 0.07, \\ &\quad \text{with } y_{n+1} - y_n = \frac{0.14}{k-1},\ 1 < n < k-1. \end{aligned} \tag{3.28}$$

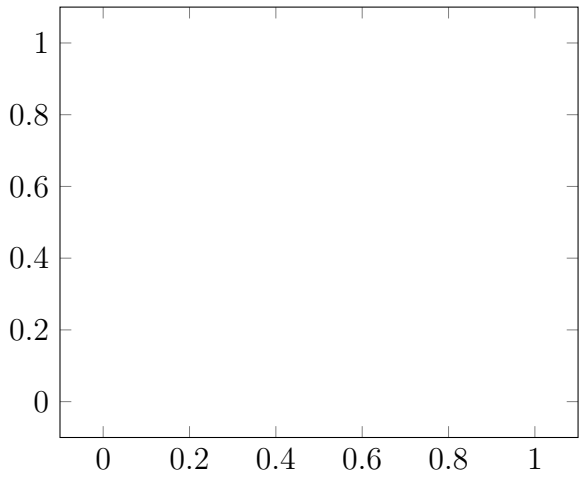

Figure 3.1: Basis Function centered at $(0, 0)$.

It is easy to check that the feature vectors as defined in (3.27) have the property given in (3.22). Further, the feature vector/basis function as defined in (3.27) is the $(\min, +)$ equivalent of radial basis functions in the conventional algebra, i.e., the basis function centered at $c = (x, y)$ takes a value equal to the multiplicative identity (in $(\min, +)$ algebra) i.e., 0, at the state c, and the values for states away from the c tend towards the additive identity in $(\min, +)$ algebra, i.e., $+\infty$. Fig. 3.1 shows the basis function centered at $(0, 0)$.

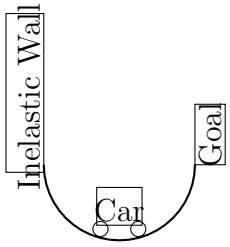

Figure 3.2: Mountain Car

In our experiments, we chose $\mathcal{L} = \{(x_i^s, y_j^s), 1 \leq i, j \leq k_1\}$, where x_i^s and y_j^s were generated according to (3.28) with $k = k_1$. We also fixed $\beta = 100$ and $\gamma = 2$, and varied $k = 5, 7, 9, 11$ and $k_1 = 30, 40, 50$, the discount factor was set to $\alpha = 0.95$, and

$\epsilon = 1e^{-5}$. The approximate value function computed by the WMPADP algorithm is shown in Fig. **??** and the actual value function is presented in Fig. 3.3. The approximate value functions computed in the various settings were used to generate corresponding greedy policies whose performance is shown in Table 3.3. The performance is evaluated via the number of steps taken by the car to reach the goal using the respective greedy policy.

k	k_1	Steps to reach the goal
5	30	285
5	40	285
5	50	285
7	30	322
7	40	322
7	50	327
9	30	218
9	40	317
9	50	324
11	30	267
11	40	260
11	50	257

Table 3.3: Number of steps taken by the *Greedy* policy

The near-optimal policy derived from the actual value function for this problem is known to achieve the goal within 150 steps [**?**]. However, finding the actual value function in this case requires significantly more computation.

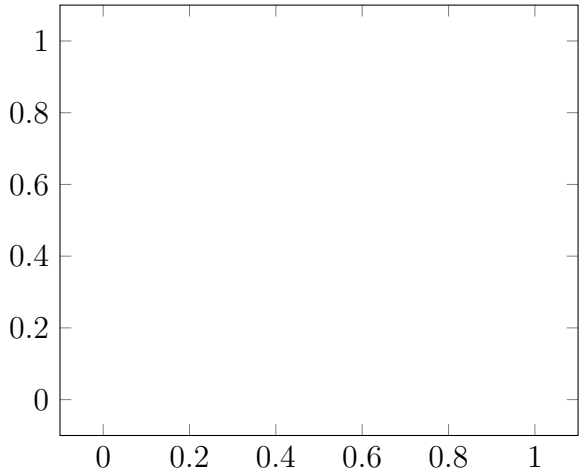

Figure 3.3: Actual Value Function Corresponding to the Mountain Car Problem

Chapter 4

Approximate Linear Program

4.1 Approximate Linear Programming

The LP formulation in (2.9) can be represented in short as,

$$\min_{J \in \mathrm{R}^n} c^\top J$$
$$\text{s.t } J \geq TJ, \tag{4.1}$$

or

$$\min_{J \in \mathrm{R}^n} c^\top J$$
$$\text{s.t } EJ \geq HJ, \tag{4.2}$$
$$\tag{4.3}$$

where $J \geq TJ$ is a shorthand for the nd constraints in (2.9) and E is an $nd \times n$ matrix given by $E = [I, \ldots, I]^\top$, i.e., E is obtained by stacking d identical $n \times n$ identity matrices one over the other. Note that (4.1) and (4.2) are identical programs and differ only in notation. We use notation of type (4.1) whenever we prefer brevity and we use notation (4.2) in some definitions and proof for the sake of clarity. The approximate linear program (ALP) is obtained by making use of LFA in the LP, i.e., by letting $J = \Phi r$ in (4.1) and

is given as

$$\begin{aligned} &\min_{r \in \mathrm{R}^k} c^\top \Phi r \\ &\text{s.t } \Phi r \geq T\Phi r. \end{aligned} \tag{4.4}$$

Unless specified otherwise we use $\tilde{r}_c$ to denote the solution to the ALP and $\tilde{J}_c = \Phi\tilde{r}_c$ to denote the corresponding approximate value function. We now state the assumptions and definitions used for the rest of the paper.

Assumption 1 *The first column of the feature matrix Φ (i.e., ϕ_1) is $\mathbf{1} \in \mathrm{R}^n$. In other words, the constant function is part of the basis.*

Assumption 2 *$c = (c(i), i = 1, \ldots, n) \in \mathrm{R}^n$ is a probability distribution, i.e., $c(i) \geq 0$ and $\sum_{i=1}^n c(i) = 1$.*

Definition 16 *Given a function $\chi \colon S \to \mathrm{R}^+$ we define the quantity β_χ as*

$$\beta_\chi = \max_{s \in S} \frac{\max_{a \in A}\left(\alpha \sum_{s'} p_a(s, s')\chi(s')\right)}{\chi(s)}. \tag{4.5}$$

Definition 17 *The function χ is then said to be a* Lyapunov *function if $\beta_\chi < 1$.*

Assumption 3 *$\psi \colon S \to \mathrm{R}^+$ is a Lyapunov function and is present in the column span of the feature matrix Φ.*

It is straightforward to check that the function $\mathbf{1}$ is a Lyapunov function and trivially is present in the column span of Φ.

Definition 18 *The modified L_1-norm is defined as*

$$||J||_{1,c} = \sum_{s \in S} c(s)|J(s)|, \tag{4.6}$$

where c obeys Assumption 2.

Definition 19 *The modified L_∞-norm is defined as follows:*

$$||J||_{\infty,\rho} = \max_{s\in S} \rho(s)|J(s)|, \tag{4.7}$$

where ρ: $\mathcal{S} \to \mathrm{R}^+$, $J \in \mathrm{R}^n$.

In Definition 19, the use of the weighting function ρ allows us to measure the error taking into account the relative importance of the various states. A lower value of $\rho(s)$ means that the state s is less important and vice-versa.
We recall below Theorem 4.2 of [**?**] which bounds the error in the approximate value function.

Theorem 20 *Let $\tilde{r}_c$ be the solution to the ALP in* (4.4)*, $\tilde{J}_c = \Phi\tilde{r}_c$, ψ be a Lyapunov function and c be a distribution as in* (2)*, then*

$$||J^* - \tilde{J}_c||_{1,c} \leq \frac{2c^\top\psi}{1-\beta_\psi} \min_r ||J^* - \Phi r||_{\infty,1/\psi}. \tag{4.8}$$

We now recall Theorem 3.1 of [**?**] that characterizes the loss in performance of the greedy policy.

Theorem 21 *Let $\tilde{u}$ be the greedy policy with respect to the solution $\tilde{J}_c$ of the ALP, then*

$$||J^* - J_{\tilde{u}}||_{1,c} \leq \frac{1}{1-\alpha}||J^* - \tilde{J}_c||_{1,c'}, \tag{4.9}$$

where $c' = (1-\alpha)c^\top(I - \alpha P_{\tilde{u}})^{-1}$.

Theorems 20 and 21 together imply that the ALP addresses both the control and prediction problems. Please refer to [**?**] for a more detailed treatment of the ALP.
Note that the ALP is a linear program in k ($<< n$) variables as opposed to the LP in (4.1) which has n variables. Nevertheless, the ALP has nd constraints (same as the LP) which is an issue when n is large and calls for constraint approximation/reduction techniques.

4.2 Lyanpunov Functions

4.3 Constraint sampling

The most important work in the direction of constraint reduction is constraint sampling [**?**] wherein a reduced linear program (RLP) is solved instead of the ALP. While the objective function of the RLP is the same as that of the ALP, the RLP has only $m << nd$ constraints sampled from the original ALP according to a given probability distribution. The reduced linear program is then given by

$$\begin{aligned} &\min_{J \in \mathrm{R}^n} c^\top J \\ &\text{s.t } J(s) \geq g_a(s) + \alpha \sum_{s'} p_a(s, s') J(s'), \ \forall (s, a) \in \mathcal{I}, \end{aligned} \tag{4.10}$$

where $\mathcal{I}$ is the index set containing the m sampled state-action pairs. Using the index set $\mathcal{I}$ we define an $nd \times m$ matrix $\mathcal{M}$, called the constraint sampling matrix as below.

Definition 22 *Let $\mathcal{I} = \{(s_1, a_1), \ldots, (s_m, a_m)\}$ be the sampled state-action pairs and let $q_i \stackrel{\Delta}{=} s_i + (a_i - 1) \times n, \forall i = 1, \ldots, m$. Then the constraint sampling matrix associated with the index set $\mathcal{I}$ is given by*

$$\begin{aligned} \mathcal{M}(i, j) &= 1, \ \textit{if } q_i = j \\ &= 0, \ \textit{otherwise.} \end{aligned} \tag{4.11}$$

The RLP can then be represented in short as

$$\begin{aligned} &\min_{r \in \mathcal{N}} c^\top \Phi r \\ &\text{s.t } \mathcal{M}^\top E \Phi r \geq \mathcal{M}^\top H \Phi r, \end{aligned} \tag{4.12}$$

where $\mathcal{M}$ is a constraint sampling matrix as in Definition 22 and $\mathcal{N} \subset \mathrm{R}^k$ is a bounded set such that $\tilde{r}_c \in \mathcal{N}$. Note that the feasible set of the RLP is a superset of the feasible set of the ALP.

The RLP based on constraint sampling has been found to perform well empirically in application domains ranging from controlled queuing networks (see section 6.2 of [**?**] and section 7 of [**?**]) to Tetris (see [**?**] and section 6 of [**?**]). However, the theoretical guarantees for the RLP are available only under a restricted setting [**?**]. We present the main result of [**?**] on constraint sampling, and to that end define the following:

Definition 23 *Given a policy u and Lyapunov function ψ in the column span of Φ, we define probability distributions μ_u, $\mu_{u,\psi}$ and constant θ as*

$$\begin{aligned} \mu^\top &\triangleq (1-\alpha)c^\top(I-\alpha P_u)^{-1}, \\ \mu_{u,\psi}(s,a) &\triangleq \frac{\mu_u(s)\psi(s)}{\mu_u^\top \psi d}, \\ \theta &\triangleq \frac{(1+\beta_\psi)}{2}\frac{\mu_{u^*}^\top \psi}{c^\top J^*}\sup_{r\in\mathcal{N}}||J^*-\Phi r||_{\infty,1/\psi}. \end{aligned} \tag{4.13}$$

We now recall Theorem 3.1 of [**?**] below.

Theorem 24 *Let ϵ and δ be scalars in $(0,1)$. Let u^* be the optimal policy and $\mathcal{I}$ an index set containing the m state-action pairs sampled independently according to the distribution $\psi_{u^*,\psi}$, for some Lyapunov function ψ, where*

$$m \geq \frac{16d\theta}{(1-\alpha)\epsilon}\big(k\ln\frac{48d\theta}{(1-\alpha)\epsilon}+\ln\frac{2}{\delta}\big). \tag{4.14}$$

Let $\tilde{r}$ be an optimal solution of the ALP and let $\hat{r}$ be the solution of the corresponding RLP, then with probability at least $1-\delta$, we have

$$||J^*-\Phi\hat{r}||_{1,c} \leq ||J^*-\Phi\tilde{r}||_{1,c}+\epsilon||J^*||_{1,c}. \tag{4.15}$$

Motivation for our work:

The result in Theorem 24 has a significant limitation in that the sampling distribution $\mu_{u^*,\psi}$ requires the knowledge of u^*. The optimal policy u^* might not be available in practice and hence it would not be possible to even formulate the specific RLP for which the bound in Theorem 24 applies. However, as mentioned before, the RLP has performed

reasonably well even when the sampling distribution is not $\mu_{u^*,\psi}$. Thus there is a gap between the theoretical understanding of the RLP and its practical efficacy which merits our attention. The gap also indicates that the RLP might be a special case of a generalized constraint reduction scheme with provable performance guarantees. Understanding such a generalized method would result in an ALP based ADP techinque that would have theoretical performance guarantees while being practically useful. In particular, we wish to answer the following open questions.

- As a natural generalization of the RLP, what happens if we define a generalized reduced linear program (GRLP) whose constraints are positive linear combinations of the original constraints of the ALP?
- Unlike [**?**] which provides error bounds for a specific RLP formulated using an idealized sampling distribution, is it possible to provide error bounds for any GRLP (and hence any RLP)?

In this paper, we address both of the questions above.

Chapter 5

Generalized Reduced Linear Program

5.1 Generalized Reduced Linear Program

In this section we present the generalized reduced linear program (GRLP) which is obtained by appropriately extending the definition of the RLP. An important property that should carry over from the RLP is that the feasible set of the GRLP should also be a superset of the feasible set of the ALP. A natural way to achieve this is to replace the set of sampled constraints in the RLP by a set of constraints which are obtained as linear combinations of the original constraints of the ALP. Formally, we define the generalized reduced linear program (GRLP) as below:

$$\begin{aligned} &\min_{r \in \chi} c^\top \Phi r, \\ &\text{s.t } W^\top E \Phi r \geq W^\top H \Phi r, \end{aligned} \tag{5.1}$$

where $W \in \mathrm{R}_+^{nd \times m}$ is an $nd \times m$ matrix with all nonnegative entries and $\chi \subset \mathrm{R}^k$ is any bounded set such that $\hat{J}_c \in \chi$. Thus the i^{th} $(1 \leq i \leq m)$ constraint of the GRLP is a positive linear combination of the original constraints of the ALP. Constraint reduction is achieved by choosing $m << nd$. The key difference between the RLP in (4.12) and the

GRLP in (5.1) despite their similar structure is that while $\mathcal{M}$ is a matrix of only zeros and ones, W is a matrix of positive entries alone. Also note that an RLP is trivially a GRLP as well. Unless specified otherwise we use $\hat{r}_c$ to denote the solution to the GRLP in (5.1), $\hat{J}_c = \Phi\hat{r}_c$, to denote the corresponding approximate value function and $\hat{u}$ to denote the greedy policy with respect to $\hat{J}_c$.

Note that we want to avoid certain uninteresting and trivial cases of W matrix such as $W = 0$ or an entire column of W being zero (which means no constraint is generated with respect to that column). Thus it is intuitive to demand that every column of W should be non-negative and have at least one entry which is strictly positive. Also, normalizing columns of W so that they sum to 1 does not make any difference to the constraints of the RLP. Keeping these in mind we also assume the following throughout the rest of the paper:

Assumption 4 *$W \in \mathrm{R}_+^{nd\times m}$ is a full rank $nd \times m$ matrix (where $m << nd$) and each of its column sums equals* 1.

The above assumption is just a technical condition that eliminates uninteresting choices such as $W = 0$ or cases when certain columns of W have all zeros, which implies that the corresponding column generates no constraint. The rest of the paper develops analytically various performance bounds and our main results provide the following:

1. A bound for $||J^* - \hat{J}_c||$, the error between the approximate value function $\hat{J}_c$ as computed by the GRLP and the optimal value function J^*;

2. a bound for $||J^* - J_{\hat{u}}||$, the loss in performance due to the greedy policy $\hat{u}$ measured with respect to the optimal policy; and

3. an important result on constraint sampling.

We achieve the above via two novel max-norm contraction operators namely the least upper bound (LUB) projection operator (denoted by Γ) and the approximate least upper bound (ALUB) projection operator (denoted by $\hat{\Gamma}$). We bound the error due to constraint approximation by analyzing the fixed points of the operators Γ and $\hat{\Gamma}$. We first establish

our results in the L_∞-norm and then extend the same in a modified L_∞-norm. The schematic in Fig. **??** provides a pictorial representation of what shall follow in the next three sections.

5.2 Least Upper Bound Projection

The least upper bound (LUB) projection operator $\Gamma\colon \mathrm{R}^n \to \mathrm{R}^n$ is defined as below:

Definition 25 *Given $J \in \mathrm{R}^n$, its least upper bound projection is denoted by ΓJ and is defined as*

$$(\Gamma J)(i) \stackrel{\Delta}{=} \min_{j=1,\ldots,k} (\Phi r_{e_j})(i),\ \forall i = 1, \ldots, n, \tag{5.2}$$

where $V(i)$ denotes the i^{th} component of the vector $V \in \mathrm{R}^n$. Also in (5.2)*, e_j is the vector with 1 in the j^{th} place and zeros elsewhere, and r_{e_j} is a solution to the linear program in* (5.3) *for $c = e_j$.*

$$\begin{aligned} r_c \stackrel{\Delta}{=} \min_{r \in \chi} & c^\top \Phi r, \\ & s.t\ \Phi r \geq TJ. \end{aligned} \tag{5.3}$$

Remark 1

1. *The definition of LUB operator $\Gamma\colon \mathrm{R}^n \to \mathrm{R}^n$ involves n associated linear programs.*
2. *Observe that $\Gamma J \geq TJ$ (follows from the fact that if $a \geq c$ and $b \geq c$, then $\min(a, b) \geq c$, where $a, b, c \in \mathrm{R}$).*
3. *Given Φ and $J \in \mathrm{R}^n$, define $\mathcal{F} \stackrel{\Delta}{=} \{\Phi r | \Phi r \geq TJ\}$. Thus $\mathcal{F}$ is the set of all vectors in the span of Φ that upper bound TJ. By fixing c in the linear program in* (5.3) *we select a unique vector $\Phi r_c \in \mathcal{F}$. The LUB projection operator Γ picks n vectors $\Phi r_{e_i}, i = 1, \ldots, n$ from the set $\mathcal{F}$ and ΓJ is obtained by computing their componentwise minimum.*

4. *Even though* ΓJ *does not belong to the span of* Φ, ΓJ *collates the various best upper bounds that can be obtained via the linear program in* (5.3).

5. *The LUB operator* Γ *in* (5.2) *bears close similarity to the ALP in* (4.4).

Definition 26 *The LUB projection of* J^* *is denoted by* $\bar{J} = \Gamma J^*$.

We now characterize the LUB projection operator Γ in the following lemmas (all the proofs are presented in the Appendix). As mentioned earlier, the error analysis depends on two max-norm contraction operators the first of which is Γ. The important result of this section is Theorem 34 and it relates the fixed point $\tilde{V}$ of Γ to J^*.

Lemma 27 *Let* $r^* \in \mathrm{R}^k$ *be defined as* $r^* \stackrel{\Delta}{=} \arg\min_{r \in R^k} ||J^* - \Phi r||_\infty$, *then*

$$||J^* - \bar{J}||_\infty \leq 2||J^* - \Phi r^*||_\infty. \tag{5.4}$$

Proof: The result follows from the definition of Γ in (5.2) and the construction of V_0, Assumption 1, and the fact that $\Phi r^* + ||J^* - \Phi r^*||_\infty \mathbf{1} \geq TJ^*$. To see this, note that

$$\begin{aligned}
&\Gamma J^* = \hat{J}_c \geq J^*, \\
&\Phi r^* + ||J^* - \Phi r^*||_\infty \geq TJ^* = J^*. \\
&\text{Thus,} \\
&\Phi r^* + ||J^* - \Phi r^*||_\infty \geq \Gamma J^* \geq TJ^*.
\end{aligned} \tag{5.5}$$

Lemma 28 *For* $J_1, J_2 \in \mathrm{R}^n$ *such that* $J_1 \geq J_2$, *we have* $\Gamma J_1 \geq \Gamma J_2$.

Proof: Choose any $i \in \{1, \ldots, n\}$ and let $r^1_{e_i}$ and $r^2_{e_i}$ be solutions to the linear program in (5.3) for $c = e_i$ with $J = J_1$ and $J = J_2$ respectively. Since $J_1 \geq J_2$, we have $TJ_1 \geq TJ_2$ and $e_i^\top \Phi r^1_{e_i} \geq e_i^\top \Phi r^2_{e_i}$, i.e., $(\Phi r^1_{e_i})(i) \geq (\Phi r^2_{e_i})(i)$. The result follows from the fact that $(\Gamma J)(i) = (\Phi r_{e_i})(i),\ \forall J \in \mathrm{R}^n$, and our choice of i was arbitrary.

Lemma 29 *Let $A \in \mathrm{R}^{u\times v}$, $b, c \in R^u$, $x_0 \in R^v$ and $b_0 = Ax_0$. Then*

$$\min\{c^\top Ax : Ax \geq b + b_0\} = \min\{c^\top Ax : Ax \geq b\} + c^\top b_0. \tag{5.6}$$

Proof: The claim can be shown by a simple change of variables.

Lemma 30 *Let $J_1 \in \mathrm{R}^n$ and $t \in \mathrm{R}$ be a constant. If $J_2 = J_1 + k\mathbf{1}$, then $\Gamma J_2 = \Gamma J_1 + \alpha t\mathbf{1}$.*

Proof: Consider the i^{th} linear programs associated with ΓJ_1 and ΓJ_2. The result follows by using Lemma 29 with $A = \Phi$, $b = TJ$, $c = e_i$, $b_0 = \alpha t\mathbf{1}$ and $x_0 = \alpha t e_i$.

Theorem 31 *The operator $\Gamma\colon \mathrm{R}^n \to \mathrm{R}^n$ obeys the* max*-norm contraction property with factor α.*

Proof: Given $J_1, J_2 \in \mathrm{R}^n$, let $\epsilon = ||J_1 - J_2||_\infty$. Thus,

$$J_2 - \epsilon\mathbf{1} \leq J_1 \leq J_2 + \epsilon\mathbf{1}. \tag{5.7}$$

From Lemmas 28 and 30, we can write

$$\Gamma J_2 - \alpha\epsilon\mathbf{1} \leq \Gamma J_1 \leq \Gamma J_2 + \alpha\epsilon\mathbf{1}. \tag{5.8}$$

Corollary 4 *The iterative scheme in* (5.9) *based on the LUB projection operator Γ in* (5.2) *converges to a unique fixed point $\tilde{V}$.*

$$V_{n+1} = \Gamma V_n, \ \forall n \geq 0. \tag{5.9}$$

Lemma 32 *$\tilde{V}$, the unique fixed point of the iterative scheme* (5.9)*, obeys $\tilde{V} \geq T\tilde{V}$.*

Proof: Consider the i^{th} linear program associated with $\Gamma\tilde{V}$. We know that $\Phi r_{e_i} \geq T\tilde{V}$, $\forall i = 1, \ldots, n$. The result follows from noting that $\tilde{V}$ is the unique fixed point of Γ and that $\tilde{V}(i) = \min\limits_{j=1,\ldots,n}(\Phi r_{e_j})(i)$.

Lemma 33 *$\tilde{V}$, the unique fixed point of the iterative scheme* (5.9)*, and the solution $\tilde{J}_c$ to the ALP in* (4.4)*, obey the relation $\tilde{J}_c \geq \tilde{V} \geq J^*$.*

Proof: Since $\tilde{V} \geq T\tilde{V}$ it follows that $\tilde{V} \geq J^*$. Let $\Phi r_1, \Phi r_2, \ldots, \Phi r_n$ be solutions to the ALP in (4.4) for $c = e_1, e_2, \ldots, e_n$ respectively. Now consider the iterative scheme in (5.9) with $V_0(i) = \min_{j=1,\ldots,n} (\Phi r_j)(i)$. It is clear from the definition of V_0 that $\tilde{J}_c(i) \geq \Phi r_i(i) \geq V_0(i),\ \forall i = 1, \ldots, n$. Also from the monotone property of T, we have

$$\begin{aligned} \Phi r_i &\geq V_0, \\ T\Phi r_i &\geq TV_0, \text{we also have} \\ \Phi r_i \geq T\Phi r_i &\geq TV_0, \text{ by taking component-wise minimum,} \\ V_0 &\geq TV_0. \end{aligned} \tag{5.10}$$

From the first three inequalitites in (5.10), $\Phi r_i \geq T\Phi r_i \geq TV_0,\ \forall i = 1 \rightarrow n$ and hence $V_0 \geq TV_0$. Since $V_1 = \Gamma V_0$, from the definition of Γ in (5.2) we have $V_0 \geq V_1$, and recursively $V_n \geq V_{n+1},\ \forall n \geq 0$. So it follows that $\tilde{J}_c \geq V_0 \geq V_1 \ldots \geq \tilde{V}$.

Theorem 34 *Let $\tilde{V}$ be the fixed point of the iterative scheme in* (5.9) *and let $\bar{J}$ be the best possible projection of J^* as in Definition 26, then*

$$||J^* - \tilde{V}||_\infty \leq \frac{1}{1-\alpha} ||J^* - \bar{J}||_\infty. \tag{5.11}$$

Proof: Let $\epsilon = ||J^* - \bar{J}||_\infty$, and $\{V_n\}, n \geq 0$ be the iterates of the scheme in (5.9) with $V_0 = \bar{J}$, then

$$\begin{aligned} ||J^* - \tilde{V}||_\infty &\leq ||J^* - V_0 + V_0 - V_1 + V_1 \ldots - \tilde{V}||_\infty \\ &\leq ||J^* - V_0||_\infty + ||V_0 - V_1||_\infty + \ldots \end{aligned}$$

Since $||V_1 - V_0||_\infty = ||\Gamma\bar{J} - \Gamma J^*||_\infty \leq \alpha ||\bar{J} - J^*||_\infty$, from Theorem 31,

$$\begin{aligned} ||J^* - \tilde{V}||_\infty &\leq \epsilon + \alpha\epsilon + \alpha^2\epsilon + \ldots \\ &= \frac{\epsilon}{1-\alpha}. \end{aligned} \tag{5.12}$$

5.3 Approximate Least Upper Bound Projection

We define an approximate least upper bound (ALUB) projection operator which has a structure similar to the GRLP and is an approximation to the LUB operator.

Definition 35 *Given $J \in \mathrm{R}^n$, its approximate least upper bound (ALUB) projection is denoted by $\hat{\Gamma}J$ and is defined as*

$$(\hat{\Gamma}J)(i) \stackrel{\Delta}{=} \min_{j=1,\ldots,k} (\Phi r_{e_j})(i),\ \forall i = 1, \ldots, n, \tag{5.13}$$

where r_{e_j} is a solution to the linear program in (5.14) *for $c = e_j$, and e_j is the same as in Definition 25.*

$$\begin{aligned} r_c \stackrel{\Delta}{=} \min_{r \in \chi} & c^\top \Phi r, \\ & s.t\ W^\top E \Phi r \geq W^\top H J, W \in \mathrm{R}_+^{nd \times m}. \end{aligned} \tag{5.14}$$

Note that W in (5.14) is the same matrix that is used in (5.1) and satisfies Assumption 4.

Lemma 36 *For $J_1, J_2 \in \mathrm{R}^n$ such that $J_1 \geq J_2$, we have $\hat{\Gamma}J_1 \geq \hat{\Gamma}J_2$.*

Proof: The proof follows from Assumptions 4 and 1 using arguments along the lines of Lemma 28.

Lemma 37 *Let $J_1 \in \mathrm{R}^n$ and $t \in \mathrm{R}$ be a constant. If $J_2 = J_1 + t\mathbf{1}$, then $\hat{\Gamma}J_2 = \hat{\Gamma}J_1 + \alpha t\mathbf{1}$.*

Proof: The proof follows from Assumption 4 and 1, as well as Lemma 29 using arguments along the lines of Lemma 30. In particular, consider the i^{th} linear program corresponding to $\hat{\Gamma}J_1$ and $\hat{\Gamma}J_2$. Now, the result follows by letting $A = W^\top E\Phi$, $b = W^\top HJ$, $c = e_i$, $b_0 = \alpha t\mathbf{1}$, $x_0 = \alpha t e_i$.

Theorem 38 *The operator $\hat{\Gamma}\colon \mathrm{R}^n \to \mathrm{R}^n$ obeys the* max*-norm contraction property with factor α and the following iterative scheme based on the ALUB projection operator $\hat{\Gamma}$, see* (5.15)*, converges to a unique fixed point $\hat{V}$.*

$$V_{n+1} = \hat{\Gamma}V_n,\ \forall n \geq 0. \tag{5.15}$$

Proof: Follows along similar lines as the proof of Theorem 31.

Lemma 39 *The unique fixed point $\hat{V}$ of the iteration in* (5.15) *and the solution $\hat{J}_c$ of the GRLP obey $\hat{J}_c \geq \hat{V}$.*

Proof: Follows in a similar manner as the proof of Lemma 33. To elaborate, let $\Phi r_1, \Phi r_2, \ldots, \Phi r_n$ be solutions to the GRLP in (5.1) for $c = e_1, e_2, \ldots, e_n$ respectively. Now consider the iterative scheme in (5.15) with $V_0(i) = \min_{j=1,\ldots,n} (\Phi r_j)(i)$. It is clear from the definition of V_0 that $\hat{J}_c(i) \geq \Phi r_i(i) \geq V_0(i),\ \forall i = 1, \ldots, n$. Also from the monotone property of T we have

$$\begin{aligned} \Phi r_i &\geq V_0, \\ H\Phi r_i &\geq HV_0, \text{we also have} \\ E\Phi r_i \geq H\Phi r_i &\geq HV_0, \text{ by taking component-wise minimum,} \\ EV_0 &\geq HV_0. \end{aligned} \tag{5.16}$$

Since $V_1 = \hat{\Gamma} V_0$, from the definition of $\hat{\Gamma}$ in (5.2) and the construction of V_0, we have $V_0 \geq V_1$, and recursively $V_n \geq V_{n+1},\ \forall n \geq 0$. So it follows that $\hat{J}_c \geq V_0 \geq V_1 \ldots \geq \hat{V}$.

Theorem 40 *Let $\hat{V}$ be the fixed point of the iterative scheme in* (5.15) *and let $\bar{J}$ be the best possible approximation of J^* as in Definition 26, then*

$$||J^* - \hat{V}||_\infty \leq \frac{||J^* - \bar{J}||_\infty + ||\Gamma J^* - \hat{\Gamma} J^*||_\infty}{1 - \alpha}. \tag{5.17}$$

Proof: Let $\epsilon = ||J^* - \bar{J}||_\infty$, and $\{V_n\}, n \geq 0$ be the iterates of the scheme in (5.15) with $V_0 = \hat{\Gamma} J^*$, then

$$\begin{aligned} ||J^* - \hat{\Gamma} J^*||_\infty &\leq ||J^* - \Gamma J^*||_\infty + ||\Gamma J^* - \hat{\Gamma} J^*||_\infty \\ &= \epsilon + \beta, \end{aligned} \tag{5.18}$$

where $\beta = ||\Gamma J^* - \hat{\Gamma} J^*||_\infty$. Now

$$\begin{aligned} ||J^* - \hat{V}||_\infty &\leq ||J^* - V_0 + V_0 - V_1 + V_1 \ldots - \hat{V}||_\infty \\ &\leq ||J^* - V_0||_\infty + ||V_0 - V_1||_\infty + ||V_1 - V_2||_\infty + \ldots \\ &= ||J^* - V_0||_\infty + ||\hat{\Gamma} J^* - \hat{\Gamma} V_0||_\infty + \ldots \\ &\leq (\epsilon + \beta) + \alpha(\epsilon + \beta) + \ldots \\ &= \frac{\epsilon + \beta}{1 - \alpha}. \end{aligned} \tag{5.19}$$

Corollary 5 *Let $\hat{V}$, $\bar{J}$ be as in Theorem 40 and let $r^* \stackrel{\Delta}{=} \arg\min_{r \in \mathrm{R}^k} ||J^* - \Phi r||_\infty$, then*

$$||J^* - \hat{V}||_\infty \leq \frac{2||J^* - \Phi r^*||_\infty + ||\Gamma J^* - \hat{\Gamma} J^*||_\infty}{1 - \alpha}. \tag{5.20}$$

Proof: The result is obtained by using Lemma 27 to replace the term $||J^* - \bar{J}||_\infty$ in Theorem 40.

5.4 A simple bound

The following lemmas relate the fixed point $\hat{V}$ of $\hat{\Gamma}$ to the solution $\hat{J}_c$ of the GRLP in (5.1).

Lemma 41 *$\hat{r} \in \mathrm{R}^k$ is a solution to GRLP in (5.1) iff it solves the following program:*

$$\begin{aligned} &\min_{r \in \chi} ||\Phi r - \hat{V}||_{1,c} \\ &s.t\ W^\top \Phi r \geq W^\top T \Phi r. \end{aligned} \tag{5.21}$$

Proof: We know from Lemma 39 that $\hat{J}_c \geq \hat{V}$, and thus minimizing $||\Phi r - \hat{V}||_{1,c} = \sum_{i=1}^n c(i)|(\Phi r)(i) - \hat{V}(i)| = c^\top \Phi r - c^\top \hat{V}$, is the same as minimizing $c^\top \Phi r$.

Theorem 42 *Let $\hat{V}$ be the solution to the iterative scheme in* (5.15) *and let $\hat{J}_c = \Phi\hat{r}_c$ be the solution to the GRLP. Let $\bar{J}$ be the best possible approximation to J^* as in Definition 26, and $||\Gamma J^* - \hat{\Gamma}J^*||_\infty$ be the error due to ALUB projection and let $r^* \stackrel{\Delta}{=} \arg\min_{r\in \mathrm{R}^k} ||J^* - \Phi r||_\infty$, then*

$$||\hat{J}_c - \hat{V}||_{1,c} \leq \frac{4||J^* - \Phi r^*||_\infty + ||\Gamma J^* - \hat{\Gamma}J^*||_\infty}{1-\alpha}. \tag{5.22}$$

Proof: Let $\gamma = ||J^* - \Phi r^*||_\infty$, then it is easy to see that

$$\begin{aligned} ||J^* - T\Phi r^*||_\infty &= ||TJ^* - T\Phi r^*||_\infty \leq \alpha\gamma, \text{ and} \\ ||T\Phi r^* - \Phi r^*||_\infty &\leq (1+\alpha)\gamma. \end{aligned} \tag{5.23}$$

From Assumption 1 there exists $r' \in \mathrm{R}^k$ such that $\Phi r' = \Phi r^* + \frac{(1+\alpha)\gamma}{1-\alpha}\mathbf{1}$ and r' is feasible to the ALP. Now

$$\begin{aligned} ||\Phi r' - J^*||_\infty &\leq ||\Phi r^* - J^*||_\infty + ||\Phi r' - \Phi r^*||_\infty \\ &\leq \gamma + \frac{(1+\alpha)\gamma}{1-\alpha} = \frac{2\gamma}{1-\alpha}. \end{aligned} \tag{5.24}$$

Since r' is also feasible for GRLP in (5.1) we have

$$\begin{aligned} ||\hat{J}_c - \hat{V}||_{1,c} &\leq ||\Phi r' - \hat{V}||_{1,c} \\ &\leq ||\Phi r' - \hat{V}||_\infty \text{ (Since } c \text{ is a distribution)} \\ &\leq ||\Phi r' - J^*||_\infty + ||J^* - \hat{V}||_\infty. \end{aligned}$$

The result follows from Corollary 5.

Prediction Error bound in the L_∞-norm

Corollary 6 *Let $\hat{J}_c$, $\hat{V}$, r^* and J^* be as in Theorem 42, then*

$$||J^* - \hat{J}_c||_{1,c} \leq \frac{6||J^* - \Phi r^*||_\infty + 2||\Gamma J^* - \hat{\Gamma}J^*||_\infty}{1-\alpha}. \tag{5.25}$$

Proof:

$$\begin{aligned} ||J^* - \hat{J}_c||_{1,c} &\leq ||J^* - \hat{V}||_{1,c} + ||\hat{V} - \hat{J}_c||_{1,c} \\ &\leq ||J^* - \hat{V}||_{\infty} + ||\hat{V} - \hat{J}_c||_{1,c} \end{aligned}$$

The result now follows from Corollary 5 and Theorem 42. The results presented in Corollary 6 is in the L_∞-norm. In the next section, we use of Lyapunov functions to provide an improved bound in a modified L_∞-norm.

5.5 Improved Bounds

In this section, we present improved error bounds by making use of Lyapunov functions.

Lemma 43 *Let $r^* \in \mathrm{R}^k$ be defined as $r^* \stackrel{\Delta}{=} \arg\min_{r \in R^k} ||J^* - \Phi r||_{\infty,1/\psi}$, then*

$$||J^* - \bar{J}||_{\infty,1/\psi} \leq 2||J^* - \Phi r^*||_{\infty,1/\psi}. \tag{5.26}$$

Proof: The result follows from the definition of Γ in (5.2), Assumption 3 and the fact that $\Phi r^* + ||J^* - \Phi r^*||_{\infty,1/\psi}\psi \geq TJ^*$.

Since most of our analysis in sections 5.2 and 5.3 depended on showing that Γ is a contraction map in the L_∞ norm we first show that Γ is also a contraction map in the modified L_∞ norm.

Lemma 44 *Let $J_1 \in \mathrm{R}^n$ and $k \in \mathrm{R}$ be a constant. If $J_2 = J_1 + k\psi$, then $\Gamma J_2 \leq \Gamma J_1 + \beta_\psi k\psi$.*

Proof: The result follows in a similar manner as the proofs for Lemmas 30 and 37 by using the result in Lemma 29.

Theorem 45 *The operator $\Gamma\colon \mathrm{R}^n \to \mathrm{R}^n$ is a contraction operator in modified L_∞ with factor β_ψ.*

Proof: Given $J_1, J_2 \in \mathrm{R}^n$ let $\epsilon = ||J_1 - J_2||_{\infty,1/\psi}$. Thus

$$J_2 - \epsilon\psi \leq J_1 \leq J_2 + \epsilon\psi. \tag{5.27}$$

From Lemmas 28 and 44, we can write

$$\Gamma J_2 - \beta_\psi \epsilon\psi \leq \Gamma J_1 \leq \Gamma J_2 + \beta_\psi \epsilon\psi. \tag{5.28}$$

Thus

$$||\Gamma J_1 - \Gamma J_2||_{\infty,1/\psi} \leq \beta_\psi ||J_1 - J_2||_{\infty,1/\psi}. \tag{5.29}$$

Corollary 7 *$\hat{\Gamma}$ is also a contraction map in the modified L_∞ norm.*

Proof: Follows from arguments similar to Theorem 45.

Lemma 46 *Let $\hat{V}$, $\bar{J}$ be as in Theorem 40 and let $r^* \stackrel{\Delta}{=} \arg\min_{r\in\mathrm{R}^k} ||J^* - \Phi r||_{\infty,1/\psi}$ then*

$$||J^* - \hat{V}||_{\infty,1/\psi} \leq \frac{2||J^* - \Phi r^*||_{\infty,1/\psi} + ||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}}{1 - \beta_\psi}. \tag{5.30}$$

Proof: The proof follows from Lemma 45, Corollary 7 and by replacing the $||\cdot||_\infty$ norm by $||\cdot||_{\infty,1/\psi}$ in the arguments presented in sections 5.2 and 5.3 leading to Corollary 5. We now recall Lemma 4.3 of [**?**].

Lemma 47 *Let ψ be a Lyapunov function that belongs to the column span of Φ , $r \in \mathrm{R}^k$ be an arbitrary vector and let r' be such that*

$$\Phi r' = \Phi r + ||J^* - \Phi r||_{\infty,1/\psi}(\frac{1+\beta_\psi}{1-\beta_\psi})\psi. \tag{5.31}$$

Then r' is feasible for the ALP in (4.4).

Theorem 48 *Let $\hat{V}$ be the solution to the iterative scheme in* (5.15) *and let $\hat{J}_c = \Phi\hat{r}_c$ be the solution to the GRLP. Let $\bar{J}$ be the best possible approximation to J^* as in Definition 26, and $||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}$ be the error due to ALUB projection and let $r^* \stackrel{\Delta}{=}$*

$\arg\min_{r\in\mathrm{R}^k} ||J^* - \Phi r||_{\infty,1/\psi}$, *then*

$$||\hat{J}_c - \hat{V}||_{1,c} \leq \frac{c^\top \psi}{1-\beta_\psi}(4||J^* - \Phi r^*||_{\infty,1/\psi} + ||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}). \tag{5.32}$$

Proof: Let $\gamma = ||J^* - \Phi r^*||_{\infty,1/\psi}$, then by choosing r' as in Lemma 47 we have

$$\begin{aligned} ||\Phi r' - J^*||_{\infty,1/\psi} &\leq ||\Phi r^* - J^*||_{\infty,1/\psi} + ||\Phi r' - \Phi r^*||_{\infty,1/\psi} \\ &= \gamma + \frac{1+\beta_\psi}{1-\beta_\psi}\gamma \\ &= \frac{2}{1-\beta_\psi}\gamma. \end{aligned}$$

Since r' is also feasible for the GRLP in (5.1) we have

$$\begin{aligned} ||\hat{J}_c - \hat{V}||_{1,c} &\leq ||\Phi r' - \hat{V}||_{1,c} \\ &= \sum_{s\in S} c(s)\psi(s)\frac{|\Phi r'(s) - \hat{V}(s)|}{\psi(s)} \\ &\leq c^\top \psi ||\Phi r' - \hat{V}||_{\infty,1/\psi} \\ &\leq c^\top \psi (||\Phi r' - J^*||_{\infty,1/\psi} + ||J^* - \hat{V}||_{\infty,1/\psi}). \end{aligned} \tag{5.33}$$

The result follows from Corollary 5.

Main Result 1: Prediction Error bound in modified L_∞-norm

Theorem 49 *Let $\hat{J}_c$, $\hat{V}$, r^* and J^* be as in Theorem 48, then*

$$||J^* - \hat{J}_c||_{1,c} \leq \frac{c^\top \psi}{1-\beta_\psi}(6||J^* - \Phi r^*||_{\infty,1/\psi} + 2||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}). \tag{5.34}$$

Proof:

$$\begin{aligned} ||J^* - \hat{J}_c||_{1,c} &\leq ||J^* - \hat{V}||_{1,c} + ||\hat{V} - \hat{J}_c||_{1,c} \\ &\leq c^\top \psi ||J^* - \hat{V}||_{\infty,1/\psi} + ||\hat{V} - \hat{J}_c||_{1,c}. \end{aligned}$$

The result now follows from Lemma 46 and Theorem 48.

Main Result 2: Control Error bound in modified L_∞-norm

We now bound the performance of the greedy policy $\hat{u}$.

Theorem 50 *Let $\hat{u}$ be the greedy policy with respect to the solution $\hat{J}_c$ of the GRLP and $J_{\hat{u}}$ be its value function. Let r^* be as in Theorem 48, then*

$$||J_{\hat{u}} - \hat{J}_c||_{1,c} \leq 2\big(\frac{c^\top \psi}{1-\beta_\psi}\big)^2\big(6||J^* - \Phi r^*||_{\infty,1/\psi} + 2||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}\big). \tag{5.35}$$

Proof:

$$\begin{aligned}
||J_{\hat{u}} - \hat{J}_c||_{1,c} &= ||(I - \alpha P_{\hat{u}})^{-1}(T\hat{J}_c - \hat{J}_c)||_{1,c} \\
&\leq c^\top (I - \alpha P_{\hat{u}})^{-1}|T\hat{J}_c - \hat{J}_c| \\
&\leq c^\top (I - \alpha P_{\hat{u}})^{-1}\psi||T\hat{J}_c - \hat{J}_c||_{\infty,1/\psi} \\
&\leq \frac{c^\top \psi}{1-\beta_\psi}||T\hat{J}_c - \hat{J}_c||_{\infty,1/\psi} \\
&\leq \frac{c^\top \psi}{1-\beta_\psi}||T\hat{J}_c - TJ^* + J^* - \hat{J}_c||_{\infty,1/\psi} \\
&\leq \frac{c^\top \psi}{1-\beta_\psi}(||T\hat{J}_c - TJ^*||_{\infty,1/\psi} + ||J^* - \hat{J}_c||_{\infty,1/\psi}) \\
&\leq \frac{c^\top \psi}{1-\beta_\psi}(1+\beta_\psi)||J^* - \hat{J}_c||_{\infty,1/\psi},
\end{aligned} \tag{5.36}$$

where in the second inequality, for $x = (x_1, \ldots, x_n)^\top \in \mathrm{R}^n$, $|x| = (|x_1|, \ldots, |x_n|)^\top \in \mathrm{R}^n$. Now

$$\begin{aligned}
||J^* - J_{\hat{u}}||_{1,c} &\leq ||J^* - \hat{J}_c||_{1,c} + ||J_{\hat{u}} - \hat{J}_c||_{1,c} \\
&\leq c^\top \psi||J^* - \hat{J}_c||_{\infty,1/\psi} + c^\top \psi \frac{1+\beta_\psi}{1-\beta_\psi}||J^* - \hat{J}_c||_{\infty,1/\psi} \\
&= \frac{2c^\top \psi}{1-\beta_\psi}||J^* - \hat{J}_c||_{\infty,1/\psi}.
\end{aligned} \tag{5.37}$$

The result now follows by substituting the value of $||J^* - \hat{J}_c||_{\infty,1/\psi}$ from Corollary 49.

Note 1 *By letting* $||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi} = ||\Gamma J^* - J^* + J^* - \hat{\Gamma} J^*||_{\infty,1/\psi} \leq 2||J^* - \Phi r^*||_{\infty,1/\psi} +$

$||J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}$ *(inequality follows from Lemma 43), we can also modifiy the bounds in* (5.25) *and* (50) *as*

$$||J^* - \hat{J}_c||_{1,c} \leq \frac{c^\top \psi}{1-\beta_\psi}(10||J^* - \Phi r^*||_{\infty,1/\psi} + 2||J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}). \tag{5.38}$$

$$||J_{\hat{u}} - \hat{J}_c||_{1,c} \leq 2\big(\frac{c^\top \psi}{1-\beta_\psi}\big)^2\big(10||J^* - \Phi r^*||_{\infty,1/\psi} + 2||J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}\big). \tag{5.39}$$

Here the term $||J^* - \hat{\Gamma} J^*||$ *in* (5.38) *and* (5.39) *captures the error due to the use of both* Φ *and* W*. Though,* (5.38) *and* (5.39) *might be loser bounds than* (5.34) *and* (5.35) *respectively, the aim here is to capture the error due to function approximation as well as constraint reduction in a single term.*

5.6 Discussion

In this section we discuss the implications and insights provided by the results presented in Theorems 49 and 50.

5.6.1 On Error Terms

- The error bounds in the main results (Theorems 49 and 50) contain two factors namely

  1. $\min_{r \in \mathrm{R}^k} ||J^* - \Phi r||_{\infty,1/\psi}$,
  2. $||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}$.

  The first factor is related to the best possible approximation that can be achieved with the chosen feature matrix Φ. This term is inherent to the ALP formulation and it appears in the bounds provided by [**?**].

  The second factor is related to constraint approximation and is completely defined in terms of Φ, W and T, and does not require knowledge of stationary distribution of the optimal policy. It makes intuitive sense since given that Φ approximates J^*, it is enough for W to depend on Φ and T without any additional requirements.

- Unlike the result in [**?**] which holds only for a specific RLP formulated under ideal assumptions, our bounds hold for any GRLP and as a result for any given RLP. Another interesting feature of our result is that it holds with probability 1.

- A salient feature of the ALP formulation is the use of Lyapunov functions to control/shape the error across the states based on their relative importance. Since the error bounds are in a modified L_∞-norm, the GRLP framework retains this salient feature of the ALP.

The fact that both the prediction and control problems can be addressed by the GRLP makes it a complete ADP method, and by addressing the constraint approximation, the GRLP framework is an important addition to the theory of ALP.

5.6.2 On Constraint Reduction and Approximation

We claim the following based on the error bounds that we derived for the GRLP.
Claim 1) It is not always necessary to sample constraints according to the stationary distribution of the optimal policy.
Claim 2) Constraint approximation is not only restricted to constraint sampling but also can be extended to include linear approximation of the constraints.
The following result (Theorem 51) supports Claim 1 in the above.
Main Result 3**: On Constraint Sampling**
The error term $||\Gamma J^* - \hat{\Gamma} J^*||_{\infty,1/\psi}$ gives new insights into constraint sampling.

Theorem 51 *Let $s \in S$ be a state whose constraint was sampled. Then*

$$|\Gamma J^*(s) - \hat{\Gamma} J^*(s)| < |\Gamma J^*(s) - J^*(s)|. \tag{5.40}$$

Proof: Let r_{e_s} and $\hat{r}_{e_s}$ be solutions to the linear programs in (5.3) and (5.14) respectively for $c = e_s$ and $J = J^*$. It is easy to note that r_{e_s} is feasible for the linear program in (5.14) for $c = e_s$ and J^*, and hence it follows that $(\Phi r_{e_s})(s) \geq (\Phi \hat{r}_{e_s})(s)$. However, since all the constraints with respect to state s have been sampled we know that $(\Phi \hat{r}_{e_s})(s) \geq J^*$. The

proof follows from noting that $(\Gamma J^*)(s) = (\Phi r_{e_s})(s)$ and $\hat{\Gamma} J^*(s) = (\Phi \hat{r}_{e_s})(s)$.

The expression in (5.40) in Theorem 51 says that the additional error $|\Gamma J^*(s) - \hat{\Gamma} J^*(s)|$ due to constraint sampling is less than the original projection error $|\Gamma J^*(s) - J^*(s)|$ due to function approximation. This means that for the RLP to perform well it is enough to retain those states for which the linear function approximation via Φ is known to perform well. The modified L_∞ norm in (5.25) comes to our rescue to control the error due to those states that are not sampled. Thus the sampling distribution need not be the stationary distribution of the optimal policy as long as it samples the *important* states, an observation that might theoretically explain the empirical successes of the RLP [**? ? ?**].

To understand the implication of Claim 2 we need to look at the Lagrangian of the ALP and GRLP in (5.41) and (5.42) respectively, i.e.,

$$\tilde{L}(r, \lambda) = c^\top \Phi r + \lambda^\top (T\Phi r - \Phi r), \tag{5.41}$$

$$\hat{L}(r, q) = c^\top \Phi r + q^\top W^\top (T\Phi r - \Phi r). \tag{5.42}$$

The insight that the GRLP is a linear function approximation of the constraints (i.e., the Lagrangian multipliers) can be obtained by noting that $Wq \approx \lambda$ in (5.42). Note that while the ALP employs LFA in its objective function (i.e., use of Φr), the GRLP employs linear approximation both in the objective function (Φr) as well as the constraints (use of W). Further, W can be interpreted as the feature matrix that approximates the Lagrange multipliers as $\lambda \approx Wq$, where $\lambda \in \mathrm{R}^{nd}, r \in \mathrm{R}^m$. One can show [**?**] that the optimal Lagrange multipliers are the discounted number of visits to the "state-action pairs" under the optimal policy u^*, i.e.,

$$\begin{aligned} \lambda^*(s, u^*(s)) &= \big(c^\top (I - \alpha P_{u^*})^{-1}\big)(s) \\ &= \big(c^\top (I + \alpha P_{u^*} + \alpha^2 P_{u^*}^2 + \ldots)\big)(s), \\ \lambda^*(s, a) &= 0, \forall a \neq u^*(s), \end{aligned}$$

where P_{u^*} is the probability transition matrix with respect to the optimal policy. Even though we might not have the optimal policy u^* in practice, the fact that λ^* is a probability distribution and that it is a linear combination of $\{P_{u^*}, P_{u^*}^2, \ldots\}$ hints at the kind of features that might be useful for the W matrix.

5.6.3 Numerical Illustration

We take up an example in the domain of controlled queues from [**?**] for which the ALP has been known to work well. For this domain, we make use of our results and observations to select various useful W matrices and present their performance.

The queuing system consists of $n = 10^4$ states and $d = 4$ actions. We chose $n = 10^4$ because it was possible to solve both the GRLP and the exact LP (the latter with significant effort) so as to enumerate the approximation errors. We hasten to mention that while we could run the GRLP for queuing systems with $n > 10^4$ without much computational overhead, solving the exact LP was not possible for $n > 10^4$ as a result of which the approximation error could not be computed.

Queuing Model: The queuing model used here is similar to the one in Section 5.2 of [**?**]. We consider a single queue with arrivals and departures. The state of the system is the queue length with the state space given by $S = \{0, \ldots, n-1\}$, where $n-1$ is the buffer size of the queue. The action set $A = \{1, \ldots, d\}$ is related to the service rates. We let s_t denote the state at time t. The state at time $t+1$ when action $a_t \in A$ is chosen is given by $s_{t+1} = s_t + 1$ with probability p, $s_{t+1} = s_t - 1$ with probability $q(a_t)$ and $s_{t+1} = s_t$, with probability $(1 - p - q(a_t))$. For states $s_t = 0$ and $s_t = n-1$, the system dynamics is given by $s_{t+1} = s_t + 1$ with probability p when $s_t = 0$ and $s_{t+1} = s_t - 1$ with probability $q(a_t)$ when $s_t = n-1$. The service rates satisfy $0 < q(1) \leq \ldots \leq q(d) < 1$ with $q(d) > p$ so as to ensure 'stabilizability' of the queue. The reward associated with the action $a \in A$ in state $s \in S$ is given by $g_a(s) = -(s + 60q(a)^3)$.

Choice of Φ : We make use of polynomial features in Φ (i.e., $1, s, \ldots, s^{k-1}$) since they are known to work well for this domain [**?**]. This takes care of the term $||J^* - \Phi r^*||_\infty$ in

(5.25).

Selection of W: For our experiments, we choose two contenders for the W-matrix and compare them with the ideal sampling matrix W_i ([**?**]) and random positive matrix W_r. Our choices of the W matrix are as below.

(**i**) W_c- matrix that corresponds to sampling according to c. This is justified by the insights obtained from Theorem 51 on the error term $||\Gamma J^* - \hat{\Gamma} J^*||_\infty$, i.e., the idea of selecting the important states.

(**ii**) W_a state-aggregation matrix, a heuristic derived using the fact that λ^* is a linear combination of $\{P_{u^*}, P_{u^*}^2, \ldots\}$. Our choice of the W_a matrix to correspond to aggregation of near by states is motivated by the observation that P^n captures n^{th} hop connectivity/neighborhood information. The aggregation matrix W_a is defined as below: $\forall i = 1, \ldots, m$,

$$\begin{aligned} W_a(i,j) = 1, \ \forall j \text{ s.t } j = (i-1) \times \frac{n}{m} + k + (l-1) \times n, \\ k = 1, \ldots, \frac{n}{m}, l = 1, \ldots, d, \\ = 0, \text{ otherwise.} \end{aligned} \tag{5.43}$$

We ran our experiments on a moderately large queuing system denoted by Q_L with $n = 10^4$ and $d = 4$ with $q(1) = 0.2$, $q(2) = 0.4$, $q(3) = 0.6$, $q(4) = 0.8$, $p = 0.4$ and $\alpha = 0.98$. We chose $k = 4$ (i.e., we used $1, s, s^2$ and s^3 as basis vectors) and we chose W_a (5.43), W_c, W_i and W_r with $m = 50$. We set $c(s) = (1-\zeta)\zeta^s$, $\forall s = 1, \ldots, 9999$, with $\zeta = 0.9$ and $\zeta = 0.999$ respectively. The results in Table 5.2 show that the performance exhibited by W_a and W_c is better by several orders of magnitude over 'random' in the case of the large system Q_L and is closer to the ideal sampler W_i. Also note that a better performance of W_a and W_c in the larger system Q_L tallies with a lower value of $||\Gamma J^* - \hat{\Gamma} J^*||_\infty$ in the smaller system Q_S.

Error Terms	W_i	W_c	W_a	W_r
$\|\|J^* - \hat{J}_c\|\|_{1,c}$ for $\zeta = 0.9$	32	32	220	5.04×10^4
$\|\|J^* - \hat{J}_c\|\|_{1,c}$ for $\zeta = 0.999$	110	180.5608	82	1.25×10^7

Table 5.1: Shows values of Error Terms for Q_L.

Performance Metric	W_i	W_c	W_a
$\|\|J_{\hat{u}}\|\|_{1,c}$ for $\zeta = 0.9$	-441.25	-450.59	-446.49
$\|\|J_{\hat{u}}\|\|_{1,c}$ for $\zeta = 0.999$	$-2.0611e+04$	$-2.0611e+04$	$-2.0612e+04$

Table 5.2: Shows performance metrics for Q_L. Here $||J^*||_{1,c} = -439.26$ for $\zeta = 0.9$ and $||J^*||_{1,c} = -2.0603e + 04$ for $\zeta = 0.999$ and a random policy yields a total reward of $-1.2661e + 03$.

Empirical evidence for the performance of RLP with various sampling distributions can also be found in [**?** **?**].

5.6.4 Reinforcement Learning

Reinforcement Learning (RL) algorithms are useful in scenarios where the system is available in the form of a simulator or only samples can be obtained via direct interaction. In particular, in the RL setting, the model parameters g and P are not known explicitly and the underlying MDP needs to be solved by using sample trajectories. In short, RL algorithms are sample trajectory based solution schemes for solving MDPs whose model information is not known. RL methods learn by filtering out the noisy sample via stochastic approximation and they also employ function approximation in order to handle MDPs with large number of states. Most RL algorithms are sample trajectory based extensions of ADP methods.

The RL extension of the ALP formulation has been applied to the optimal stopping problem in [**?**]. Function approximation is employed to approximate the square root of the Lagrange multipliers. However, since the approximation is not linear, convergence of the resulting RL algorithm cannot be guaranteed. Our results theoretically justify linear

function approximation of the Lagrange multipliers, an immediate implication of which is that the RL extension of the ALP can be guaranteed to converge if the updates in [**?**] use LFA for the Lagrange multipliers instead of a non-linear approximator.

5.7 Conclusion

The approximate linear programming (ALP) is an approximate dynamic programming method that addresses the prediction and control problems successfully. However, an important shortcoming of the ALP is that it has large number of constraints, which is tackled in practice by sampling a tractable number of constraints from the ALP to formulate and solve a reduced linear program (RLP). Though RLP has been found to work well empirically in various domains ranging from queues to Tetris games, performance guarantees are available only in the case of a specific RLP formulated under idealized assumptions. Thus there has been a gap in the theory of constraint reduction.

In this paper, we introduced a novel framework based on the generalized reduced linear program formulation to study constraint reduction. The constraints of the GRLP were obtained as positive linear combinations of the original ALP. We provided an error bound that relates the optimal value function to the solution of the GRLP. Our error bound contained two terms, one inherent to the ALP formulation and the other due to constraint reduction. We also made qualitative and quantitative observations about the nature of the error term that arose due to constraint reduction. Our analysis also revealed the fact that it is not always necessary to sample according to the stationary distribution of the optimal policy and, in fact, potentially several different constraint sampling/approximation strategies might work. In particular, we also theoretically justified linear function approximation of the constraints. We also discussed the results and provided a numerical example in the domain of controlled queues. To conclude, we observe that by providing a novel theoretical framework to study constraint approximation, this paper provides important results that add to the theory of ALP.

Chapter 6

Stochastic Approximation and Reinforcement Learning

Chapter 7

Stability of Iterates in Two Timescale Stochastic Approximation

7.1 Introduction

A typical stochastic approximation [**?**] (SA) scheme can be specified as under:

$$x_{n+1} = x_n + a(n)[h(x_n) + M_{n+1}], n \geq 0, \tag{7.1}$$

where the iterates $x_n \in \mathrm{R}^d$, $h\colon \mathrm{R}^d \to \mathrm{R}^d$, and M_{n+1} are zero-mean noise terms and $a(n)$ are diminishing step sizes. The earliest and most representative example of an SA scheme is the Robbins-Monro (RM) algorithm, an iterative procedure to compute the zeros of a function using only its noisy observations. Here, h is the function whose unique zero say $x^* \in \mathrm{R}^d$ needs to be computed and at any point x_n, the value of the function $h(x_n)$ is corrupted by an additive noise M_{n+1}. Other instances of SA schemes are the stochastic gradient schemes wherein h is the gradient of some other function $f\colon \mathrm{R}^d \to \mathrm{R}$, i.e., $h(x) = \nabla f(x)$. Stochastic gradient schemes are found to be useful in actor-critic [**? ? ? ? ? ? ? ? ? ?**] and policy gradient [**? ? ? ? ? ?**] reinforcement learning algorithms as well as stochastic optimization algorithms [**? ? ? ? ? ? ? ? ? ?**].

The conditions under which the above scheme converges have been studied widely ([**?**

?]) in the literature. In general, analysis of SA schemes involves two important steps namely,

1. establishing *stability* of the iterates x_n, i.e., showing that $\sup_n \parallel x_n \parallel < \infty$ with probability one, and

2. establishing *convergence* to a desired solution x^*, i.e., showing that $x_n \rightarrow x^*$ almost surely as $n \rightarrow \infty$.

In situations where stability of iterates cannot be established analytically, a common procedure is to pick a suitable compact subset of the parameter space onto which one would projects the iterates, thereby ensuring their boundedness, i.e., $\sup_n \parallel x_n \parallel < \infty$.

The ODE method ([**?** **?**]) is a useful analytical tool to study the stability as well as convergence of the stochastic iterates. In the ODE method, one considers (7.1) to be a *noisy-discretized* version of the ODE,

$$\dot{x}(t) = h(x(t)). \tag{7.2}$$

An important notion associated with the ODE method is that of *timescale* that is defined using the quantity $t(n) = \sum_{m=0}^{n-1} a(m),\ n \geq 1$. Let $\phi(t, x), t \geq 0$ denote the trajectory of (7.2) with initial condition $x(0) = x$. When $\sup_n \parallel x_n \parallel < \infty$, the trajectory $\phi(t, x)$ sampled at instants $t(n)$, i.e., $\{\phi(t(n), x_0), n \geq 0\}$ approximates the iterates x_n (Lemma 1, Chapter 2,[**?**]). The SA scheme can then be shown to converge to a compact connected internally chain transitive invariant set of the ODE in (7.2). Thus the convergence of iterates in the SA scheme (7.1) can be understood by looking at the asymptotic equilibria of the ODE in (7.2).

The ODE method can also be used to establish the stability of iterates $\{x_n\}$ which involves understanding the behavior of (7.1) when $\parallel x_n \parallel$ gets larger. The idea is to observe the evolution of the continuously interpolated trajectory obtained from the iterates $\{x_n\}$ every T instants and then rescaling it to the unit ball if the trajectory goes out of it. The rescaled iterates $\{\tilde{x}_n\}$ then differ from the original iterates $\{x_n\}$ only by a constant factor.

The rescaled iterates track the rescaled ODE,

$$\dot{x}(t) = h_c(x(t)), \text{where } h_c(x) \stackrel{def}{=} \frac{h(cx)}{c}, \forall c \geq 1. \tag{7.3}$$

As $c \to \infty$, one obtains the following *limiting* ODE (assuming the limit exists):

$$\dot{x}(t) = h_\infty(x(t)), \text{where } h_\infty(x) = \lim_{c\to\infty} h_c(x). \tag{7.4}$$

When the ODE (7.4) is globally asymptotically stable to the origin, [**?**] shows that $\sup_n \| x_n \| < \infty$ almost surely. This is because outside a set of zero probability, as $\| x_n \|$ grows larger, the scaling factor tends to ∞ and $\tilde{x}_n$ can be approximated by the trajectory of the limiting ODE (7.4). Since the ODE (7.4) is stable to the origin, the scaled iterates $\{\tilde{x}_n\}$ fall back to the origin. The stability of $\{x_n\}$ then follows upon noting that these iterates differ from $\tilde{x}_n, n \geq 1$ only by a scaling factor.

The SA scheme in (7.1) is a *single* timescale stochastic approximation scheme. A two-timescale stochastic approximation scheme [**? ? ? ? ? ?**] is as under:

$$x_{n+1} = x_n + a(n)[h(x_n, y_n) + M^{(1)}_{n+1}], \tag{7.5a}$$

$$y_{n+1} = y_n + b(n)[g(x_n, y_n) + M^{(2)}_{n+1}], \tag{7.5b}$$

where, $x_n \in \mathrm{R}^{d_1}, y_n \in \mathrm{R}^{d_2}$, $a(n)$ and $b(n)$ are both diminishing step-size schedules except that $\frac{b(n)}{a(n)} \to 0$ as $n \to \infty$. The two timescales here are, the *slower*, corresponding to $t^s(n) = \sum_{m=0}^{n-1} b(m)$, and the *faster*, corresponding to $t^f(n) = \sum_{m=0}^{n-1} a(m)$ respectively, on which the two recursions proceed. The terms *faster* and *slower* signify the rates of growth of $t^f(n)$ and $t^s(n)$ respectively. Many iterative schemes might involve calls to subroutines which themselves are iterative in nature. Thus the *outer*-loop in such schemes must typically wait for the iterates in the *inner*-loop to converge before proceeding to its next step. Instead, the same effect can also be achieved by a two-timescale recursion with outer and inner loop iterates on the slower and faster timescales respectively. The condition $\frac{b(n)}{a(n)} \to 0$ ensures that the slower timescale iterates do not move much compared to the

faster timescale iterates, thereby producing the effect equivalent to that of a *nested*-loop. As a result, multi-timescale stochastic approximation is commonly used in algorithms that involve estimation of *nested* quantities [**?** **?** **?** **?** **?** **?**]. Study of two timescale SA schemes is also important as it is easy to extend the arguments made for two timescale recursions to those with even more timescales.

Under the assumption that $\sup_n \parallel x_n \parallel < \infty$, $\sup_n \parallel y_n \parallel < \infty$, the ODE method can be applied to establish the convergence of iterates in (7.5), see [**?**] and Chapter 6, [**?**]. In this paper, we provide *the first* conditions in the literature that imply stability of the iterates in (7.5). These are conditions that imply $\sup_n \parallel x_n \parallel < \infty$ and $\sup_n \parallel y_n \parallel < \infty$ with probability one. Stability results for the single timescale recursions, see [**?**] and Chapter 3 of [**?**], cannot be applied directly to the two-timescale scheme in (7.5), because the iterates $\{x_n\}$ and $\{y_n\}$ are coupled and they evolve along separate timescales. Nonetheless, while our analysis resembles [**?** **?**] in that we study scaled ODEs, it differs considerably from [**?**] in that we study two separate sets of scaled ODEs for the two different timescales. Since the iterates x_n evolve on the faster timescale, it is natural to first characterize the growth of $\{x_n\}$ before studying the stability of $\{y_n\}$. So, we first study the following scaled ODE and its limiting version:

$$\dot{x}(t) = h_c(x(t), y), \text{where } h_c(x,y) \stackrel{def}{=} \frac{h(cx, cy)}{c}, c \geq 1, \tag{7.6}$$

$$\dot{x}(t) = h_\infty(x(t), y), \text{where } h_\infty(x,y) = \lim_{c \to \infty} h_c(x,y). \tag{7.7}$$

Equations (7.6) and (7.7) also allude to the fact that the slower timescale iterates appear to be constant, i.e., $y(t) \equiv y \ \forall t$, when viewed from the faster timescale [**?**].

Under the assumption that the limiting ODE (7.7) has a unique globally asymptotically stable equilibrium $\lambda_\infty(y)$, where $\lambda_\infty \colon \mathrm{R}^{d_2} \to \mathrm{R}^{d_1}$ is a Lipschitz continuous map with $\lambda_\infty(0) = 0$, we show that $\parallel x_n \parallel \leq K^*(1 + \parallel y_n \parallel)$ for some $K^* > 0$. Subsequently, we analyze the slower recursion using the following ODE:

$$\dot{y}(t) = g_c(y(t)), \text{where } \ g_c(y) = \frac{g(c\lambda_\infty(y), cy)}{c}. \tag{7.8}$$

We show that $\sup_n \parallel y_n \parallel < \infty$ with probability one, when the limiting ODE $\dot{y}(t) = g_\infty(y(t))$ (with $g_\infty(y) = \lim_{c\to\infty} g_c(y)$) is stable to the origin. By making use of the fact that $\parallel x_n \parallel \leq K^*(1+ \parallel y_n \parallel)$ (see above), we establish the stability of two timescale SA iterates given in (7.5).

The paper is organized as follows. In section 7.2 we summarize certain results (mostly from [**?** **?**]) regarding the scaled ODEs that will be used in our arguments. In sections 7.3.1 and 7.3.2, we present the analysis for the faster and slower timescale recursions, respectively. In section 7.4, we apply our analysis to establish the stability of iterates in a two-timescale stochastic approximation algorithm arising in an application in reinforcement learning. Conclusions are provided in section 7.5. Finally, an Appendix at the end presents the Gronwall inequalities that are used in the proofs and a result whose proof follows along the lines of a similar result in [**?**].

7.2 Results for the Scaled ODE

Let $u\colon \mathrm{R}^{d_1+d_2} \to \mathrm{R}^{d_1}$ be a Lipschitz map, i.e., there exists $L > 0$ such that $\parallel u(x_1) - u(x_2) \parallel \leq L \parallel x_1 - x_2 \parallel, x_1, x_2 \in \mathrm{R}^{d_1+d_2}$. Let $\eta^{q(t)}(t,p)$ denote the solution to the ODE

$$\dot{p}(t) = u(p(t), q(t)),\ t \geq 0, \tag{7.9}$$

with initial condition $p \in \mathrm{R}^{d_1}$ and the external input $q(t) \in \mathrm{R}^{d_2}$. The superscript $q(t)$ in $\eta^{q(t)}(t,p)$ is to be understood as a symbol that indicates the fact that the external input is $q(t)$. We denote by $\eta^q(t,p)$ the solution to the ODE

$$\dot{p}(t) = u(p(t), q),\ t \geq 0. \tag{7.10}$$

Thus, in (7.10), $q(t) = q, \forall t \geq 0$ for some $q \in \mathrm{R}^{d_2}$. Hence, in (7.10), the external input is a constant as opposed to being a function of time as with (7.9). Let $B(p_0, r) = \{p\colon \parallel p - p_0 \parallel \leq r\}$ denote the closed ball of radius r around p_0. The lemma below is a generalization of Lemma 1, Chapter 3 of [**?**].

Lemma 52 *Let $K \subset \mathrm{R}^{d_1}$ be a compact set and $q \in \mathrm{R}^{d_2}$ be a fixed external input to the ODE in* (7.10)*. If this ODE has a unique globally asymptotically stable equilibrium (a.s.e) $\lambda(q) \in \mathrm{R}^{d_1}$, then given any $\delta > 0$, there exists a $T_\delta > 0$ such that for all initial conditions $p \in K$, we have*

$$\eta^q(t,p) \in B(\lambda(q),\delta), \forall t \geq T_\delta.$$

Proof: Since $\lambda(q)$ is the unique globally a.s.e, it is Lyapunov stable. Hence, there exists a $\delta' > 0$ (assume $\delta' < \delta$ without loss of generality) such that any trajectory starting in $B(\lambda(q),\delta')$ stays within $B(\lambda(q),\delta)$ forever, i.e., for any $s \geq 0$,

$$\eta^q(s,p) \in B(\lambda(q),\delta') \Rightarrow \eta^q(t,p) \in B(\lambda(q),\delta),\ \forall t \geq s. \tag{7.11}$$

Let $p \in K$ be a given initial condition and since $\lambda(q)$ is a globally a.s.e, there exists a T_p such that $\eta^q(T_p,p) \in B(\lambda(q),\delta'/2)$. Let p' be some other initial condition, then for $t \in [0,T_p]$,

$$\eta^q(t,p) = p + \int_0^t u(\eta^q(s,p),q)ds,$$
$$\eta^q(t,p') = p' + \int_0^t u(\eta^q(s,p'),q)ds.$$

Subtracting the two equations above, we get

$$\begin{aligned} \parallel \eta^q(t,p) - \eta^q(t,p') \parallel &\leq \parallel p - p' \parallel + L\int_0^t \parallel \eta^q(s,p) - \eta^q(s,p') \parallel ds \\ &\leq \parallel p - p' \parallel e^{LT_p}, \forall t \in [0,T_p]. \end{aligned} \tag{7.12}$$

The last inequality above follows from the Gronwall inequality (see Appendix). Given any $p \in K$, one can always choose p' close enough to p such that $\parallel p - p' \parallel e^{LT_p} < \delta'/2$ and $\eta^q(T_p,p') \in B(\lambda(q),\delta')$. Thus there is a neighborhood U_p of p such that for all $p' \in U_p$, $\eta^q(t,p') \in B(\lambda(q),\delta'), \forall t \geq T_p$. Now $\underset{p\in K}{\cup} U_p$ is a cover for K and since K is compact there exists a finite sub-cover $\{U_{p_1},\ldots,U_{p_n}\}$ such that $K \subset \cup_{i=1}^n U_{p_i}$. Let

$T_\delta = \max\{T_{p_i}, i = 1, \ldots, n\}$, then for any $p \in K$,

$$\eta^q(T_\delta, p) \in B(\lambda(q), \delta') \Rightarrow \eta^q(t, p) \in B(\lambda(q), \delta),\ \forall t \geq T_\delta. \tag{7.13}$$

Assumption 5 *Let functions* $u_c \colon \mathrm{R}^{d_1+d_2} \to \mathrm{R}^{d_1}, c \geq 1$ *be a family of Lipschitz continuous functions with Lipschitz constant* L. *Let* $u_c \to u_\infty$ *as* $c \to \infty$ *uniformly on compacts to some* $u_\infty \in \mathcal{C}(R^{d_1+d_2})$.

Lemma 53 $u_\infty \colon \mathrm{R}^{d_1+d_2} \to \mathrm{R}^{d_1}$ *is also a Lipschitz continuous function with Lipschitz constant* L.

Proof: Given $x_1, x_2 \in \mathrm{R}^{d_1+d_2}$, we have

$$\begin{aligned} \| u_\infty(x_1) - u_\infty(x_2) \| \leq & \| u_\infty(x_1) - u_c(x_1) \| + \| u_c(x_1) - u_c(x_2) \| + \| u_c(x_2) - u_\infty(x_2) \| \\ & \leq 2\epsilon(c) + L \| x_1 - x_2 \|, \end{aligned} \tag{7.14}$$

where since c is arbitrary, we let $c \to \infty$, whereby $\epsilon(c) \to 0$. The second inequality above follows from Assumption 5. So we have

$$\| u_\infty(x_1) - u_\infty(x_2) \| \leq L \| x_1 - x_2 \| . \tag{7.15}$$

Let $\eta_c^{q(t)}(t, p)$ and $\eta_\infty^{q(t)}(t, p)$ be solutions[1] to the ODEs (7.16) and (7.17), respectively, with $p(0) = p$.

$$\dot{p}(t) = u_c(p(t), q(t)), \tag{7.16}$$

$$\dot{p}(t) = u_\infty(p(t), q(t)). \tag{7.17}$$

Lemma 54 below is a generalization of Lemma 2, Chapter 3 of [**?**].

[1]When the external input is constant, i.e., $q(t) = q, \forall t \geq 0$, we denote the same by $\eta_c^q(t, p)$ and $\eta_\infty^q(t, p)$, respectively.

Lemma 54 *Let $p \in \mathrm{R}^{d_1}$, $q \in \mathrm{R}^{d_2}$, $[0,T]$ be a given time interval and $r > 0$ be a small positive constant. Let $q'(t) \in B(q,r),\ \forall t \in [0,T]$, then*

$$\| \eta_c^{q'(t)}(t,p) - \eta_\infty^q(t,p) \| \leq (\epsilon(c) + Lr)Te^{LT},\ \forall t \in [0,T],$$

where $\epsilon(c)$ is independent of p and q and $\epsilon(c) \to 0$, as $c \to \infty$.

Proof:

$$\begin{aligned} \eta_c^{q'(t)}(t,p) &= p + \int_0^t u_c(\eta_c^{q'(t)}(s,p), q'(t))ds, \\ \eta_\infty^q(t,p) &= p + \int_0^t u_\infty(\eta_\infty^q(s,p), q)ds, \end{aligned} \tag{7.18}$$

and hence

$$\| \eta_c^{q'(t)}(t,p) - \eta_\infty^q(t,p) \| \leq \int_0^t \| u_c(\eta_c^{q'(t)}(s,p), q'(t)) - u_\infty(\eta_\infty^q(s,p), q) \| ds. \tag{7.19}$$

Since u_c and u_∞ are Lipschitz continuous, we have

$$\begin{aligned} &\| u_c(\eta_c^{q'(t)}(s,p), q'(t)) - u_\infty(\eta_\infty^q(s,p), q) \| \\ &\leq \| u_c(\eta_c^{q'(t)}(s,p), q'(t)) - u_c(\eta_\infty^q(s,p), q'(t)) \| \\ &+ \| u_c(\eta_\infty^q(s,p), q'(t)) - u_\infty(\eta_\infty^q(s,p), q'(t)) \| \\ &+ \| u_\infty(\eta_\infty^q(s,p), q'(t)) - u_\infty(\eta_\infty^q(s,p), q) \| \\ &\leq L \| \eta_c^{q'(t)}(s,p) - \eta_\infty^q(s,p) \| + \epsilon(c) + L \| q - q'(t) \| \\ &\leq L \| \eta_c^{q'(t)}(s,p) - \eta_\infty^q(s,p) \| + \epsilon(c) + Lr, \end{aligned}$$

where $\epsilon(c)$ is independent of $p \in K$, and $\epsilon(c) \to 0$ as $c \to \infty$. The last line follows from the fact that $q'(t) \in B(q,r),\ \forall t \in [0,T]$. Now we have

$$\| \eta_c^{q'(t)}(s,p) - \eta_\infty^q(s,p) \| \leq \epsilon(c)T + LrT + L \int_0^t \| \eta_c^{q'(t)}(s,p) - \eta_\infty^q(s,p) \| ds. \tag{7.20}$$

The claim now follows from the Gronwall inequality.

Lemma 55 *Let $q \in \mathrm{R}^{d_2}$ and let $\lambda_\infty(q)$ be the unique globally asymptotically stable equilibrium of (7.17). Given any $\epsilon > 0$ and $T > 0$, there exist $c_{\epsilon,T} > 0$, $\delta_{\epsilon,T} > 0$ and $r_{\epsilon,T} > 0$ such that $\forall t \in [0,T), \forall p \in B(\lambda_\infty(q), \delta_{\epsilon,T}),\ \forall c > c_{\epsilon,T}$ and external input $q'(s)$ with*

$$q'(s) \in B(q, r_{\epsilon,T}), s \in [0,T], \tag{7.21}$$

we have

$$\eta_c^{q'(t)}(t,p) \in B(\lambda_\infty(q), 2\epsilon), \forall t \in [0,T]. \tag{7.22}$$

Proof: Given any $\epsilon > 0$, it follows from Lyapunov stability that there exists $\delta_{\epsilon,T} > 0$ such that any trajectory of (7.17) starting in $B(\lambda_\infty(q), \delta_{\epsilon,T})$ stays within the ball $B(\lambda_\infty(q), \epsilon)$. Without loss of generality we can assume $\delta_{\epsilon,T} < \epsilon$. From Lemma 54 we know that there exist $c_{\epsilon,T}$ and $r_{\epsilon,T}$ such that

$$\| \eta_c^{q'(t)}(t,p) - \eta_\infty^q(t,p) \| \leq \epsilon. \tag{7.23}$$

Hence, $\forall t \in [0,T],\ \forall q'(t) \in B(q, r_{\epsilon,T}),\ \forall p \in B(\lambda_\infty(q), \delta_{\epsilon,T})$ and $\forall c > c_{\epsilon,T}$,

$$\begin{aligned} \| \eta_c^{q'(t)}(t,p) - \lambda_\infty(q) \| \leq & \| \eta_c^{q'(t)}(t,p) - \eta_\infty^q(t,p) \| + \| \eta_\infty^q(t,p) - \lambda_\infty(q) \| \\ & \leq \delta/2 + \epsilon \\ & \leq 2\epsilon. \end{aligned} \tag{7.24}$$

The claim follows.

Lemma 56 *Let $p \in B(0,1) \subset \mathrm{R}^{d_1}$, $q \in K' \subset \mathrm{R}^{d_2}$ and let $\lambda_\infty(q)$ be the unique globally asymptotically stable equilibrium of (7.17). Then, given $\epsilon > 0$, there exists $c_\epsilon \geq 1$, $r_\epsilon > 0$ and $T_\epsilon > 0$ such that for any external input $q'(s)$ satisfying*

$$q'(s) \in B(q, r_\epsilon), \forall s \in [0,T], \tag{7.25}$$

we have

$$\| \eta_c^{q'(t)}(t,p) - \lambda_\infty(q) \| \leq 2\epsilon, \forall t \geq T_\epsilon, \forall c > c_\epsilon. \tag{7.26}$$

Proof: Given any $\epsilon > 0$, it follows from Lyapunov stability that there exists $\delta > 0$ such that any trajectory of (7.17) starting in $B(\lambda_\infty(q), \delta)$ stays within the ball $B(\lambda_\infty(q), \epsilon)$. Without loss of generality we can assume $\delta < \epsilon$. Let $K = B(0,1) \cup B(\lambda_\infty(q), \delta)$, then from Lemma 52, there exists a $T_{\delta/2}$ such that $\eta_\infty^q(t,p) \in B(\lambda_\infty(q), \delta/2)$, $\forall t \geq T_{\delta/2}$. Pick $T_\epsilon \stackrel{\Delta}{=} T_{\delta/2}$ as given by Lemma 52 and divide the time-line into intervals T_ϵ apart, i.e., $t \in \bigcup_n [nT_\epsilon, (n+1)T_\epsilon), n \geq 0$. We know $\eta_\infty^q(t,p) \in B(\lambda_\infty(q), \delta/2), \forall t \geq T_\epsilon$. From Lemma 54, it follows that there exists $c^1_{\epsilon,T_\epsilon}$ and $r^1_{\epsilon,T_\epsilon}$ such that $\| \eta_c^{q'(t)}(T_\epsilon,p) - \eta_\infty^q(T_\epsilon,p) \| \leq \delta/2, \forall c > c^1_{\epsilon,T_\epsilon}$ and $q'(t)$ satisfying (7.25). This implies that $\eta_c^{q'(t)}(T_\epsilon,p) \in B(\lambda_\infty(q),\delta) \subset B(\lambda_\infty(q), 2\epsilon), \forall c > c^1_{\epsilon,T_\epsilon}$ and $q'(t)$ satisfying (7.25). Thus starting from p, the trajectory $\eta_c^{q'(t)}(t,p)$ falls into the ball $B(\lambda(\infty),\delta)$ for all $c > c^1_{\epsilon,T_\epsilon}$ and $q'(t)$ satisfying (7.25).

It is easy to note that the trajectory $\eta_c^{q'(t)}(t,p)$ of the ODE (7.16) in the time interval $[T_\epsilon, 2T_\epsilon)$ starting from p is the same as the trajectory of the ODE (7.16) in the time interval $[0, T_\epsilon]$ but starting from the initial condition $\eta_c^{q'(T_\epsilon)}(T_\epsilon,p)$. Now we know from Lemma 54 that $\eta_c^{q'(t)}(t, \eta_c^{q'(t)}(T_\epsilon,p))$ and $\eta_\infty^q(t, \eta_c^{q'(t)}(T_\epsilon,p))$ track each other closely in the time $t \in [0, T_\epsilon)$. Formally, $\forall t \in [T_\epsilon, 2T_\epsilon)$,

$$\begin{aligned}
&\| \eta_c^{q'(t)}(t,p) - \eta_\infty^q(t - T_\epsilon, \eta_c^{q'(t)}(T_\epsilon,p)) \| \\
=&\| \eta_c^{q'(t)}(t - T_\epsilon, \eta_c^{q'(t)}(T_\epsilon,p)) - \eta_\infty^q(t - T_\epsilon, \eta_c^{q'(t)}(T_\epsilon,p)) \| \\
\leq& \delta/2.
\end{aligned} \tag{7.27}$$

Since $\eta_\infty^q(T_\epsilon, \eta_c^{q'(t)}(T_\epsilon,p)) \in B(\lambda_\infty(q), \delta/2)$ and from (7.27) we can conclude that $\eta_c^{q'(t)}(2T_\epsilon,p) \in B(\lambda_\infty(q),\delta)$. Also, since $\eta_c^{q'(t)}(T_\epsilon,p) \in B(\lambda_\infty(q),\delta)$, we know from Lemma 55 that there exists $c^2_{\epsilon,T_\epsilon}$ and $r^2_{\epsilon,T_\epsilon}$ such that $\eta_c^{q'(t)}(t,p) \in B(\lambda_\infty(q), 2\epsilon), \forall t \in [T_\epsilon, 2T_\epsilon)$.
Notice that by arguing in a similar manner one can show that $\eta_c^{q'(t)}(nT_\epsilon,p) \in B(\lambda_\infty(q), \delta), \forall n \geq 0$ and $\eta_c^{q'(t)}(t,p) \in B(\lambda_\infty(q), 2\epsilon), \forall t \in [nT_\epsilon, (n+1)T_\epsilon)$. The proof is complete by choosing $c_\epsilon = \max(c^1_{\epsilon,T_\epsilon}, c^2_{\epsilon,T_\epsilon})$ and $r_\epsilon = \min(r^1_{\epsilon,T_\epsilon}, r^2_{\epsilon,T_\epsilon})$.

7.3 Stability of Two Timescale Stochastic Approximation Algorithms

Our goal is to study the stability of the iterates in the following coupled stochastic recursions (we repeat the same here for convenience):

$$x_{n+1} = x_n + a(n)[h(x_n, y_n) + M_{n+1}^{(1)}], \tag{7.28a}$$

$$y_{n+1} = y_n + b(n)[g(x_n, y_n) + M_{n+1}^{(2)}], \tag{7.28b}$$

where the iterates $x_n \in \mathbf{R}^{d_1}$ and $y_n \in \mathbf{R}^{d_2}$. As and when necessary we use $z_n \in \mathbf{R}^{d_1+d_2}$ to denote $z_n = (x_n, y_n)$, $n \geq 0$. We now state the conditions that will be seen to show stability of two timescale SA schemes.

Assumption 6

1. $h\colon \mathbf{R}^{d_1+d_2} \to \mathbf{R}^{d_1}$ *and* $g\colon \mathbf{R}^{d_1+d_2} \to \mathbf{R}^{d_2}$ *are Lipschitz continuous functions.*

2. $\{M_n^{(1)}\}$, $\{M_n^{(2)}\}$ *are martingale difference sequences w.r.t. the increasing sequence of* σ*-fields* $\{\mathcal{F}_n\}$*, where*

$$\mathcal{F}_n \stackrel{def}{=} \sigma(x_m, y_m, M_m^{(1)}, M_m^{(2)}, m \leq n), n \geq 0,$$

and such that

$$\mathbf{E}[\| M_{n+1}^{(i)} \|^2 \, |\mathcal{F}_n] \leq K(1+ \| x_n \|^2 + \| y_n \|^2), i = 1, 2, n \geq 0, \tag{7.29}$$

for some constant $K > 0$.

3. $\{a(n)\}$, $\{b(n)\}$ *are step-size schedules satisfying*

$$a(n) > 0,\ b(n) > 0,\ \sum_n a(n) = \sum_n b(n) = \infty, \sum_n (a(n)^2 + b(n)^2) < \infty,$$
$$\frac{b(n)}{a(n)} \to 0 \text{ as } n \to \infty. \tag{7.30}$$

4. *The functions* $h_c(x,y) \stackrel{def}{=} \frac{h(cx,cy)}{c}$, $c > 1$, *satisfy* $h_c \to h_\infty$ *as* $c \to \infty$, *uniformly on compacts for some* h_∞. *Also, the ODE*

$$\dot{x}(t) = h_\infty(x(t), y) \tag{7.31}$$

has a unique globally asymptotically stable equilibrium $\lambda_\infty(y)$, *where* $\lambda_\infty : \mathbf{R}^{d_2} \to \mathbf{R}^{d_1}$, *is a Lipschitz map. Further* $\lambda_\infty(0) = 0$, *i.e., the ODE* $\dot{x}(t) = h_\infty(x(t), 0)$ *has the origin in* R^{d_1} *as its unique globally a.s.e.*

5. *The functions* $g_c(y) \stackrel{def}{=} \frac{g(c\lambda_\infty(y),cy)}{c}, c \geq 1$, *satisfy* $g_c \to g_\infty$ *as* $c \to \infty$, *uniformly on compacts for some* g_∞. *Also, theODE*

$$\dot{y}(t) = g_\infty(y(t)) \tag{7.32}$$

has the origin in R^{d_2} *as its unique globally asymptotically stable equilibrium.*

7.3.1 Faster Timescale Results

It is easy to see that in a two timescale scheme, the slower timescale iterates *appear* to be constant when *viewed* from the faster timescale, while the faster timescale iterates *appear* equilibrated when *viewed* from the slower timescale. It is straightforward to show that the faster timescale iterates are bounded for a fixed $y \in \mathrm{R}^{d_2}$. However, since y changes along the slower timescale, we need a bound on $\| x_n \|$ in terms of $\| y_n \|$. Conditions $1-4$ of Assumption 6 help us achieve such a result (Theorem 64 at the end of this section).

In all the statements in this section, we make use of the following definition of 'time points'.

Definition 57 *For $n = 1, 2, \ldots$ and $T > 0$,*

$$t(0) = 0, t(n) = \sum_{i=0}^{n-1} a(i),\ n \geq 1,$$

$$T_0 = 0, T_n = \min\{t(m) \colon t(m) \geq T_{n-1} + T\}.$$

Note that

$$T_n = t(m(n)), \textit{for suitable } m(n) \uparrow \infty \textit{ as } n \uparrow \infty. \tag{7.33}$$

We hasten to point out that the above variables in Definition 57 only hold for the arguments in this section. In the next section, corresponding to the slower timescale, we will re-define these variables suitably for the slower timescale.

We analyze the evolution of iterates on the faster timescale and to this end, we re-write (7.28) as a single faster timescale recursion in z_n, i.e.,

$$z_{n+1} = z_n + a(n)[f(z_n) + N_{n+1}], \tag{7.34}$$

where,

$$\begin{aligned} z_n =& (x_n, y_n), \\ f(z_n) =& \begin{pmatrix} h(x_n, y_n) \\ \frac{b(n)}{a(n)} g(x_n, y_n) \end{pmatrix}, \\ N_{n+1} =& \begin{pmatrix} M_{n+1}^{(1)} \\ \frac{b(n)}{a(n)} M_{n+1}^{(2)} \end{pmatrix}. \end{aligned}$$

We define the piecewise linear trajectory $\bar{z}(t) = (\bar{x}(t), \bar{y}(t))$ as follows:

Definition 58

$$\bar{z}(t) = z_n + (z_{n+1} - z_n)\frac{t - t(n)}{t(n+1) - t(n)}, t \in [t(n), t(n+1)].$$

Since the boundedness of $\bar{z}(t)$ is not known, we monitor its growth every T_n time instants

and rescale it to the unit ball around the origin in $\mathrm{R}^{d_1+d_2}$. The rescaled trajectory $\hat{z}(t)$ defined below is bounded and tracks the trajectory of a scaled ODE (Lemma 62).

Definition 59 *Define the piecewise continuous trajectory $\hat{z}(t), t \geq 0$, by*

$$\hat{z}(t) = \frac{\bar{z}(t)}{r(n)} \text{ for } t \in [T_n, T_{n+1}), \tag{7.35}$$

$$\text{where } r(n) \stackrel{def}{=} \max(r(n-1), \| \bar{z}(T_n) \|, 1), n \geq 1, r(0) = 1. \tag{7.36}$$

Also, we define $\hat{z}(T_{n+1}^-) \stackrel{def}{=} \frac{\bar{z}(T_{n+1})}{r(n)}$. This is the same as $\hat{z}(T_{n+1})$ if there is no jump at T_{n+1} and is equal to $\lim_{t \uparrow T_{n+1}} \hat{z}(t)$.

The scaled iterates for $m(n) \leq k \leq m(n+1) - 1$ are given by

$$\hat{x}_{m(n)} = \frac{x_{m(n)}}{r(n)}, \tag{7.37a}$$

$$\hat{y}_{m(n)} = \frac{y_{m(n)}}{r(n)}, \tag{7.37b}$$

$$\hat{x}_{k+1} = \hat{x}_k + a(k)[h_c(\hat{x}_k, \hat{y}_k) + \hat{M}_{k+1}^{(1)}], \tag{7.37c}$$

$$\hat{y}_{k+1} = \hat{y}_k + a(k)[\epsilon_k + \hat{M}_{k+1}^{(2)}], \tag{7.37d}$$

where $c = r(n)$, $\epsilon_k = \frac{b(k)}{a(k)}\left(\frac{g(c\hat{x}_k, c\hat{y}_k)}{c}\right)$, $\hat{M}_{k+1}^{(1)} = \frac{M_{k+1}^{(1)}}{r(n)}$, $\hat{M}_{k+1}^{(2)} = \frac{M_{k+1}^{(2)}}{r(n)}$. The idea behind the definition of $r(n)$ in (7.36) is to check every time interval roughly T apart and rescale the iterates only if $\hat{z}(t)$ goes outside of the unit ball.

Lemma 60 *The sequence $\{r(n)\}$ as defined in (7.36) is monotonically nondecreasing.*

Proof: Follows directly from the definition of $r(n)$ in (7.36). Lemmas $4-6$, Chapter 3 of [**?**] continue to hold for $\hat{z}(t)$. In particular, we re-state a part of Lemma 6, Chapter 3 of [**?**], since we require it in the proof of Theorem 64.

Lemma 61 *For $0 < k \leq m(n+1) - m(n)$, we have almost surely*

$$\| \hat{z}(t(m(n)+k)) \| \leq K_2, \tag{7.38}$$

for some $K_2 > 0$.

Proof: See (3.2.6) in the proof of Lemma 6, Chapter 3, of [**?**].

Lemma 62 *Let $z_n(t) = (x_n(t), y_n(t)),\ t \in [T_n, T_{n+1}]$, denote the trajectory of the following ODE:*

$$\dot{x}(t) = h_{r(n)}(x(t), y(t)), \tag{7.39a}$$

$$\dot{y}(t) = 0, \tag{7.39b}$$

with initial conditions $x_n(T_n) = \hat{x}(T_n)$ and $y_n(T_n) = \hat{y}(T_n)$. Then

$$\lim_{n\to\infty} \parallel \hat{z}(t) - z_n(t) \parallel = 0, a.s., \forall t \in [t(m(n)), t(m(n+1))). \tag{7.40}$$

Proof: Follows from Lemma 1, Chapter 6, [**?**].

Lemma 63 *Outside a set of zero probability, for n large, there exists a $C_1 > 0$ such that if*

$$\parallel \bar{x}(T_n) \parallel > C_1(1 + \parallel \bar{y}(T_n) \parallel), \tag{7.41}$$

then

$$\parallel \bar{x}(T_{n+1}) \parallel < \frac{3}{4} \parallel \bar{x}(T_n) \parallel . \tag{7.42}$$

Proof: Note that (7.41) implies that $r(n) \geq C_1$, $\parallel \hat{y}(T_n) \parallel < \frac{1}{C_1}$ and $\parallel \hat{x}(T_n) \parallel > \frac{1}{1+1/C_1}$. Let $\phi_\infty^{y(t)}(t, x)$ and $\phi_c^{y(t)}(t, x)$ be the solutions to the ODEs

$$\dot{x}(t) = h_\infty(x(t), y(t)) \text{ and} \tag{7.43}$$

$$\dot{x}(t) = h_c(x(t), y(t)), \tag{7.44}$$

respectively, with initial condition x in both. Let $y'(t - T_n) = y_n(t),\ \forall t \in [T_n, T_{n+1}]$. It is

easy to see that

$$x_n(t) = \phi^{y'(t)}(t - T_n, \hat{x}(T_n)), \forall t \in [T_n, T_{n+1}], \tag{7.45}$$

where $x_n(t)$ and $y_n(t)$ are as in Lemma 62.

We also know from Lemma 56 that there exists $r_{1/4}$, $c_{1/4}$, and $T_{1/4}$ such that $\| \phi_c^{y'(t)}(t, \hat{x}(T_n)) \| \leq \frac{1}{4}$, $\forall t \geq T_{1/4}$, $\forall c \geq c_{1/4}$, whenever $y'(t) \in B(0, r_{1/4}), \forall t \in [0, T]$.

Now, let us pick $C_1 > \max(c_{1/4}, \frac{2}{r_{1/4}})$ and T in Definition 57 as $T \stackrel{def}{=} T_{1/4}$. Since $y'(t - T_n) = y_n(t) = \hat{y}(T_n), \forall t \in [T_n, T_{n+1}]$, we have for our choice of C_1, $y'(s) \in B(0, r_{1/4}), \forall s \in [0, T]$. We also know from Lemma 62 that $\| \hat{x}(T_{n+1}^-) - x_n(T_{n+1}) \| < \frac{1}{4}$ for sufficiently large n. Since $\| x_n(T_{n+1}) \| = \| \phi^{y'(t)}(T_{n+1} - T_n, \hat{x}(T_n)) \| \leq 1/4$, we have $\| \hat{x}(T_{n+1}^-) \| \leq \| \hat{x}(T_{n+1}^-) - x_n(T_{n+1}) \| + \| x_n(T_{n+1}) \| \leq \frac{1}{2}$. Since $\frac{\bar{x}(T_{n+1})}{\bar{x}(T_n)} = \frac{\hat{x}(T_{n+1}^-)}{\hat{x}(T_n)}$, it follows that $\| \bar{x}(T_{n+1}) \| < \frac{1+1/C_1}{2} \| \bar{x}(T_n) \|$, and the result holds by assuming without loss of generality that $C_1 > \max(c_{1/4}, \frac{2}{r_{1/4}}) > 2$.

Corollary 8 *$\| \bar{x}(T_n) \| \leq C^*(1 + \| \bar{y}(T_n) \|)$ almost surely, for some $C^* > 0$.*

Proof: On a set of positive probability, let us assume on the contrary that there exists a monotonically increasing sequence $\{n_k\}$ for which $C_{n_k} \uparrow \infty$ as $k \to \infty$ and $\| \bar{x}(T_{n_k}) \| \geq C_{n_k}(1 + \| \bar{y}(T_{n_k}) \|)$. Now from Lemma 63, we know that if $\| \bar{x}(T_n) \| > C_1(1 + \| \bar{y}(T_n) \|)$, then $\| \bar{x}(T_k) \|$ for $k \geq n$ falls at an exponential rate until it is within the ball of radius $C_1(1 + \| \bar{y}(T_k) \|)$. Thus corresponding to the sequence $\{n_k\}$, there must exist another sequence $\{n'_k\}$ such that $n_{k-1} \leq n'_k \leq n_k$ and $\| \bar{x}(T_{n'_k - 1}) \|$ is within the ball of radius $C_1(1 + \| \bar{y}(T_{n'_k - 1}) \|)$ but $\| \bar{x}(T_{n'_k}) \|$ is greater than $C_{n_k}(1 + \| \bar{y}(T_{n'_k}) \|)$. However, from Lemma 61 we know that the iterates can grow only by a factor of K_2 between $m(n'_k - 1)$ and $m(n'_k)$. This leads to a contradiction. So we conclude that $\| \bar{x}(T_n) \| \leq C^*(1 + \| \bar{y}(T_n) \|)$ for some $C^* > 0$.

Theorem 64 *We have $\| x_n \| \leq K^*(1 + \| y_n \|)$ almost surely for some $K^* > 0$.*

Proof: From Corollary 8, we know that $\| \bar{x}(T_n) \| \leq C^*(1 + \| \bar{y}(T_n) \|)$. From Lemma 61, we know that $\| \bar{z}(t) \| \leq K_2 \| \bar{z}(T_n) \|$, $\forall t \in [T_n, T_{n+1})$. The result follows by choosing $K^* = K_2 C^*$. Note, henceforth the quantity K^* is to be understood as in Theorem 64.

Theorem 65 *Given any $\epsilon > 0$ and $y \in \mathrm{R}^{d_2}$, define the set $A^\epsilon(y) \subset \mathrm{R}^{d_1}$ as $A^\epsilon(y) \stackrel{def}{=} \{x\colon \parallel x - \lambda_\infty(y) \parallel < \epsilon\}$. For any given $\epsilon > 0$, there exists $c_\epsilon > 0$ such that, if $r(n) > c_\epsilon$ for some n, then it follows that $(\hat{x}_k, \hat{y}_k) \in (A^\epsilon(\hat{y}_k), \hat{y}_k), \forall n \geq k$.*

Proof: The proof follows by a repeated application of Lemma 62 and Lemma 56 to the intervals $[T_k, T_{k+1}], \forall k \geq n$ and using the fact that $r(k) \geq r(n), \forall k \geq n$ from Lemma 60.

7.3.2 Slower Timescale Analysis

We now use Theorem 64 of section 7.3.1 to prove the stability of the slower timescale iterates. The 'time points' in Definition 57 are no longer valid here and we re-define these suitably below.

Definition 66 *For $n = 1, 2, \ldots$ and $T > 0$,*

$$t(0) = 0, t(n) = \sum_{i=0}^{n-1} b(i),$$

$$T_0 = 0, T_n = \min\{t(m)\colon t(m) \geq T_{n-1} + T\}.$$

Note that

$$T_n = t(m(n)), \textit{for suitable } m(n) \uparrow \infty \textit{ as } n \uparrow \infty. \tag{7.46}$$

Note that $a(i)$ in Definition 57 are now replaced by $b(i)$ (see Definition 66). We define the interpolated trajectory $\bar{z}(t) = (\bar{x}(t), \bar{y}(t))$ on the slower timescale as follows:

Definition 67

$$\bar{z}(t) = z_n + (z_{n+1} - z_n)\frac{t - t(n)}{t(n+1) - t(n)}, t \in [t(n), t(n+1)].$$

Also, we define $\hat{z}(T_{n+1}^-) \stackrel{def}{=} \frac{\bar{z}(T_{n+1})}{r(n)}$. This is the same as $\hat{z}(T_{n+1})$ if there is no jump at T_{n+1} and is equal to $\lim_{t \uparrow T_{n+1}} \hat{z}(t)$.

As with the case of the faster timescale, we keep the growth of the trajectory $\bar{z}(t)$ under check, by monitoring it roughly every T instants and then normalizing to the unit ball in $\mathbf{R}^{d_1+d_2}$ as described below.

Definition 68 *Define the piecewise continuous trajectory* $\hat{z}(t) = (\hat{x}(t), \hat{y}(t)), t \geq 0$, *by*

$$\hat{z}(t) = \frac{\bar{z}(t)}{r(n)} \text{ for } t \in [T_n, T_{n+1}), \tag{7.47}$$

$$\text{where } r(n) \stackrel{def}{=} \max(r(n-1), \| \bar{z}(T_n) \|, 1), n \geq 1, r(0) = 1. \tag{7.48}$$

$$\tag{7.49}$$

The scaled iterates for $m(n) \leq k < m(n+1) - 1$ are given by

$$\hat{x}_{m(n)} = \frac{x_{m(n)}}{r(n)}, \tag{7.50a}$$

$$\hat{y}_{m(n)} = \frac{y_{m(n)}}{r(n)}, \tag{7.50b}$$

$$\hat{x}_{k+1} = \hat{x}_k + a(k)[h_c(\hat{x}_k, \hat{y}_k) + \hat{M}^{(1)}_{k+1}], \tag{7.50c}$$

$$\hat{y}_{k+1} = \hat{y}_k + b(k)[g'_c(\hat{x}_k, \hat{y}_k) + \hat{M}^{(2)}_{k+1}], \tag{7.50d}$$

where $c = r(n)$, $g'_c \stackrel{def}{=} \frac{g(cx,cy)}{c}$, $\hat{M}^{(1)}_{k+1} = \frac{M^{(1)}_{k+1}}{r(n)}$ and $\hat{M}^{(2)}_{k+1} = \frac{M^{(2)}_{k+1}}{r(n)}$.

Lemma 69 $\sup_t \mathbf{E} \| \hat{y}(t) \|^2 < \infty$.

Proof: Follows from Lemma 4, Chapter 3 of [**?**] upon using the fact that $\| \hat{x}(t) \| \leq K^*(1 + \| \hat{y}(t) \|)$ from Theorem 64.

Lemma 70 *For* $0 < k \leq m(n+1) - m(n)$, *we have*

$$\| \hat{y}(t(m(n) + k)) \| \leq K_3, \tag{7.51}$$

for some $K_3 > 0$. *Also, there exists a* $B > 0$ *such that* $\| \hat{x}(t(m(n) + k)) \| < B$.

Proof: The claim (7.51) can be shown in a manner similar to the proof of Lemma 6, Chapter 3, [**?**]. The proof is complete by choosing $B = K^*(1 + K_3)$.

Lemma 71 *Pick any* $\epsilon > 0$ *and let* $y_n(t)$ *be the trajectory to the ODE* $\dot{y}(t) = g_c(y(t)), t \in [T_n, T_{n+1})$, *with* $y_n(T_n) = \hat{y}(T_n)$. *Then there exists a* $c_\epsilon > 0$ *such that If* $r(k) > c_\epsilon$ *for*

some $k > 0$, then for sufficiently large n we have

$$\sup_{t\in[T_n,T_{n+1})} \| \hat{y}(t) - y_n(t) \| \leq \epsilon LTe^{L(L+1)T}, a.s. \tag{7.52}$$

Proof: Follows in the same manner as Lemma 1, Chapter 2, [**?**] (see Appendix for a sketch of the proof).

Lemma 72 *Let K^* be as in Theorem 64, then it follows that $\| \hat{y}(T_n) \| \geq \frac{1}{K^*+2}$ for sufficiently large $\| \bar{y}(T_n) \|$.*

Proof: From Theorem 64, we know that

$$\| r(n) \| \leq \| \bar{y}(T_n) \| + K^*(1+ \| \bar{y}(T_n) \|).$$

Also,

$$\begin{aligned} \| \hat{y}(T_n) \| &= \frac{\| \bar{y}(T_n) \|}{r(n)} \\ &\geq \frac{\| \bar{y}(T_n) \|}{\| \bar{y}(T_n) \| + K^*(1+ \| \bar{y}(T_n) \|)} \\ &= \frac{1}{1 + \frac{K^*}{\|\bar{y}(T_n)\|} + K^*}. \end{aligned} \tag{7.53}$$

The claim follows for any $\| \bar{y}(T_n) \| > K^*$.

Let $g_c \colon \mathrm{R}^{d_2} \to \mathrm{R}^{d_2}$ and $g_\infty \colon \mathrm{R}^{d_2} \to \mathrm{R}^{d_2}$ be functions as defined in condition 5 of Assumption 6, and let $\chi_\infty(t, y)$ denote the solution to the ODE

$$\dot{y}(t) = g_\infty(y(t)), \tag{7.54}$$

with initial condition y, and let $\chi_c(t, y)$ denote the solution to the ODE

$$\dot{y}(t) = g_c(y(t)), \tag{7.55}$$

with initial condition y.

Note that the ODEs in (7.54) and (7.55) are degenerate versions of the ODE in (7.16)

and (7.17) in that there is no external input. However, results of Section 7.2 continue to hold for (7.54) and (7.55) as well with $q = 0$.

Lemma 73 *For n large there exists a $C > 0$ such that if*

$$\| \bar{y}(T_n) \| > C, \tag{7.56}$$

then

$$\| \bar{y}(T_{n+1}) \| < \frac{1}{2} \| \bar{y}(T_n) \| . \tag{7.57}$$

Proof: We make use of condition 5 of Assumption 6 that $\mathbf{0} \in \mathrm{R}^{d_2}$ is the unique globally asymptotically stable equilibrium of (7.54), and as a consequence of Lemma 56 there exist $c_{1/4}$ and $T_{1/4}$ such that $\| \chi_c(t, y) \| < \frac{1}{4(K^*+2)}, \forall t \geq T_{1/4}, c > c_{1/4}$.
Also, if $\| \bar{y}(T_n) \| > K^*$, it follows from Lemma 72 that $\| \hat{y}(T_n) \| \geq \frac{1}{K^*+2}$. We know from Lemma 71 that for sufficiently large n, there exists $C_1 > 0$ such that $\| \hat{y}(T^-_{n+1}) - y_n(T_{n+1}) \| < \frac{1}{4(K^*+2)}$, for $r(n) > C_1$. Now, let us pick $C = \max(c_{1/4}, C_1, K^*)$ and $T = T_{1/4}$. Then for sufficiently large n it follows that $\| \hat{y}(T^-_{n+1}) \| \leq \| \hat{y}(T^-_{n+1}) - y_n(T_{n+1}) \| + \| y_n(T_{n+1}) \| \leq \frac{1}{2(K^*+2)}$. Since $\frac{\bar{y}(T_{n+1})}{\bar{y}(T_n)} = \frac{\hat{y}(T^-_{n+1})}{\hat{y}(T_n)}$, it follows that $\| \bar{y}(T_{n+1}) \| < \frac{1}{2} \| \bar{y}(T_n) \|$.

Corollary 9 $\| \bar{y}(T_n) \| \leq C'$ *a.s., for some $C' > 0$.*

Proof: Let us assume on the contrary that on a set of positive probability, there exists a sequence $\{n_k\}$ such that $C_{n_k} \uparrow \infty$ as $k \to \infty$ and $\| \bar{y}(T_{n_k}) \| \geq C_{n_k}$. From Lemma 73 we know that if $\| \bar{y}(T_n) \| > C$, then $\| \bar{y}(T_k) \|$ for $k \geq n$ falls at an exponential rate to the ball of radius C. Thus corresponding to the sequence $\{n_k\}$, there exists another sequence $\{n'_k\}$ such that $n_{k-1} \leq n'_k \leq n_k$ and $\| \bar{y}(T_{n'_k - 1}) \|$ is within the ball of radius C but jumps outside this ball of radius C (i.e.,$\| \bar{y}(T_{n'_k}) \| > C$) to points which are at a distance greater than C_{n_k} from the origin. However, from Lemma 70 we know that the iterates can grow only by a factor of K_3 between $m(n'_k - 1)$ and $m(n'_k)$. This leads to a contradiction and the claim follows.

Theorem 74 *Under Assumptions 6, we have* $\sup_n \| y_n \| < \infty$*, almost surely.*

Proof: From Corollary 9, we know that $\| \bar{y}(T_n) \| \leq C'$. From Lemma 70, we know that $\| \bar{y}(t) \| \leq K_3 \| \bar{y}(T_n) \|, \ \forall t \in [T_n, T_{n+1})$. The result follows by noting that $\| y_n \| \leq K_3 C'$ almost surely.

7.4 An Application in Reinforcement Learning

Reinforcement Learning (RL) algorithms are sample trajectory based methods for solving Markov Decision Processes (MDP). RL algorithms make use of stochastic approximation to *filter* the noise in the estimates obtained from the samples. We apply our analysis to establish the stability of iterates in an actor-critic (AC) algorithm (Algorithm 1 of [**?**]), which is a two-timescale stochastic approximation scheme. Our goal here is to demonstrate the application of our stability results to Algorithm 1 of [**?**]. Hence, we present only limited background material and refer the reader to [**?**] for the detailed development of the algorithm. We consider an MDP with finite numbers of states and actions. The state space is denoted by $S = \{1, 2, \ldots, n\}$, and the action space by $A = \{1, 2, \ldots, m\}$. In state s and under action a, let $c_a(s)$ denote the single-stage cost incurred and $p_a(s, s')$ be the probability of transition to s'. By policy, we mean a sequence $\pi = \{\pi_1, \pi_2, \ldots, \pi_k, \ldots\}$, where each $\pi_k, k = 1, 2, \ldots$, specifies a way by which states are mapped to actions and such mapping can be *deterministic* or *randomized*. In deterministic policies, $\pi_k \colon S \to A$ while, in randomized policies, $\pi_k(s, a)$ is the probability of performing action a in state s. When $\pi_k = \pi, \forall k$, we call the policy *stationary*. In the following example, we consider classes of stationary randomized policies (SRP) denoted by Π. Under an SRP, the MDP is a Markov chain with transition probability kernel denoted by P_π.

The *average*-cost associated with a policy π is given by

$$J(\pi) = \lim_{N \to \infty} \frac{1}{N} \mathbf{E}[\sum_{n=0}^{N-1} c_{a_n}(s_n) | \pi], \tag{7.58}$$

where a_n is sampled from the distribution $\pi(s_n, \cdot), \forall n \geq 0$ and given s_n, s_{n+1} is distributed as $p_{a_n}(s_n, \cdot)$. The differential cost associated with state-action pair (s, a) under policy π

is denoted by $Q^\pi(s,a)$, where

$$Q^\pi(s,a) = \mathbf{E}[\sum_{n=0}^{\infty} c_{a_n}(s_n) - J(\pi)]. \tag{7.59}$$

Similarly, one can define the differential cost associated with state s, denoted $V^\pi(s)$, as

$$V^\pi(s) = \sum_{a\in A} \pi(s,a) Q^\pi(s,a). \tag{7.60}$$

The average and differential costs are related by the Bellman equation as below:

$$Q^\pi(s,a) = g^\pi(s) - J(\pi) + \sum_a p_a(s,s') V^\pi(s'). \tag{7.61}$$

We are interested in finding the *optimal*-policy π^* such that

$$J(\pi^*) = \min_{\pi\in\Pi} J(\pi). \tag{7.62}$$

The term *curse of dimensionality* or simply *curse* refers to the exponential dependence of the number of states on the number of state variables. Most practical problems suffer from the *curse* and (7.62) is difficult to solve due to the large search space. So, in practice, we restrict the search space to a smaller set of SRPs wherein each policy π is a differentiable function of a parameter $\theta \in \mathrm{R}^d$, where $d << n \times m$. We are interested in finding the *optimal*-parameter θ^* such that

$$J(\pi^{\theta^*}) = \min_{\theta\in\mathrm{R}^d} J(\pi^\theta). \tag{7.63}$$

Henceforth, by abuse of notation, we use the symbols π and θ interchangeably to denote the policy π^θ.

The gradient of $J(\theta)$ is given by

$$\nabla J(\theta) = \sum_s d^\pi(s) \sum_a \nabla\pi(s,a) Q^\pi(s,a), \tag{7.64}$$

where $(d^{\pi}(s), s \in S)$ denotes the stationary distribution under the SRP π.

We now verify our stability conditions for the iterates in Algorithm 1 of [**?**] (except for some differences in the aforementioned algorithm and the one we present below). We present the algorithm here for the sake of completeness.

Algorithm 4 The Actor-Critic Algorithm

1: Input:

- Randomized parameterized policy $\pi^{\theta}(\cdot, \cdot)$,
- Value function feature vector.

2: Initialization:

- Policy parameters $\theta_0 = \theta$,
- Value function weight vector $v_0 = v$,
- Initial step sizes $a(0) = a, b(0) = b, c(0) = c$,
- Initial state s_0.

3: **for** $t = 0, 1, 2, \ldots$ **do**

4: Execution

- Draw action $a_n \sim \pi^{\theta_n}(s_n, \cdot)$,
- Observe next state $s_{n+1} \sim p_{a_n}(s_n, \cdot)$,
- Observe cost $c_{a_n}(s_n)$.

5: Average Cost Update: $\hat{J}_{n+1} = \hat{J}_n + a(n)(c_{a_n}(s_n) - \hat{J}_n)$

6: TD Error: $\delta_n = c_{a_n}(s_n) - \hat{J}_n + v_n^{\top} f_{s_{n+1}} - v_n^{\top} f_{s_n}$

7: Critic Update: $v_{n+1} = v_n + a(n)\delta_n f_{s_n}$

8: Actor Update: $\theta_{n+1} = \theta_n - b(n)(\delta_n \psi_{s_n a_n} + \epsilon\theta_n)$

9: **end for**

10: Return the policy and value function parameters θ and v.

Remark 2 *We now hasten to point out certain minor differences between Algorithm* 1 *of [***?** *] and the Algorithm 4 presented here. Algorithm* 1 *of [***?** *] corresponds to the average reward setting while the algorithm here is for the setting of average cost. Also, the update in equation* (23) *of Algorithm* 1 *of [***?** *] makes use of a projection operator in order to ensure boundedness of the iterates. However, in the actor update in line 8 of Algorithm 4 (here), we do not make use of projection and instead we introduce an additional term* $\epsilon\theta_n$ *(where* $\epsilon > 0$ *is a small positive constant). This is equivalent to adding a quadratic penalty to the objective. Thus, the actor update in line 8 of Algorithm 4 corresponds to the regularized version of the minimization problem in* (7.63) *given by*

$$J(\pi^{\theta^*}) = \min_{\theta \in \mathrm{R}^d} J(\pi^{\theta}) + \epsilon\theta. \tag{7.65}$$

Remark 3 *In Algorithm 4, the randomized policy* π^{θ} *is given via the Gibbs distribution below:*

$$\pi^{\theta}(s,a) = \frac{e^{\phi_{sa}^{\top}\theta}}{\sum_{a'} e^{\phi_{sa'}^{\top}\theta}}, \tag{7.66}$$

where $\phi_{sa} \in \mathrm{R}^d$ *is a d-dimensional feature vector, and* $\theta \in \mathrm{R}^d$ *is the parameter. This policy parameterization has also been suggested in [***?** *].*

We also assume that assumptions $A1 - A3$ of [**?**] hold. The iterates in Algorithm 4 form a two-timescale stochastic approximation scheme. While iterates $\{\hat{J}_n\}$ and $\{v_n\}$ evolve on the faster timescale, the iterates $\{\theta_n\}$ evolve along the slower timescale. We now write the iterates in the standard form as below:

$$\hat{J}_{n+1} = \hat{J}_n + a(n)[h^1(\hat{J}_n, v_n, \theta_n) + M_{n+1}^{(1)}], \tag{7.67}$$

$$v_{n+1} = v_n + a(n)[h^2(\hat{J}_n, v_n, \theta_n) + M_{n+1}^{(2)}], \tag{7.68}$$

$$\theta_{n+1} = \theta_n + b(n)[g(\hat{J}_n, v_n, \theta_n) + M_{n+1}^{(3)}], \tag{7.69}$$

where[2]

$$h^1(\hat{J}_n, v_n, \theta_n) = \mathbf{E}[c_{a_n}(s_n)|\mathcal{F}_n] - \hat{J}_n = \sum_s d^\pi(s) \sum_a \pi(s,a) c_a(s) - \hat{J}_n, \tag{7.70}$$

$$\begin{aligned} h^2(\hat{J}_n, v_n, \theta_n) &= \mathbf{E}[\delta_n f_{s_n}|\mathcal{F}_n] \\ &= \sum_s d^\pi(s) \sum_a \pi(s,a)[c_a(s) - \hat{J}_n + \sum_{s'} p_a(s,s') v_n^\top f_{s'} - v_n^\top f_s] f_s, \end{aligned} \tag{7.71}$$

$$g(\hat{J}_n, v_n, \theta_n) = \mathbf{E}[\delta_n \psi_{s_n a_n}|\mathcal{F}_n] = -\sum_s d^\pi(s) \sum_a \nabla \pi(s,a) \hat{A}^\pi(s,a) - \epsilon\theta_n. \tag{7.72}$$

Note that, in the above equations, π stands for π^{θ_n} and
$\mathcal{F}_n \stackrel{def}{=} \sigma(\theta_m, v_m, \hat{J}_m, M_m^{(1)}, M_m^{(2)}, M_m^{(3)}, 0 \leq m \leq n)$. To handle the *curse*, the differential cost $V^\pi(s)$ is approximated as $V^\pi(s) \approx v^{\pi\top} f_s$ (where $f_s \in \mathrm{R}^k$ is the feature vector of state s, $v^\pi \in \mathrm{R}^k$ is a learnt weight vector) and $\hat{A}^\pi(s,a) = c_a(s) - \hat{J}_n + \sum_{s'} p_a(s,s') v_n^\top f_{s'} - v_n^\top f_s$ is the approximate *advantage*-function (see [**?**]). The martingale terms are given by

$$M_{n+1}^{(1)} = c_{a_n}(s_n) - \mathbf{E}[c_{a_n}(s_n)|\mathcal{F}_n], \tag{7.73}$$

$$M_{n+1}^{(2)} = \delta_n f_{s_n} - \mathbf{E}[\delta_n f_{s_n}|\mathcal{F}_n], \tag{7.74}$$

$$M_{n+1}^{(3)} = \delta_n \psi_{s_n a_n} - \mathbf{E}[\delta_n \psi_{s_n a_n}|\mathcal{F}_n]. \tag{7.75}$$

It is easy to see that there exist $C_i, i = 1, 2, 3$ such that $\mathbf{E}[||M_{n+1}^{(i)}||^2|\mathcal{F}_n] \leq C_i(1 + ||\theta_n||^2 + ||v_n||^2 + ||\hat{J}_n||^2)$. By letting $x = (\hat{J}, v) \in \mathrm{R}^{1+k}$ and $y = \theta \in \mathrm{R}^d$, we can rewrite the iterates in (7.67)-(7.69) in the standard format of (7.28) as below:

$$x_{n+1} = x_n + a(n)[h(x_n, y_n) + N_{n+1}^{(1)}], \tag{7.76a}$$

$$y_{n+1} = y_n + b(n)[g(x_n, y_n) + N_{n+1}^{(2)}], \tag{7.76b}$$

where $h = (h^1, h^2)$ (see (7.70), (7.71)), g is the same as in (7.72), $N_{n+1}^{(1)} = (M_{n+1}^{(1)}, M_{n+1}^{(2)})$ and $N_{n+1}^{(2)} = M_{n+1}^{(3)}$.

[2]For simplicity, we make here the mild technical assumption that given θ_n, the states s_n are independently sampled from the stationary distribution $d^{\pi^{\theta_n}}$ of the Markov chain, a scenario also discussed in Chapter 6 of [**?**].

We show that Assumption 6 holds for the iterates in Algorithm 4 in the proposition below.

Proposition 75

1. *The functions* $h^1 \colon \mathrm{R}^{d+1+k} \to \mathrm{R}, h^2 \colon \mathrm{R}^{d+1+k} \to \mathrm{R}^k, g \colon \mathrm{R}^{d+1+k} \to \mathrm{R}^d$ *are Lipschitz.*

2. *The functions* $h^1_c(\hat{J}, v, \theta) \stackrel{def}{=} \frac{h^1(c\hat{J}, cv, c\theta)}{c}, c \geq 1$ *are Lipschitz, and satisfy* $h^1_c \to h^1_\infty$*, as* $c \to \infty$*, uniformly on compacts.*

3. *The functions* $h^2_c(\hat{J}, v, \theta) \stackrel{def}{=} \frac{h^2(c\hat{J}, cv, c\theta)}{c}$ *are Lipschitz, and satisfy* $h^2_c \to h^2_\infty$ *as* $c \to \infty$*, uniformly on compacts.*

4. *The ODE* $(\dot{\hat{J}}(t), \dot{v}(t)) = h_\infty(\hat{J}(t), v(t), \theta)$ *has a unique asymptotically stable equilibrium* $\lambda_\infty(\theta)$*, where* $\lambda_\infty \colon \mathrm{R}^d \to \mathrm{R}^{1+k}$ *is a Lipschitz map that satisfies* $\lambda_\infty(\mathbf{0}) = \mathbf{0}$*.*

5. *The functions* $g_c(\theta) \stackrel{def}{=} \frac{g(c\theta, c\lambda_\infty(\theta))}{c}$ *are Lipschitz and satisfy* $g_c \to g_\infty$*, as* $c \to \infty$*, uniformly on compacts and the ODE* $\dot{\theta} = g_\infty(\theta(t))$ *has the origin in* R^k *as its unique globally asymptotically stable equilibrium.*

Proof:

1. $h^1(\hat{J}, v, \theta) = \sum_s d^\pi(s) \sum_a \pi(s,a) c_a(s) - \hat{J}$. h^1 is clearly Lipschitz (in fact linear) in $\hat{J}$. We now show that the derivatives of h^1 are bounded w.r.t. θ.

$$\nabla_\theta h^1(\hat{J}, v, \theta) = \sum_s \nabla_\theta d^\pi(s) \sum_a \pi(s,a) c_a(s) + \sum_s d^\pi(s) \sum_a \nabla_\theta \pi(s,a) c_a(s). \quad (7.77)$$

We know from [**?**] that $d^\pi(s)$ is continuously differentiable in θ and has a bounded derivative. Now since

$$d^\pi(s) = \sum_{s'} d^\pi(s') \sum_a \pi(s', a) \sum_s p_a(s', s),$$

and $\nabla \pi_\theta(s,a) = \pi(s,a)(\phi_{sa} - \sum_{a'} \phi(s,a')\pi(s,a'))$, it follows that $d^\pi(s)$ also has a bounded derivative w.r.t θ. Using similar arguments on the boundedness of the

derivatives of $d^\pi(s)$ and $\pi(s,a)$ w.r.t θ we can show that h^2 and g are Lipschitz as well. For more details see [**?**].

Given θ, define quantity $\pi_c(\theta) \stackrel{def}{=} \pi^{c\theta}, c \geq 1$. For any s, let $a^* = \arg\max_a \phi_{sa}^\top \theta$, and let $\pi_\infty^\theta(s,a) \stackrel{def}{=} \mathbf{1}_{\{a=a^*\}}$, where $\mathbf{1}$ is the indicator function. We show that $\pi_c^\theta(s,a) \to \pi_\infty^\theta(s,a)$, uniformly on compacts. For any $a \neq a^*$,

$$\begin{aligned}\pi_c^\theta(s,a) &= \frac{e^{c\phi_{sa}^\top\theta}}{\sum_{a'} e^{c\phi_{sa'}^\top\theta}} \\ &\leq \frac{e^{c\phi_{sa}^\top\theta}}{e^{c\phi_{sa^*}^\top\theta}} \\ &= e^{c(\phi_{sa}^\top - \phi_{sa^*}^\top)\theta}.\end{aligned}$$

Now since $a \neq a^*$ and $\pi_c^\theta(s,a^*) = 1 - \sum_{a\neq a^*} \pi_c^\theta(s,a)$, we have $\pi_c^\theta(s,a) \to 0$, as $c \to \infty$, $\forall s \in S, a \in A$, uniformly on compacts. We now drop the superscript θ and simply use π_∞ and π_c for notational simplicity.

2. $h_c^1(\hat{J}, v, \theta) = \frac{\sum_s d^{\pi_c}(s) \sum_a \pi_c(s,a) c_a(s) - c\hat{J}}{c}$, thus we have $h_\infty^1(\hat{J}, v, \theta) = -\hat{J}$.

3. In a similar fashion,

$$\begin{aligned}h_c^2(\hat{J}, v, \theta) \stackrel{def}{=} \Bigg(\sum_s d^{\pi_c}(s) \sum_a \pi_c(s,a)[c_a(s) - c\hat{J} \\ + \sum_{s'} p_a(s,s') v_n^\top f_{s'} - v_n^\top f_s] \Bigg)/c,\end{aligned}$$

and $h_\infty^2(\hat{J}, v, \theta) = F^\top D^{\pi_\infty}(-I + P_\pi)Fv - \hat{J}$, where F is the feature matrix whose s^{th} row is f_s, and D^π is a diagonal matrix where the s^{th} diagonal element is $d^\pi(s)$. Also, we make the assumption that F has full column rank and $Fe \neq e$, where $e = (1, 1, \ldots, 1)$. This is a technical requirement, see also [**? ?**].

4. Consider the ODE $(\dot{\hat{J}}(t), \dot{v}(t)) = h_\infty(\hat{J}, v, \theta)$, where $h_\infty = (h_\infty^1, h_\infty^2)$. It is straightforward to see that $\dot{\hat{J}}(t) = -\hat{J}(t)$ is stable to the origin in R. Now, we know from [**?**] (provided *A*3 of [**?**] holds) that $\dot{v}(t) = F^\top D^{\pi_\infty}(-I + P_\pi)Fv$ has the origin in R^d as

its unique asymptotically stable equilibrium. Thus, $\lambda(\theta) = (0, \mathbf{0}) \in \mathrm{R}^{1+k}, \forall \theta \in \mathrm{R}^k$, where $0 \in \mathrm{R}$ and $\mathbf{0} \in \mathrm{R}^k$.

5. Now

$$g_c(\theta) = \Bigg(\sum_s d^{\pi_c}(s) \sum_a \nabla \pi_c(s,a)[c_a(s) - c \times 0 + \sum_{s'} p_a(s,s') c \, \mathbf{0}^\top f_{s'} - c \, \mathbf{0}^\top f_s] \Bigg) / c - \frac{\epsilon c \theta}{c}, \tag{7.78}$$

and $g_\infty(\theta) = \lim_{c \to \infty} \frac{g(c\theta)}{c}$. Note that in (7.78) we have made use of the fact that $\lambda_\infty(\theta) = (0, \mathbf{0})$. Also, it is easy to see that all the quantities in the numerator of the first term on the right hand side of (7.78) are bounded and therefore, $g_\infty(\theta) = -\theta$. Then the ODE $\dot{\theta}(t) = -\theta(t)$, has the origin in R^d as its unique globally asymptotically stable equilibrium.

The claim follows.

We have thus shown that the iterates θ_n, v_n and $\hat{J}_n$ are stable. It is important to note that for the stability analysis we have not explicitly projected the iterates θ_n as in [**?**].

7.5 Conclusions

In this paper, we presented and proved conditions that guarantee the stability of two-timescale stochastic approximation algorithms. This problem had not been addressed previously in the literature. The sufficient conditions turn out to be weaker than those needed to establish convergence (Chapter 6 of [**?**]). Our stability analysis of two-timescale stochastic approximation is quite general and may easily be extended to the case of multi-timescale stochastic approximation, where the number of timescales is more than two.

7.6 Appendix

7.6.1 Gronwall Inequalities

Discrete Gronwall Inequality

Lemma 76 *Let* $\{x_n, n \geq 0\}$ *(resp.*$\{a_n, n \geq 0\}$*) be non-negative (resp. positive) sequences and* $C, L \geq 0$ *scalars such that for all* n*,*

$$x_{n+1} \leq C + L(\sum_{n}^{n} a_m x_m) \tag{7.79}$$

Then for $T_n = \sum_{m=0}^{n} a_m$*,*

$$x_{n+1} \leq Ce^{LT_n}.$$

Continuous Gronwall Inequality

Lemma 77 *For continuous* $u(\cdot), v(\cdot) \geq 0$ *and scalars* $0 < C, K < T$*, we have*

$$u(t) \leq C + K \int_0^t u(s)v(s)ds, \forall t \in [0, T], \tag{7.80}$$

implies

$$u(t) \leq Ce^{K\int_0^t v(s)ds}, t \in [0, T].$$

Define

$$\zeta_n = \sum_{m=0}^{n-1} a(m)M_{m+1}, n \geq 1. \tag{7.81}$$

Lemma 78 *Pick any* $\epsilon > 0$ *and let* $y_n(t)$ *be the trajectory to the ODE* $\dot{y}(t) = g_c(y(t)), t \in [T_n, T_{n+1})$*, with* $y_n(T_n) = \hat{y}(T_n)$*. Then there exists a* c_ϵ *such that If* $r(k) > c_\epsilon$ *for some*

$k > 0$, then for sufficiently large n we have

$$\sup_{t\in[T_n,T_{n+1})} \| \hat{y}(t) - y_n(t) \| \leq \epsilon L T e^{L(L+1)T}, a.s. \tag{7.82}$$

Proof: The arguments below follow close those in Lemma 1, Chapter 2 of [**?**]. For the sake of brevity, we present only those steps that differ from the ones in Lemma 1, Chapter 2 of [**?**] and skip the steps that are exactly the same.

$$\hat{y}(t(n+m)) = \hat{y}(t(n)) + \sum_{k=0}^{m-1} b(n+k) g_c'(\hat{x}(t(n+k)), \hat{y}(t(n+k))) + \delta_{n,n+m}, \tag{7.83}$$

where $\delta_{n,n+m} \stackrel{def}{=} \zeta_{n+m} - \zeta_n$.

$$\begin{aligned} y^{t(n)}(t(m+n)) &= \hat{y}(t(n)) + \int_{t(n)}^{t(n+m)} g_c(y^{t(n)}(t))dt \\ &= \hat{y}(t(n)) + \sum_{k=0}^{m-1} b(n+k) g_c(y^{t(n)}(t(n+k))) \\ &+ \int_{t(n)}^{t(n+m)} (g_c(y^{t(n)}(y)) - g_c(y^{t(n)}([y])))dy. \end{aligned} \tag{7.84}$$

Since $\sup_n \| \hat{y}_n \| < 1$, then for $0 \leq k \leq (m-1)$ and $t \in (t(n+k), t(n+k+1))$ one can then show that

$$\| y^{t(n)}(t) - y^{t(n)}(t(n+k)) \| \leq C_T b(n+k), \tag{7.85}$$

where $C_T \stackrel{def}{=} \| g_c(0) \| + L(1 + \| g_c(0) \| T)e^{LT} < \infty$, a.s. and hence

$$\| \int_{t(n)}^{t(n+m)} (g_c(y^{t(n)}(t)) - g_c(y^{t(n)}([t])))dt \| \leq C_T L \sum_{k=0}^{\infty} b(n+k)^2 \stackrel{n\uparrow\infty}{\to} 0, \text{ a.s.} \tag{7.86}$$

Also, the martingale $(\zeta_n, \mathcal{F}_n)$ converges a.s, we have

$$\sup_{k\geq 0} \| \delta_{n,n+k} \| \stackrel{n\uparrow\infty}{\to} 0, \text{ a.s.} \tag{7.87}$$

Subtracting (7.84) from (7.83) and taking norm we have,

$$\begin{aligned}
\| \hat{y}(t(m+n)) - y^{t(n)}(t(m+n)) \| &\leq \sum_{i=0}^{m-1} b(n+i) \| g_c'(\hat{x}(t(n+i)), \hat{y}(t(n+i))) - g_c(y^{t(n)}(t(n+i))) \| \\
&\quad + C_T \sum_{k\geq 0} b(n+k)^2 + \sup_{k\geq 0} \| \delta_{n,n+k} \|, \text{ a.s.} \\
&= \sum_{i=0}^{m-1} b(n+i) \Big(\| \frac{g(c\hat{x}(t(n+i)), c\hat{y}(t(n+i)))}{c} \\
&\quad - \frac{g(c\lambda_\infty(y^{t(n)}(t(n+i))), cy^{t(n)}(t(n+i)))}{c} \| \Big) \\
&\quad + C_T \sum_{k\geq 0} b(n+k)^2 + \sup_{k\geq 0} \| \delta_{n,n+k} \|, \text{ a.s.} \\
&\leq L \sum_{i=0}^{m-1} b(n+i) \Big(\| \hat{x}(t(n+i)) - \lambda_\infty(y^{t(n)}(t(n+i))) \| \\
&\quad + \| \hat{y}(t(n+i))) - y^{t(n)}(t(n+i))) \| \Big) \\
&\quad + C_T \sum_{k\geq 0} b(n+k)^2 + \sup_{k\geq 0} \| \delta_{n,n+k} \|, \text{ a.s.} \\
&\leq L \sum_{i=0}^{m-1} b(n+i) \Big(\| \epsilon + \lambda_\infty(\hat{y}(t(n+i))) - \lambda_\infty(y^{t(n)}(t(n+i))) \| \Big) \\
&\quad + \| \hat{y}(t(n+i))) - y^{t(n)}(t(n+i))) \| \\
&\quad + C_T \sum_{k\geq 0} b(n+k)^2 + \sup_{k\geq 0} \| \delta_{n,n+k} \|, \text{ a.s.}
\end{aligned}$$

$$\begin{aligned}
&\leq L \sum_{i=0}^{m-1} b(n+i)\epsilon + L(L+1) \sum_{i=0}^{m-1} b(n+i) \| \hat{y}(t(n+i))) - y^{t(n)}(t(n+i))) \| \\
&+ C_T \sum_{k\geq 0} b(n+k)^2 + \sup_{k\geq 0} \| \delta_{n,n+k} \|, \text{ a.s.} \\
&\leq LT\epsilon + L(L+1) \sum_{i=0}^{m-1} b(n+i) \| \hat{y}(t(n+i))) - y^{t(n)}(t(n+i))) \| \\
&+ C_T \sum_{k\geq 0} b(n+k)^2 + \sup_{k\geq 0} \| \delta_{n,n+k} \|, \text{ a.s.}
\end{aligned}$$

In a manner similar to Lemma 1, Chapter 2 of [**?**], one can then define $K_{T,n} = LT\epsilon + C_T L \sum_{k\geq 0} b(n+k)^2 + \sup_{k\geq 0} \parallel \delta_{n,n+k} \parallel$, $z_i \stackrel{def}{=} \parallel \hat{y}(t(n+i)) - y^{t(n)}(t(n+i)) \parallel$ and $d_i \stackrel{def}{=} b(n+i)$. The above inequality becomes

$$z_m \leq K_{T,n} + L(L+1) \sum_{i=0}^{m-1} d_i z_i. \tag{7.88}$$

Then following the lines of arguments as in Lemma 1, Chapter 2 of [**?**] we have

$$\sup_{t\in[T_n,T_{n+1}]} \parallel \hat{y}(t) - y^{t(n)} \parallel \leq K_{T,n} e^{L(L+1)T} + C_T b(n+k), a.s. \tag{7.89}$$

The proof follows by noting that $b(n+k) \to 0$ and $K_{T,n} \to LT\epsilon$ as $n \to \infty$.