# Learning and Value Function Approximation in Complex Decision Processes

by

Benjamin Van Roy

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1998



Author .....
Department of Electrical Engineering and Computer Science
May 1, 1998

Certified by .....
John N. Tsitsiklis
Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

# Learning and Value Function Approximation in Complex Decision Processes

by

Benjamin Van Roy

Submitted to the Department of Electrical Engineering and Computer Science
on May 1, 1998, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

## Abstract

In principle, a wide variety of sequential decision problems – ranging from dynamic resource allocation in telecommunication networks to financial risk management – can be formulated in terms of stochastic control and solved by the algorithms of dynamic programming. Such algorithms compute and store a *value function*, which evaluates expected future reward as a function of current state. Unfortunately, exact computation of the value function typically requires time and storage that grow proportionately with the number of states, and consequently, the enormous state spaces that arise in practical applications render the algorithms intractable.

In this thesis, we study tractable methods that *approximate* the value function. Our work builds on research in an area of artificial intelligence known as *reinforcement learning*. A point of focus of this thesis is temporal–difference learning – a stochastic algorithm inspired to some extent by phenomena observed in animal behavior. Given a selection of basis functions, the algorithm updates weights during simulation of the system such that the weighted combination of basis functions ultimately approximates a value function. We provide an analysis (a proof of convergence, together with bounds on approximation error) of temporal–difference learning in the context of autonomous (uncontrolled) systems as applied to the approximation of (1) infinite horizon discounted rewards and (2) average and differential rewards.

As a special case of temporal–difference learning in a context involving control, we propose variants of the algorithm that generate approximate solutions to optimal stopping problems. We analyze algorithms designed for several problem classes: (1) optimal stopping of a stationary mixing process with an infinite horizon and discounted rewards; (2) optimal stopping of an independent increments process with an infinite horizon and discounted rewards; (3) optimal stopping with a finite horizon and discounted rewards; (4) a zero–sum two–player stopping game with an infinite horizon and discounted rewards. We also present a computational case study involv-

ing a complex optimal stopping problem that is representative of those arising in the financial derivatives industry.

In addition to algorithms for tuning basis function weights, we study an approach to basis function generation. In particular, we explore the use of "scenarios" that are representative of the range of possible events in a system. Each scenario is used to construct a basis function that maps states to future rewards contingent on the future realization of the scenario. We derive, in the context of autonomous systems, a bound on the number of "representative scenarios" that suffices for uniformly accurate approximation of the value function. The bound exhibits a dependence on a measure of "complexity" of the system that can often grow at a rate much slower that the state space size.

Thesis Supervisor: John N. Tsitsiklis
Title: Professor of Electrical Engineering

# Acknowledgments

I greatly acknowledge the support of my thesis advisor Professor John Tsitsiklis who has been an invaluable resource over the past five years. Both my understanding of topics relating to this thesis and the directions taken by the research have been shaped to a large extent by his guidance. The reported work represents an outcome of our collaboration.

I would like to extend special thanks to my thesis reader Professor Dimitri Bertsekas, who has also offered me a great deal of encouragement and advice. Our interaction has had a significant influence on my thinking.

Professors Andrew Barto and Sanjoy Mitter also served as thesis readers. Professor Barto offered valuable insight and historical perspective through discussions that took place at our occasional encounters over the past few years. Professor Mitter has introduced me to several interesting related areas of research and has also generously shared his broad perspective on academic research and careers.

Professor Vivek Borkar read an early draft of the thesis and provided many useful comments. This thesis has also benefited from his deep understanding of stochastic control, which he shared with me during our frequent discussions.

My understanding of reinforcement learning and neuro–dynamic programming benefited from participation in a reading/discussion group led by Professors Bertsekas and Tsitsiklis. Other participants with whom I interacted included Jinane Abounadi, Serafim Batzoglou, Peter Marbach, Wesley McDermott, Steve Patek, Amine Tazi–Riffi, Elias Vyzas, and Cynara Wu.

Interactions with other students at LIDS have also been fruitful. I would especially like to acknowledge Randy Berry, Mike Branicky, and Sekhar Tatikonda, with whom I have discussed a number of research ideas. Randy Berry also read a draft of Chapter 3 and suggested Example 3.5.

I have benefited greatly from participation in the very active reinforcement learning research community. A few individuals have contributed to this thesis in tangible ways. Dr. Richard Sutton's suggestion that the use of simulated trajectories is important to the convergence of temporal–difference learning started me on the path that led to the line of analysis upon which much of this thesis is based. Professor Peter Dayan's suggestion that the line of analysis used in the context of discounted reward temporal–difference learning should be applicable to the study of average reward temporal–difference learning resulted in the contents of Chapter 7. Vassilis Papavassiliou suggested a improvement that led to the current form of error bounds for temporal–difference learning employed in this thesis.

This work was partially supported by the National Science Foundation under grants ECS–9216531 and DMI–9625489.

# Contents

# Chapter 1

# Introduction

In the study of decision–making, there is a dividing line between those who seek an understanding of how decisions *are made* and those who analyze how decisions *ought to be made* in the light of clear objectives. Among the former group are psychologists and economists who examine participants of physical systems in their full complexity. This often entails the consideration of both "rational" and "irrational" behavior. The latter group – those concerned with *rational decision–making* – includes engineers and management scientists who focus on the strategic behavior of sophisticated agents with definite purposes. The intent is to devise strategies that optimize certain criteria and/or meet specific demands. The problems here are well–defined and the goal is to find a "correct" way to make decisions, if one exists.

The self–contained character of rational decision problems has provided a ground for the development of much mathematical theory. Results of this work provide an understanding of various possible models of dynamics, uncertainties, and objectives, as well as whether there exist optimal decision strategies in these settings. In cases where optimal strategies do exist, the theory is complemented by computational methods that deliver them.

In this thesis, we will focus on a particular class of rational decision problems – those involving a single decision–maker that generates a sequence of decisions to influence the evolution of a stochastic dynamic system. A salient characteristic that makes these problems difficult is the need to consider long–term in addition to immediate consequences of decisions. The theory and computational methods associated with this setting are collectively termed *dynamic programming*, and examples of such problems prevail in engineering, operations research, and finance.

## 1.1 Rational Decisions and Natural Systems

In contrast to rational decision–making, there is no clear–cut mathematical theory about decisions made by participants of natural systems. Scientists are forced to propose speculative theories, and to refine their ideas through experimentation. In this context, one approach has involved the hypothesis that behavior is in some sense rational. Ideas from the study of rational decision–making are then used to

characterize such behavior. In financial economics, this avenue has lead to utility and equilibrium theory. To this day, models arising from this school of economic thought – though far from perfect – are employed as mainstream interpretations of the dynamics of capital markets. The study of animal behavior presents another interesting case. Here, evolutionary theory and its popular precept – "survival of the fittest" – support the possibility that behavior to some extent concurs with that of a rational agent.

There is also room for reciprocal contributions from the study of natural systems to the science of rational decision–making. The need arises primarily due to the computational complexity of decision problems and the lack of systematic approaches for dealing with it. For example, practical problems addressed by the theory of dynamic programming can rarely be solved using dynamic programming algorithms because the computational time required for the generation of optimal strategies typically grows exponentially in the number of variables involved – a phenomenon known as the *curse of dimensionality*. This deficiency calls for an understanding of suboptimal decision–making in the presence of computational constraints. Unfortunately, no satisfactory theory has been developed to this end.

It is interesting to note that similar computational complexities arise in attempts to automate decision tasks that are naturally performed by humans or animals. The fact that biological mechanisms facilitate the efficient synthesis of adequate strategies motivates the possibility that understanding such mechanisms can inspire new and computationally feasible methodologies for strategic decision–making.

## 1.2 Temporal–Difference Learning

A focus of this thesis is temporal–difference learning – a computational method inspired by phenomena observed in animal behavior that addresses complex sequential decision problems. The primary goal of this thesis is to advance the state–of–the–art in temporal–difference learning as a useful engineering methodology by developing a theory that guides its application.

Temporal–difference learning can be viewed as a model of how an animal might conceivably learn to make strategic decisions through interaction with a dynamic environment. It involves the construction of a *value function*, which associates expected future rewards with states. This value function, which is also central to dynamic programming, serves as a tool for ranking alternatives in order to guide effective decision–making. Dynamic programming algorithms compute an *optimal value function*, which provides expected future rewards contingent on the fact that the agent will behave optimally. A standard result from dynamic programming theory is that the optimal value function can be used to generate optimal decisions. The main thrust of temporal–difference learning is to *approximate* the optimal value function, and to view the approximation as an adequate guide.

In approximating a value function, temporal–difference learning requires prior specification of a manageably small set of basis functions. Weights associated with the basis functions are iteratively tuned during an interaction with the environment. Ideally, the optimal value function will be either within or "close to" the span of

these basis functions, and the weighted sum of basis functions will converge to a good approximation.[1] It is sometimes convenient to think about the two components of temporal–difference learning in anthropomorphic terms:

1. **Preconceived attributes (basis functions)**
   The basis functions can be viewed as preconceived attributes that are "hard–wired" and customized to the decision task faced by an agent. The drastic reduction from all possible value functions to the span of the bases may curtail the complexity of a decision problem to the point where it becomes computationally tractable.

2. **Learning (iterative computation of weights)**
   The iterative tuning of basis function weights may be viewed as a learning process. In particular, as the agent interacts with the environment, it adjusts these weights based on experience in order to improve its decision capabilities. In temporal–difference learning, this phase is to a large extent problem–independent. Hence, in this model of learning, preconceived attributes, rather than general procedures for learning, distinguish agents from one another.

Though the decision–making problems addressed by temporal–difference learning are fundamentally different from the statistical problems amenable to linear regression, the combination of preselected basis functions and a numerical scheme for computing weights is common to both methodologies. In both contexts, a good choice of basis functions is critical to success. Unfortunately, producing appropriate basis functions may pose a computationally intractable problem, in which case it would probably not be possible to implement a fully automated approach with broad applicability. Instead, we must rely on the intuition or understanding of a human user to provide a set of basis functions that will act as a key to unlock the complexity of the underlying decision problem. The entire problem solving process therefore involves two stages:

1. A human user provides a partial solution by selecting basis functions.

2. A computer complements the partial solution by generating weights.

This interaction between man and machine offers hope for capabilities beyond that which each can supply independently. In particular, human intuition serves a purpose that may be computationally unmanageable (given our current understanding of human intuition), while a machine provides number–crunching capabilities with speed and accuracy that can not be replicated by humans.

[1] Temporal–difference learning can be extended to situations involving nonlinear parameterizations (see, e.g., [13, 76]). However, our primary focus will be on approximations comprised of linear combinations of preselected basis functions.

## 1.3 Control of Complex Systems

Our primary interest in temporal–difference learning concerns its use as a methodology for the control of "complex systems." It is difficult to provide a precise definition for this term, but let us mention two characteristics that are common to such systems: an intractable state space and severe nonlinearities. Intractable state spaces preclude the use of classical dynamic programming algorithms, which compute and store one numerical value per state. At the same time, methods of traditional linear control, which are applicable even when state spaces are large, are ruled out by severe nonlinearities. To give a better feel for the types of problems we have in mind, let us provide a few examples.

1. **Call Admission and Routing**
   With rising demand in telecommunication network resources, effective management is as important as ever. Admission (deciding which calls to accept/reject) and routing (allocating links in the network to particular calls) are examples of decisions that must be made at any point in time. The objective is to make the "best" use of limited network resources. In principle, such sequential decision problems can be addressed by dynamic programming. Unfortunately, the enormous state spaces involved render dynamic programming algorithms inapplicable, and heuristic control strategies are used in lieu.

2. **Strategic Asset Allocation**
   Strategic asset allocation is the problem of distributing an investor's wealth among assets in the market in order to take on a combination of risk and expected return that best suits the investor's preferences. In general, the optimal strategy involves dynamic rebalancing of wealth among assets over time. If each asset offers a fixed rate of risk and return, and some additional simplifying assumptions are made, the only state variable is wealth, and the problem can be solved efficiently by dynamic programming algorithms. There are even closed form solutions in cases involving certain types of investor preferences [51]. However, in the more realistic situation involving risks and returns that fluctuate with economic conditions (see, e.g., [17]), economic indicators must be taken into account as state variables, and this quickly leads to an intractable state space.

3. **Supply–Chain Management**
   With today's tight vertical integration, increased production complexity, and diversification, the inventory flow within a corporation can be viewed as a complex network – called a *supply chain* – consisting of storage, production, and distribution sites. In a supply chain, raw materials and parts from external vendors are processed through several stages to produce finished goods. Finished goods are then transported to distributors, then to wholesalers, and finally retailers, before reaching customers. The goal in supply–chain management is to achieve a particular level of product availability while minimizing costs. The solution is a policy that decides how much to order or produce at various sites

given the present state of the company and the operating environment. See [45] and references therein for further discussion of this problem.

4. **Emissions Reductions**
   The threat of global warming that may result from accumulation of carbon dioxide and other "greenhouse gasses" poses a serious dilemma. In particular, cuts in emission levels bear a detrimental short–term impact on economic growth. At the same time, a depleting environment can severely hurt the economy – especially the agricultural sector – in the longer term. To complicate the matter further, scientific evidence on the relationship between emission levels and global warming is inconclusive, leading to uncertainty about the benefits of various cuts. One systematic approach to considering these conflicting goals involves the formulation of a dynamic system model that describes our understanding of economic growth and environmental science, as is done in [54]. Given such a model, the design of environmental policy amounts to dynamic programming. Unfortunately, classical algorithms are inapplicable due to the size of the state space.

5. **Semiconductor Wafer Fabrication**
   The manufacturing floor at a semiconductor wafer fabrication facility is organized into service stations, each equipped with specialized machinery. There is a single stream of jobs arriving on a production floor. Each job follows a deterministic route that revisits the same station multiple times. This leads to a scheduling problem where, at any time, each station must select a job to service such that (long term) production capacity is maximized (see, e.g., [44]). Such a system can be viewed as a special class of queueing networks, which are models suitable for a variety of applications in manufacturing, telecommunications, and computer systems. Optimal control of queueing networks is notoriously difficult, and this reputation is strengthened by formal characterizations of computational complexity in [55].

When dealing with complex systems of the types we have described, it is common to develop a simulator that can be used to test performance of particular decision policies. We envisage the interfacing of temporal–difference learning with such simulators. In applying temporal–difference learning, a user would first select a set of basis functions, possibly based on a combination of analysis, experience, and intuition. Then, with basis function weights initialized to some arbitrary values, the temporal–difference learning algorithm would be executed. During interaction with the simulator, the algorithm would incrementally tune the basis function weights, which should hopefully converge to values that generate a good approximation. The resulting approximate value function can then be used to produce a policy for deployment in the real world.

## 1.4 Approaches to Approximation

The intractability of exact solutions to important sequential decision problems such as those described in the previous section has spawned the development of many approximation methods. To place temporal–difference learning in perspective, it is worth discussing the range of common approaches to approximation in terms of a few broad categories.

1. **Model Approximation**
   One approach to approximation involves replacing the problem with one that is tractable. There are many domain specific methods that fit into this category. To name one concrete example, in the context of queueing networks, dynamics under heavy traffic conditions can sometimes be approximated via a diffusion process. There are cases where the optimal policy for controlling such an approximate model can be efficiently computed (see, e.g., [30]). Another example arises in the problem of strategic asset allocation that was discussed earlier. Instead of dealing with expected rates of risk and return that fluctuate with market conditions, it is common to consider a model in which rates are constant. An optimal policy can then be generated for this model. This policy is employed until economic conditions, as well as estimated rates of risk and return, change substantially. At such a time, a model involving new constant rates is solved, and the policy under current use is replaced.

2. **Policy Approximation**
   Another approach involves selecting a parameterized class of policies and optimizing over parameter values. As one example of a such a parametric class, in supply–chain management, it is common to limit attention to "$s$–type" (or "order–up–to") policies, which at each site, order inventory to bring levels back up to some fixed target. The targets constitute parameters to be optimized. Unfortunately, the problem of optimizing these targets is itself likely to be intractable, and one must resort to gradient–based methods that search for local optima, such as those considered in the infinitesimal perturbations analysis literature (e.g., [20]), or heuristics for assigning targets (e.g., [45]).

3. **Value Function Approximation**
   Finally, instead of policies, one can select a parameterization of value functions and then try to compute parameters that lead to an accurate approximation to the optimal value function. Algorithms for computing parameters may be variants of exact dynamic programming algorithms. Clearly, temporal–difference learning fits into this category.

The approaches we have described may not be exhaustive and are certainly not exclusive of one another. For instance, approximate models may be used to motivate particular policy or value function parameterizations. Similarly, policy or value function approximation methods might be applied to an approximate model that is simpler than the original but still intractable. An example of this arises with "fluid

approximations" of queueing networks, which result in deterministic continuous time systems that are still intractable but may be easier to deal with. Finally, there are methods that combine elements of policy and value function approximation, such as "actor–critic" algorithms (see, e.g, [13, 76]).

## 1.5 Organization of the Thesis

The rest of the thesis is organized as follows. In the next chapter, we introduce the temporal–difference learning algorithm. We also discuss the current state of the art with regards to both theory and practice, and we summarize the contributions of this thesis. The level of rigor in Chapter 2 is low, as the focus is on developing a general understanding. Chapters 3 through 8 present technical results, and in these chapters, algorithms are formally defined and analyzed. Chapters 3 through 7 focus on methods for tuning basis function weights. Chapter 3 develops some abstract theory that is applied to analyze particular variants of temporal–difference learning in Chapters 4, 5, and 7. Chapter 6 presents a case study involving the application of an approximation algorithm developed in Chapter 5 to a problem of financial derivatives pricing. Chapter 8 departs from the study of weight–tuning algorithms to explore an approach for basis function generation using "representative scenarios." Finally, concluding remarks are made in a closing chapter.

# Chapter 2

# Temporal–Difference Learning

In this chapter, we introduce temporal–difference learning. We will begin by presenting stochastic control as a framework for sequential decision–making and the use of dynamic programming value functions in this context. We then discuss approximations comprised of weighted combinations of basis functions. We define in Sections 3 and 4, temporal–difference learning as applied to tuning basis function weights in autonomous and controlled systems. Finally, we discuss the current state of the art with regards to both theory and practice (Section 5) and summarize the contributions of this thesis (Section 6).

The exposition in this chapter is not rigorous. Instead, emphasis is placed on conveying basic ideas at an intuitive level. The focus is on background material that is relevant to the work presented in the remainder of the thesis. We refer the reader to the texts of Bertsekas and Tsitsiklis [13] and Sutton and Barto [76] for more extensive introductions pertaining to broader classes of algorithms.

## 2.1 Stochastic Control

We consider a discrete–time dynamic system that, at each time $t$, takes on a state $x_t$ and evolves according to

$$x_{t+1} = f(x_t, u_t, w_t),$$

where $w_t$ is a disturbance and $u_t$ is a control decision. Though more general (infinite/continuous) state spaces will be treated later in the thesis, to keep the exposition in this chapter simple, we restrict attention to finite state, disturbance, and control spaces, denoted by $S$, $W$, and $U$, respectively. Each disturbance $w_t \in W$ is independently sampled from some fixed distribution.

A function $g : S \times U \mapsto \Re$ associates a reward $g(x_t, u_t)$ with a decision $u_t$ made at state $x_t$. A *policy* is a mapping $\mu : S \mapsto U$ that generates state–contingent decisions. For each policy $\mu$, we define a value function $J^\mu : S \mapsto \Re$ by

$$J^\mu(x) = \mathrm{E}\left[\sum_{t=0}^{\infty} \alpha^t g(x_t, \mu(x_t)) \middle| x_0 = x\right],$$

where $\alpha \in [0, 1)$ is a discount factor and the state sequence is generated according to $x_0 = x$ and $x_{t+1} = f(x_t, \mu(x_t), w_t)$. Each $J^\mu(x)$ can be interpreted as an assessment of long term rewards given that we start in state $x$ and control the system using a policy $\mu$. The optimal value function $J^*$ is defined by

$$J^*(x) = \max_\mu J^\mu(x).$$

A standard result in dynamic programming states that any policy $\mu^*$ given by

$$\mu^*(x) = \operatorname*{argmax}_{u \in U} \mathop{\mathrm{E}}_{w} \Big[ g(x, u) + \alpha J^*(f(x, u, w)) \Big],$$

where $\mathop{\mathrm{E}}_{w}[\cdot]$ denotes expectation with respect to the distribution of disturbances, is optimal in the sense that

$$J^*(x) = J^{\mu^*}(x),$$

for every state $x$ (see, e.g., [8]).

For illustrative purposes, let us provide one example of a stochastic control problem.

> **Example 2.1** The video arcade game of Tetris can be viewed as an instance of stochastic control (we assume that the reader is familiar with this popular game). In particular, we can view the state $x_t$ as an encoding of the current "wall of bricks" and the shape of the current "falling piece." The decision $u_t$ identifies an orientation and horizontal position for placement of the falling piece onto the wall. Though the arcade game employs a more complicated scoring system, consider for simplicity a reward $g(x_t, u_t)$ equal to the number of rows eliminated by placing the piece in the position described by $u_t$. Then, a policy $\mu$ that maximizes the value
>
> $$J^\mu(x) = \mathrm{E}\left[ \sum_{t=0}^{\infty} \alpha^t g(x_t, \mu(x_t)) \Big| x_0 = x \right],$$
>
> essentially optimizes a combination of present and future row elimination, with decreasing emphasis placed on rows to be eliminated at times farther into the future.

Classical dynamic programming algorithms compute the optimal value function $J^*$. The result is stored in a "look–up" table with one entry $J^*(x)$ per state $x \in S$. When the need arises, the value function is used to generate optimal decisions. In particular, given a current state $x_t \in S$, a decision $u_t$ is selected according to

$$u_t = \operatorname*{argmax}_{u \in U} \mathop{\mathrm{E}}_{w} \Big[ g(x_t, u) + \alpha J^*(f(x_t, u, w)) \Big].$$

Unfortunately, in many practical situations, state spaces are intractable. For example, in a queueing network, every possible configuration of queues corresponds to a different state, and therefore, the number of states increases exponentially with

the number of queues involved. For this reason, it is essentially impossible to compute (or even store) one value per state.

## 2.2 Approximations

The intractability of state spaces calls for value function approximation. There are two important preconditions for the development of an effective approximation. First, we need to choose a parameterization $\tilde{J} : S \times \Re^K \mapsto \Re$ that yields a good approximation

$$\tilde{J}(x, r) \approx J^*(x),$$

for some setting of the parameter vector $r \in \Re^K$. In this respect, the choice of a suitable parameterization requires some practical experience or theoretical analysis that provides rough information about the shape of the function to be approximated. Second, we need algorithms for computing appropriate parameter values. In this section, we introduce linear parameterizations, which are characterized by weighted combinations of basis functions. Such approximations constitute the main class studied in this thesis. Subsequent sections of this chapter present numerical methods for computing parameters.

We will consider parameterizations of the form

$$\tilde{J}(x, r) = \sum_{k=1}^{K} r(k)\phi_k(x),$$

where $\phi_1, \ldots, \phi_K$ are "basis functions" mapping $S$ to $\Re$, and $r = (r(1), \ldots, r(K))'$ is a vector of scalar weights. In a spirit similar to that of statistical regression, the basis functions $\phi_1, \ldots, \phi_K$ are selected by a human user based on intuition or analysis specific to the problem at hand. One interpretation that is useful for the construction of basis functions involves viewing each function $\phi_k$ as a "feature" – that is, a numerical value capturing a salient characteristic of the state that may be pertinent to effective decision making. This general idea is probably best illustrated by a concrete example.

> **Example 2.2** In our stochastic control formulation of Tetris (Example 2.1), the state is an encoding of the current wall configuration and the current falling piece. There are clearly too many states for exact dynamic programming algorithms to be applicable. However, we may believe that most information relevant to game–playing decisions can be captured by a few intuitive features. In particular, one feature, say $\phi_1$, may map states to the height of the wall. Another, say $\phi_2$, could map states to a measure of "jaggedness" of the wall. A third might provide a scalar encoding of the type of the current falling piece (there are seven different shapes in the arcade game). Given a collection of such features, the next task is to

select weights $r(1), \ldots, r(K)$ such that

$$\sum_{k=1}^{K} r(k)\phi_k(x) \approx J^*(x),$$

for all states $x$. This approximation could then be used to generate a game–playing strategy. Such an approach to Tetris has been developed in [83] and [12]. In the latter reference, with 22 features, the authors are able to generate a strategy that eliminates an average of 3554 rows per game, reflecting performance comparable to that of an expert player.

## 2.3 Autonomous Systems

After selecting basis functions for a given stochastic control problem, we are left with the task of computing weights. In the remainder of this chapter, we study temporal–difference learning as an algorithm for computing such weights. We begin in this section by presenting the algorithm in the context of autonomous systems (i.e., those that are not influenced by decisions). In particular, we consider a process

$$x_{t+1} = f(x_t, w_t),$$

and aim at approximating a value function

$$J^*(x) = \mathrm{E}\left[\sum_{t=0}^{\infty} \alpha^t g(x_t) \Big| x_0 = x\right],$$

where $g(x)$ is a scalar reward associated with state $x$ and $\alpha \in [0, 1)$ is a discount factor. Note that this setting is equivalent to one where we are dealing with a controlled system and wish to approximate the value function $J^\mu$ corresponding to a fixed policy $\mu$.

Let $\phi_1, \ldots, \phi_K$ be a collection of basis functions (scalar functions over the state space $S$), and let $\tilde{J} : S \times \Re^K \mapsto \Re$ be defined by

$$\tilde{J}(x, r) = \sum_{k=1}^{K} r(k)\phi_k(x).$$

Suppose that we observe a sequence of states $x_0, x_1, x_2, \ldots$ and that at time $t$ the weight vector has been set to some value $r_t$. We define the *temporal difference* $d_t$ corresponding to the transition from $x_t$ to $x_{t+1}$ by

$$d_t = g(x_t) + \alpha\tilde{J}(x_{t+1}, r_t) - \tilde{J}(x_t, r_t).$$

Then, given an arbitrary initial weight vector $r_0$, the temporal-difference learning algorithm generates subsequent weight vectors according to

$$r_{t+1} = r_t + \gamma_t d_t z_t,$$

where $\gamma_t$ is a scalar step size, and $z_t \in \Re^K$ is an *eligibility vector* defined by

$$z_t = \sum_{\tau=0}^{t} (\alpha\lambda)^{t-\tau}\phi(x_\tau),$$

where $\phi(x) = (\phi_1(x), \ldots, \phi_K(x))'$. The parameter $\lambda$ takes on values in $[0, 1]$, and to emphasize its presence, the temporal-difference learning is often referred to as TD($\lambda$). Note that the eligibility vectors can be recursively updated according to

$$z_{t+1} = \alpha\lambda z_t + \phi(x_{t+1}).$$

In Chapter 4, we will present a formal analysis of temporal–difference learning. For now, let us provide one (heuristic) interpretation of the algorithm. Note that the temporal difference $d_t$ can be viewed as a difference between two predictions of future rewards:

1. $\tilde{J}(x_t, r_t)$ is a prediction of $\sum_{\tau=t}^{\infty} \alpha^{\tau-t} g(x_\tau)$ given our current approximation $\tilde{J}(\cdot, r_t)$ to the value function.

2. $g(x_t) + \alpha\tilde{J}(x_{t+1}, r_t)$ is an "improved prediction" that incorporates knowledge of the reward $g(x_t)$ and the next state $x_{t+1}$.

Roughly speaking, the learning process tries to make predictions $\tilde{J}(x_t, r_t)$ consistent with their improved versions. Note that $\phi(x_t) = \nabla_r \tilde{J}(x_t, r)$. Consequently, when $\lambda = 0$, the update can be rewritten as

$$r_{t+1} = r_t + \gamma_t \nabla_r \tilde{J}(x_t, r_t)\Big(g(x_t) + \alpha\tilde{J}(x_{t+1}, r_t) - \tilde{J}(x_t, r_t)\Big).$$

The gradient can be viewed as providing a direction for the adjustment of $r_t$ such that $\tilde{J}(x_t, r_t)$ moves towards the improved prediction. In the more general case of $\lambda \in [0, 1]$, the direction of the adjustment is determined by the eligibility vector $z_t = \sum_{\tau=0}^{t} (\alpha\lambda)^{t-\tau} \nabla_r \tilde{J}(x_\tau, r_t)$. Here, each gradient term in the summation corresponds to one of the previous states, and the temporal difference can be viewed as "triggering" adjustments of all previous predictions. The powers of $\alpha$ account for discounting effects inherent to the problem, while the powers of $\lambda$ influence the "credit assignment" – that is, the amounts by which previous predictions are to be adjusted based on the current temporal difference.

## 2.4 Controlled Systems

The algorithm described in the previous section involves simulating a system and updating weights of an approximate value function based on observed state transitions. Unlike an autonomous system, a controlled system cannot be passively simulated and observed. Control decisions are required and influence the system's dynamics. In this section, we discuss extensions of temporal–difference learning to this context. The objective is to approximate the optimal value function of a controlled system.

### 2.4.1 Approximate Policy Iteration

A well–known result in dynamic programming is that, given a value function $J^\mu$ corresponding to a policy $\mu$, an improved policy $\overline{\mu}$ can be defined by

$$\overline{\mu}(x) = \operatorname*{argmax}_{u \in U} \mathop{\mathrm{E}}_{w} \Big[ g(x,u) + \alpha J^\mu(f(x,u,w)) \Big].$$

In particular, $J^{\overline{\mu}}(x) \geq J^\mu(x)$ for all $x \in S$. Furthermore, a sequence of policies $\{\mu_m | m = 0, 1, 2, \ldots\}$ initialized with some arbitrary $\mu_0$ and updated according to

$$\mu_{m+1}(x) = \operatorname*{argmax}_{u \in U} \mathop{\mathrm{E}}_{w} \Big[ g(x,u) + \alpha J^{\mu_m}(f(x,u,w)) \Big],$$

converges to an optimal policy $\mu^*$. This iterative method for generating an optimal policy constitutes *policy iteration*, a classical dynamic programming algorithm due to Howard [38].

As with other dynamic programming algorithms, policy iteration suffers from the curse of dimensionality. In particular, each value function $J^{\mu_m}$ generated during the course of the algorithm can not be efficiently computed or stored. A possible approach to overcoming such limitations involves approximating each iterate $J^{\mu_m}$ in terms of a weighted combination of basis functions. For instance, letting $\phi_1, \ldots, \phi_K$ be a set of basis functions and letting $\tilde{J}(x,r) = \sum_{k=1}^K r(k)\phi_k(x)$, consider generating a sequence of weight vectors $r^1, r^2, \ldots$ by selecting each $r^{m+1}$ such that

$$\tilde{J}(x, r^{m+1}) \approx J^{\tilde{\mu}_m}(x),$$

where $\tilde{\mu}_0$ is an arbitrary initial policy and for $m = 1, 2, 3, \ldots$,

$$\tilde{\mu}_m(x) = \operatorname*{argmax}_{u \in U} \mathop{\mathrm{E}}_{w} \Big[ g(x,u) + \alpha \tilde{J}(f(x,u,w), r^m) \Big].$$

We will refer to such an algorithm as *approximate policy iteration.*

There is one key component missing in our description of approximate policy iteration – a method for generating each iterate $r^m$. The possibility we have in mind is, of course, temporal–difference learning. In particular, we can apply the temporal–difference learning algorithm to the autonomous system resulting from simulation of the controlled system under a fixed policy $\tilde{\mu}_m$. (The dynamics are described by $x_{t+1} = f(x_t, \tilde{\mu}_m(x_t), w_t)$.) Initializing with $r_0^{m+1} = r^m$, the algorithm would generate a sequence of vectors $r_1^{m+1}, r_2^{m+1}, r_3^{m+1}, \ldots$ that converges (as will be proven in Chapter 4). The limiting vector provides the subsequent iterate $r^{m+1}$.

To clarify the interplay between the two types of iterations involved in approximate policy iteration, let us note that we have nested sequences:

- An "external" sequence is given by $r^0, r^1, r^2, \ldots$
- For each $m = 1, 2, 3, \ldots$, an "internal" sequence is given by $r_0^m, r_1^m, r_2^m, \ldots$

For each $m$, the internal sequence is initialized with $r_0^{m+1} = r^m$ and the limit of convergence becomes the next element $r^{m+1}$ of the external sequence.

### 2.4.2 Controlled TD

Any function $J : S \mapsto \Re$ can be used to generate a policy

$$\mu(x) = \underset{u \in U}{\operatorname{argmax}} \underset{w}{\mathrm{E}} \Big[ g(x,u) + \alpha J(f(x,u,w)) \Big].$$

In this respect, one can view $J$ as a guide for decision–making. The value functions $J^{\mu_0}, J^{\mu_1}, J^{\mu_2}, \ldots$ generated by (exact) policy iteration can then be viewed as a monotonically improving sequence of guides.

Recall that given a policy $\mu$, the value function $J^{\mu}$ generates an improved policy. It seems therefore seems reasonable to hope that the approximation $\tilde{J}(\cdot, r^{m+1})$ to $J^{\tilde{\mu}_m}$ similarly generates a policy $\tilde{\mu}_{m+1}$ that improves on $\tilde{\mu}_m$. Now recall that, approximate policy iteration employs temporal–difference learning to compute $r^{m+1}$ given $r^m$. This is done by simulating the system under the control policy $\tilde{\mu}_m$, initializing a sequence with $r_0^{m+1} = r^m$, and generating $r_1^{m+1}, r_2^{m+1}, r_3^{m+1}, \ldots$ according to the temporal–difference learning iteration. Since the corresponding sequence of functions $\tilde{J}(\cdot, r^1), \tilde{J}(\cdot, r^2), \tilde{J}(\cdot, r^3), \ldots$ converges to $\tilde{J}(\cdot, r^{m+1})$, one might speculate that these intermediate functions themselves provide improving guides to decision–making, each of which can be used to control the system. This possibility motivates an alternative algorithm, which we refer to as *controlled TD*.

Controlled TD simulates a state trajectory $x_0, x_1, x_2, \ldots$ and then generates weight vectors $r_0, r_1, r_2, \ldots$. The initial state $x_0$ and weight vector $r_0$ can be arbitrary. Given a state $x_t$ and a weight vector $r_t$, a decision $u_t$ is generated according to

$$u_t = \underset{u \in U}{\operatorname{argmax}} \underset{w}{\mathrm{E}} \Big[ g(x_t,u) + \alpha \tilde{J}(f(x_t,u,w), r_t) \Big].$$

The next state $x_{t+1}$ is then given by

$$x_{t+1} = f(x_t, u_t, w_t).$$

Analogously with the autonomous case, let the temporal difference $d_t$ be defined by

$$d_t = g(x_t, u_t) + \alpha \tilde{J}(x_{t+1}, r_t) - \tilde{J}(x_t, r_t).$$

Then, the weight vector is updated according to

$$r_{t+1} = r_t + \gamma_t d_t z_t,$$

where $\gamma_t$ is a scalar step size and the eligibility vector $z_t \in \Re^K$ is once again defined by

$$z_t = \sum_{\tau=0}^{t} (\alpha\lambda)^{t-\tau} \phi(x_\tau).$$

In practice, controlled TD often suffers from getting "stuck" in "deadlock" situations. In particular, viewing the procedure in an anthropomorphic light, the state $x_t$ constitutes an animal's operating environment and $u_t$ is the action it takes. The

action is selected based on an approximate value function $\tilde{J}(\cdot, r_t)$, and the weight vector $r_t$ is improved based on experience. If the animal always selects actions in terms of a deterministic function of $x_t$ and $\tilde{J}(\cdot, r_t)$, there is a possibility that only a small subset of the state space will ever visited and that the animal will never "learn" the value of states outside that region. This is related to the notion of a "self–fulfilling prophecy," whereby the unexplored is never explored because values learned from the explored do not promote further exploration. A modification that has been found to useful in practical applications involves adding "exploration noise" to the controls. In particular, during execution of controlled TD, one might generate decisions according to

$$u_t = \underset{u \in U}{\operatorname{argmax}} \underset{w}{\mathrm{E}} \left[ g(x_t, u) + \alpha \tilde{J}(f(x_t, u, w), r_t) \right] + \eta_t,$$

where $\eta_t$ is a random perturbation that induces exploration of the state space.

### 2.4.3 Approximating the $Q$–Function

Given the optimal value function $J^*$, the generation of optimal control decisions

$$u_t = \underset{u \in U}{\operatorname{argmax}} \underset{w}{\mathrm{E}} \left[ g(x_t, u) + \alpha J^*(f(x_t, u, w)) \right],$$

requires computing one expectation per element of the decision space $U$, which requires in turn repeated evaluation of the system function $f$. One approach to avoiding this computation involves obtaining a "$Q$–function," which maps $S \times U$ to $\Re$ and is defined by

$$Q^*(x, u) = \underset{w}{\mathrm{E}} \left[ g(x, u) + \alpha J^*(f(x, u, w)) \right].$$

Given this function, optimal decisions can be computed according to

$$u_t = \underset{u \in U}{\operatorname{argmax}} Q^*(x_t, u),$$

which no longer involves taking expectations or evaluating the system function.

$Q$–learning is a variant of temporal–difference learning that approximates $Q$ functions, rather than value functions. The basis functions $\phi_1, \ldots, \phi_K$ now map $S \times U$ to $\Re$, and the objective is to obtain a weight vector $r = (r(1), \ldots, r(K))'$ such that

$$Q^*(x, u) \approx \tilde{Q}(x, u, r) = \sum_{k=1}^{K} r(k) \phi_k(x, u).$$

Like in controlled TD, $Q$–learning simulates a state trajectory $x_0, x_1, x_2, \ldots$ and then generates weight vectors $r_0, r_1, r_2, \ldots$. Given a state $x_t$ and a weight vector $r_t$, a decision $u_t$ is generated according to

$$u_t = \underset{u \in U}{\operatorname{argmax}} \tilde{Q}(x_t, u, r_t).$$

The next state $x_{t+1}$ is then given by

$$x_{t+1} = f(x_t, u_t, w_t).$$

The temporal difference $d_t$ is defined by

$$d_t = g(x_t, u_t) + \alpha \tilde{Q}(x_{t+1}, u_{t+1}, r_t) - \tilde{Q}(x_t, u_t, r_t),$$

and the weight vector is updated according to

$$r_{t+1} = r_t + \gamma_t d_t z_t,$$

where $\gamma_t$ is a scalar step size and the eligibility vector $z_t \in \Re^K$ is defined by

$$z_t = \sum_{\tau=0}^{t} (\alpha\lambda)^{t-\tau} \phi(x_\tau, u_\tau).$$

Like in the case of controlled TD, it is often desirable to add exploration noise $\eta_t$, which would result in decisions of the form

$$u_t = \operatorname*{argmax}_{u \in U} \tilde{Q}(x_t, u, r_t) + \eta_t.$$

## 2.5 State of the Art

There is a long history behind the algorithms discussed in the preceding sections. We will attempt to provide a brief account of items that are particularly relevant to understanding the current state of the art, and we refer the reader to the books of Sutton and Barto [76] and Bertsekas and Tsitsiklis [13] for further discussions of the historical development.

Our work builds on ideas that originated in an area of artificial intelligence known as *reinforcement learning*. A major development in this area was the temporal–difference learning algorithm, which was proposed by Sutton [72], but draws on earlier work by Barto and Sutton [75, 6] on models for classical conditioning phenomena observed in animal behavior and by Barto, Sutton, and Anderson on "actor–critic methods." Another major development came with the thesis of Watkins [91], in which "$Q$–learning" was proposed, and the study of temporal–difference learning was integrated with classical ideas from dynamic programming and stochastic approximation theory.[1] The work of Werbos [93, 94, 95] and Barto, Bradtke, and Singh [5] also contributed to this integration.

In addition to advancing the understanding of temporal–difference learning, the marriage with classical engineering ideas furthered the view of the algorithm as one for addressing complex engineering problems and lead to a number of applications. The

[1] Several variants of $Q$–learning have been proposed since the publication of Watkins' thesis, and the one we have presented bears closest resemblance to that studied experimentally by Rummery and Niranjan [62] and Rummery [61].

practical potential was first demonstrated by Tesauro [79, 80, 81], who used a variant of controlled TD to produce a world–class Backgammon playing program. Several subsequent case studies involving problems such as channel allocation in cellular communication networks [68], elevator dispatching [22, 23], inventory management [87], and job–shop scheduling [96], have also shown signs of promise.

Since the completion of Watkin's thesis, there has been a growing literature involving the application of ideas from dynamic programming and stochastic approximation to the analysis of temporal–difference learning and its variants. However, the existing theory does not provide sufficient support for applications, as we will now explain. In controlled TD, approximation accuracy is limited by the choice of a parameterization. The hope, however, is that the iterative computation of parameters should lead to a good approximation relative to other possibilities allowed by this choice. Unfortunately, there is a shortage of theory that ensures desirable behavior of this kind. Most results involving temporal–difference learning apply either to cases where the optimal value function is represented exhaustively (i.e., by as many parameters as there are states) or the system is autonomous. Exceptions include work involving very restrictive types of parameterizations such as those arising from state aggregation [69, 83, 31] and results concerning the performance of approximate policy iteration that rely on overly restrictive assumptions [13].

Due to the absence of adequate theory, there is a lack of streamlined and widely accepted algorithms. Instead, there is a conglomeration of variants to controlled TD, and each one is parameterized by values that must be selected by a user. It is unclear which algorithms and parameter settings will work on a particular problem, and when a method does work, it is still unclear which ingredients are actually necessary for success. As a result, applications often require trial and error in a long process of parameter tweaking and experimentation.

## 2.6 Contributions of the Thesis

A central theme of our work involves the advancement of theory in a way that leads towards algorithms that are widely accessible and applicable to real–world problems. In the context of autonomous systems, significant advances are made in the understanding of temporal–difference learning. As a first step in developing theory pertaining to controlled systems, we propose streamlined algorithms for solving optimal stopping problems and provide rigorous analyses. Finally, in addition to algorithms for tuning parameters, we study a new approach for basis function selection that involves the use of representative scenarios. The remainder of this section describes in greater detail the contributions made in various chapters. Discussions of how these contributions fit into the context of previous research are saved for the closing sections of respective chapters.

### Chapter 3

This chapter starts by reviewing some standard ideas concerning Hilbert spaces and fixed point approximation. We then introduce algorithms that generate an ele-

ment in a prespecified subspace that approximates the fixed point of a given contraction $F$. One algorithm is deterministic, and involves iterations in which compositions of a projection operator and the contraction $F$ are applied. We prove that such an algorithm converges, and we provide an error bound on the resulting approximation. Unfortunately, in many practical situations, this algorithm is computationally infeasible. To overcome this limitation, we develop a stochastic approximation algorithm. We prove that this algorithm converges (almost surely) and that the limit is the same as that generated by the deterministic algorithm. This result is employed in establishing convergence of temporal–difference learning and related algorithms addressed in the remainder of the thesis.

### Chapter 4

We provide an analysis of temporal–difference learning in autonomous systems. Though our analysis offers new results even in the context of simpler settings, our treatment of temporal–difference learning is the first that involves infinite state spaces or infinite horizons. The following results are established.

1. The algorithm converges almost surely for approximations comprised of weighted combinations of (possibly unbounded) basis functions over a (possibly infinite) state space. Earlier results [24] established convergence in the mean – a much weaker form of convergence – in finite–state absorbing Markov chains.

2. The limit of convergence is characterized as the solution to a set of interpretable linear equations, and a bound is placed on the resulting approximation error. Previous works lack interpretable results of this kind.

3. We reconcile positive and negative results in the literature concerning the soundness of temporal–difference learning by proving a theorem that identifies the importance of simulated trajectories.

4. We provide an example demonstrating the possibility of divergence in an extension of temporal-difference learning that is used in conjunction with nonlinear parameterizations.

These results have appeared previously in a paper [84].

### Chapter 5

We provide the first convergence results involving the use of temporal–difference learning in conjunction with general linear parameterizations to solve a class of control problems. In particular, we consider several types of optimal stopping problems:

1. Optimal stopping of a stationary mixing process with an infinite horizon and discounted rewards.

2. Optimal stopping of an independent increments process with an infinite horizon and discounted rewards.

3. Optimal stopping with a finite horizon and discounted rewards

4. A zero–sum two–player stopping game with an infinite horizon and discounted rewards.

In each case, we establish that a value function exists and we use it to characterize optimal stopping times. Though such results are standard in flavor, the nature of our assumptions and analysis are not. The most important aspect of our line of analysis is that it naturally leads to approximation algorithms similar to temporal–difference learning. For each class of optimal stopping problems we consider, we propose an algorithm that tunes basis function weights to approximate the value function, and we prove several results.

1. The algorithm converges almost surely.

2. A bound is placed on the resulting approximation error.

3. A bound is placed on the difference between the performance of the resulting stopping time and the optimal.

These results have appeared previously in a technical report [86].

### Chapter 6

A computational case study is presented involving a complex optimal stopping problem that is representative of those arising in the financial derivatives industry. This application represents a contribution to the growing literature on numerical methods for pricing high–dimensional options, and it demonstrates significant promise for the algorithms of Chapter 5. This case study has appeared in a technical report [86].

### Chapter 7

Most existing work on temporal–difference learning involves a discounted reward criterion, as that presented in the preceding sections. However, in practical applications, it is often actually the average reward that is the criterion of interest. The average reward formulation has largely been avoided in the literature because such a setting was thought to pose greater difficulties, and the discounted reward criterion has been used as a proxy. In this chapter, we propose variant of temporal–difference learning that approximates differential value functions, which substitute for the role of value functions when the average reward criterion is employed. We establish results analogous to the discounted case studied in Chapter 4, and these results suggest that there is no need to introduce discounting when the average reward criterion is desirable.

1. The algorithm converges almost surely.

2. The limit of convergence is characterized as the solution to a set of interpretable linear equations, and a bound is placed on the resulting approximation error.

The line of analysis used appeared previously in a technical report [85].

In addition to proving results analogous to the discounted case, we argue that there are actually advantages to using the average reward version of temporal–difference learning. We show that, as the discount factor $\alpha$ approaches 1, the asymptotic results delivered by the two algorithms are virtually equivalent. However, the transient behavior can be very different, and the average reward version can be computationally more efficient. Our analysis confirms observations made in previous empirical work [49].

### Chapter 8

We explore the use of "scenarios" that are representative of the range of possible events in a system. Each scenario is used to construct a basis function that maps states to future rewards contingent on the future realization of the scenario. We derive, in the context of autonomous systems, a bound on the number of "representative scenarios" that suffice for accurate approximation of the value function using weighted combinations of the basis functions. The bound exhibits a dependence on a measure of "complexity" of system that can often grow at a rate much slower than the state space size. Though we only provide an analysis in the context of autonomous systems, we also discuss possible approaches for generating basis functions from representative scenarios in controlled systems.

# Chapter 3

# Hilbert Space Approximation of Contraction Fixed Points

In this chapter, we define some notation and present a few concepts that are central to our analysis of temporal–difference learning. The intention is to expose the fundamental ideas in an abstract setting divorced from the details of the algorithm. The framework involves approximation in a Hilbert space of the fixed point of a contraction. In the next section, we review some relevant definitions and results from Hilbert space theory. Section 3.2 then presents the contraction mapping theorem together with the successive approximations method for fixed point computation. Subsequent sections present original material pertaining to deterministic and stochastic approaches to fixed point approximation. Our analyses of temporal–difference learning and related algorithms are to a large extent applications of results from these sections.

## 3.1 Hilbert Space

Temporal–difference learning involves the approximation of future reward as a scalar function of state. As a formalism for discussing such approximations, we will introduce Hilbert space – a space of functions that will contain both the value function and its approximations. In this section, we define Hilbert space and present without proof a few mainstream results. Many standard texts (e.g., [2, 27, 47, 59]) offer proofs of these results as well as far more extensive treatments of Hilbert space theory.

Hilbert spaces are a special class of inner–product spaces. Let us begin by defining the latter notion.

**Definition 3.1 (inner–product space)** *An inner–product space is a linear vector space $\mathcal{J}$ together with a real function $\langle \cdot, \cdot \rangle$ on $\mathcal{J} \times \mathcal{J}$, which is referred to as the inner product. The inner product is endowed with the following properties:*

1. $\langle J_1, J_2 \rangle = \langle J_2, J_1 \rangle$ *for all* $J_1, J_2 \in \mathcal{J}$.
2. $\langle cJ_1 + J_2, J_3 \rangle = c\langle J_1, J_3 \rangle + \langle J_2, J_3 \rangle$ *for all* $c \in \Re$ *and* $J_1, J_2 \in \mathcal{J}$.

3. $\langle J, J \rangle \geq 0$ *for all* $J \in \mathcal{J}$.

4. $\langle J, J \rangle = 0$ *if and only if* $J = 0$.

Given an inner product $\langle \cdot, \cdot \rangle$, a norm $\| \cdot \|$ can be defined by letting $\|J\| = \langle J, J \rangle^{1/2}$. The fact that this function constitutes a norm follows from the Cauchy–Shwartz inequality, which we state for future use.

**Theorem 3.2 (Cauchy–Schwartz inequality)** *For any inner product space* $(\mathcal{J}, \langle \cdot, \cdot \rangle)$ *and any* $J_1, J_2 \in \mathcal{J}$,

$$|\langle J_1, J_2 \rangle| \leq \|J_1\| \, \|J_2\|.$$

We say that $J_1, J_2 \in \mathcal{J}$ are *orthogonal* if $\langle J_1, J_2 \rangle = 0$. Orthogonal elements of an inner product space obey the Pythagorean theorem.

**Theorem 3.3 (Pythagorean theorem)** *Let* $(\mathcal{J}, \langle \cdot, \cdot \rangle)$ *be an inner–product space. If* $J_1, J_2 \in \mathcal{J}$ *are orthogonal then*

$$\|J_1 + J_2\|^2 = \|J_1\|^2 + \|J_2\|^2.$$

To motivate our interest in inner product spaces, let us discuss two examples that are relevant to our study of temporal–difference learning.

**Example 3.1** Consider a finite–state Markov chain with a unique "steady–state" distribution $\pi$, which assigns a probability $\pi(x)$ to each state $x$ in the finite state space $S$. (Each $\pi(x)$ represents the relative frequency with which state $x$ is visited.) We define an inner product space $\ell_2(S, \pi)$ with the space of vectors $\Re^{|S|}$ (i.e., functions of state) and an inner product

$$\langle J_1, J_2 \rangle_\pi = \sum_{x \in S} J_1(x) J_2(x) \pi(x).$$

The associated norm is given by

$$\|J\|_\pi = \left( \sum_{x \in S} J^2(x) \pi(x) \right)^{1/2}.$$

Given an approximation $\tilde{J}$ to a value function $J^*$, a natural measure of approximation error is given by $\|J - \tilde{J}\|_\pi$. This is simply a weighted quadratic norm, where states are weighted according to their relative frequencies. Note that the first three properties of an inner–product are easily verified for $\langle \cdot, \cdot \rangle_\pi$. The fourth property, on the other hand, is valid if and only if $\pi$ is strictly positive, which is not true when there are transient states (for which $\pi(x)$ would be equal to zero). This limitation is circumvented, however, by considering any two elements $J_1, J_2 \in \ell_2(S, \pi)$ to be "equal" if $\|J_1 - J_2\|_\pi = 0$. Hence, the elements of $\ell_2(S, \pi)$ are equivalence classes from $\Re^{|S|}$. Consequently, even if some components of $\pi$ are equal to zero, the fourth property is satisfied via considering the condition $J = 0$ to be equivalent to $\|J\|_\pi = 0$.

The inner–product space described in Example 3.1 extends naturally to the case of a Markov process with a continuous state space. We discuss this extension as a second example.

**Example 3.2** Consider a Markov process on a state space $S = \Re^d$ with a unique "steady–state" distribution. Letting $\mathcal{B}(\Re^d)$ be the Borel $\sigma$–algebra associated with $\Re^d$, the distribution is given by a probability measure $\pi$, which assigns a probability $\pi(A)$ to each $A \in \mathcal{B}(\Re^d)$. This probability represents the fraction of time during which the process takes on values in a set $A \subseteq S$. We can define an inner product by

$$\langle J_1, J_2 \rangle_\pi = \int J_1(x) J_2(x) \pi(dx),$$

and its associated norm is given by

$$\|J\|_\pi = \left( \int J^2(x) \pi(dx) \right)^{1/2}.$$

Together with this inner product, the space of Borel–measurable functions over $\Re^d$ with finite norm defines an inner product space that we will denote by $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. Analogously with the finite state case, given an approximation $\tilde{J}$ to a value function $J^*$, $\|J^* - \tilde{J}\|_\pi$ is a natural measure of approximation error. Again, as in the finite state example, elements $J_1, J_2 \in \mathcal{J}$ are considered "equal" if $\|J_1 - J_2\|_\pi = 0$.

A classical framework for approximation involves the concept of "projection." This notion plays a central role in our analysis of temporal–difference learning, and we will now introduce it in terms of the projection theorem. To do so, we must first define Hilbert spaces, which are inner product spaces that possess an additional property of completeness.

**Definition 3.4 (Hilbert space)** *An inner product space $(\mathcal{J}, \langle \cdot, \cdot \rangle)$ is a Hilbert space if it is complete with respect to the metric $\mathbf{d}$ defined by $\mathbf{d}(J_1, J_2) = \|J_1 - J_2\| = \langle J_1 - J_2, J_1 - J_2 \rangle^{1/2}$. By complete, we mean that every Cauchy sequence (with respect to $\mathbf{d}$) in $\mathcal{J}$ has a limit in $\mathcal{J}$.*

It is well–known that the inner–product spaces of Examples 3.1 and 3.2 are Hilbert spaces. In fact, for any measurable space $(S, \mathcal{S})$ (which includes finite spaces and $(\Re^d, \mathcal{B}(\Re^d))$ as special cases) and any probability measure $\pi$ defined over this space, an inner product

$$\langle J_1, J_2 \rangle_\pi = \int J_1(x) J_2(x) \pi(dx),$$

together with the set of measurable functions of finite norm defines a Hilbert space (see, e.g., [27]). We denote such a Hilbert space by $L_2(S, \mathcal{S}, \pi)$.

We now state a version of the projection theorem, adapted from [47].

**Theorem 3.5 (projection theorem)** *Let $(\mathcal{J}, \langle \cdot, \cdot \rangle)$ be a Hilbert space and let $\mathcal{H}$ be a closed subspace of $\mathcal{J}$. Corresponding to any $J \in \mathcal{J}$, there is a unique element*

$\overline{J}_0 \in \mathcal{H}$ *such that* $\|J - \overline{J}_0\| \leq \|J - \overline{J}\|$ *for all* $\overline{J} \in \mathcal{H}$. *Furthermore, a necessary and sufficient condition that* $\overline{J}_0$ *be the unique minimizing element of* $\mathcal{H}$ *is that* $J - \overline{J}_0$ *be orthogonal to* $\mathcal{H}$ *(i.e., orthogonal to every element of* $\mathcal{H}$*).*

We will generally be interested in projections from high or infinite dimensional Hilbert spaces onto finite/low dimensional subspaces. One way to characterize a finite dimensional subspace is as the span of a collection of functions. In particular, given a Hilbert space $(\mathcal{J}, \langle \cdot, \cdot \rangle)$ and elements $\phi_1, \ldots, \phi_K \in \mathcal{J}$, the space

$$\mathcal{H} = \left\{ \sum_{k=1}^{K} r(k)\phi_k \middle| r(1), \ldots, r(K) \in \Re \right\},$$

is a closed subspace referred to as the *span* of $\phi_1, \ldots, \phi_K$.

**Example 3.3** Let us revisit the Hilbert space $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ associated with a Markov processes which has a steady–state distribution $\pi$. Suppose that we would like to approximate a function $J^* \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ via a weighted combination of a set of basis functions $\phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, in the spirit of "feature–based approximation," as motivated in Chapter 2. One approach involves selecting scalar weights $r(1), \ldots, r(K) \in \Re$ that minimize

$$\left\| \sum_{k=1}^{K} r(k)\phi_k - J^* \right\|_\pi .$$

The span of $\phi_1, \ldots, \phi_K$ forms a closed subspace. Hence, the function $\overline{J} = \sum_{k=1}^{K} r(k)\phi_k$ that optimizes the error criterion is the projection of $J^*$.

Given a Hilbert space $(\mathcal{J}, \langle \cdot, \cdot \rangle)$ and a closed subspace $\mathcal{H}$, it is sometimes convenient to define a *projection operator* $\Pi : \mathcal{J} \mapsto \mathcal{H}$ so that for any $J \in \mathcal{J}$, $\Pi J$ is the unique vector $\overline{J} \in \mathcal{H}$ that minimizes $\|J - \overline{J}\|$. This operator enjoys properties presented in the following theorem.

**Theorem 3.6** *Let* $\Pi$ *be a projection operator that projects onto a closed subspace of a Hilbert space* $(\mathcal{J}, \langle \cdot, \cdot \rangle)$. *The following properties hold:*

1. **linearity:** $\Pi(J_1 + J_2) = \Pi J_1 + \Pi J_2$ *for any* $J_1, J_2 \in \mathcal{J}$.
2. **nonexpansiveness:** $\|\Pi J\| \leq \|J\|$ *for any* $J \in \mathcal{J}$.
3. **idempotence:** $\Pi^2 J = \Pi J$ *for any* $J \in \mathcal{J}$.
4. **self–adjointedness:** $\langle \Pi J_1, J_2 \rangle = \langle J_1, \Pi J_2 \rangle$ *for any* $J_1, J_2 \in \mathcal{J}$.

## 3.2 Contractions and Fixed Point Computation

Many problems in numerical computation can be formulated in terms of solving an equation of the form $J = FJ$. A solution $J^*$ to such an equation is called a fixed point of $F$. In this section, we consider the case where $F$ is a contraction on a Hilbert space. Let us begin by defining the term *contraction.*

**Definition 3.7 (contraction)** *Let $(\mathcal{J}, \langle\cdot,\cdot\rangle)$ be a Hilbert space. An operator $F : \mathcal{J} \mapsto \mathcal{J}$ is a contraction if there exists a scalar $\beta \in [0,1)$ such that*

$$\|FJ_1 - FJ_2\| \leq \beta \|J_1 - J_2\|,$$

*for all $J_1, J_2 \in \mathcal{J}$. The contraction factor of $F$ is defined to be the smallest $\beta \in [0,1)$ such that the above inequality is satisfied for all $J_1, J_2 \in \mathcal{J}$.*

When $F$ is a contraction, the set of fixed points is particularly simple, as elucidated by the contraction mapping theorem:

**Theorem 3.8 (contraction mapping theorem)** *Let $F$ be a contraction on a Hilbert space $(\mathcal{J}, \langle\cdot,\cdot\rangle)$. Then, there exists a unique fixed point $J^* \in \mathcal{J}$.*

The proof of this theorem is simple and can be found in many texts (see, e.g., [47]).

A standard method called *successive approximations* computes the fixed point of a contraction $F$ by starting with an approximation $J_0 \in \mathcal{J}$ and generating a sequence $J_1, J_2, J_3, \ldots$ according to

$$J_{m+1} = FJ_m.$$

It is easy to show that this iterative algorithm converges to the fixed point $J^*$. In particular, for each positive integer $m$,

$$\|J_{m+1} - J^*\| = \|FJ_m - FJ^*\| \leq \beta \|J_m - J^*\|,$$

where $\beta \in [0,1)$ is the contraction factor. It follows that, for each nonnegative $m$,

$$\|J_m - J^*\| \leq \beta^m \|J_0 - J^*\|,$$

and therefore, $\lim_{m\to\infty} \|J_m - J^*\| = 0$.

## 3.3 Approximation of Fixed Points

When a Hilbert space contains functions over a large or infinite domain, storage of iterates $J_m$ generated by the successive approximations method becomes infeasible. In this section, we consider a variant of successive approximations that operates on a tractable subspace.

Let $F$ be a contraction in a Hilbert space $(\mathcal{J}, \langle\cdot,\cdot\rangle)$, and suppose that we would like to obtain an approximation $\tilde{J}$ within some closed subspace $\mathcal{H}$. We consider an iterative algorithm that begins with an approximation $J_0 \in \mathcal{H}$ and generates a sequence $\tilde{J}_1, \tilde{J}_2, \tilde{J}_3, \ldots, \in \mathcal{H}$ according to

$$\tilde{J}_{m+1} = \Pi F \tilde{J}_m.$$

Note that each iterate $\tilde{J}_m$ is in $\mathcal{H}$ because this subspace constitutes the range of the projection. Hence, when $\mathcal{H}$ is finite dimensional, each iterate $\tilde{J}_m$ can be represented in terms of basis function weights. This can make storage on a computer possible and/or tractable.

The algorithm we have proposed can be thought of as a variant of successive approximations that approximates each iterate $J_m \in \mathcal{J}$ with $\tilde{J}_m \in \mathcal{J}$. Its use is justified by the following theorem.

**Theorem 3.9** *Let $F$ be a contraction on a Hilbert space $(\mathcal{J}, \langle \cdot, \cdot \rangle)$ with a contraction factor $\beta$, and let $\Pi$ be a projection operator that projects onto a closed subspace of $\mathcal{J}$. Then, the composition $\Pi F$ is a contraction with contraction factor $\kappa \leq \beta$, and its unique fixed point $\tilde{J}$ satisfies*

$$\|\tilde{J} - J^*\| \leq \frac{1}{\sqrt{1-\kappa^2}} \|\Pi J^* - J^*\|.$$

**Proof:** For any $J_1, J_2 \in \mathcal{J}$,

$$\|\Pi F J_1 - \Pi F J_2\| \leq \|F J_1 - F J_2\| \leq \beta \|J_1 - J_2\|,$$

where the first inequality follows from the fact that projections are nonexpansive. It follows that $\Pi F$ is a contraction with a contraction factor $\kappa \leq \beta$. Let $\tilde{J}$ be the fixed point of $\Pi F$. Then, by the Pythagorean Theorem,

$$\begin{aligned} \|\tilde{J} - J^*\|^2 &= \|\tilde{J} - \Pi J^*\|^2 + \|\Pi J^* - J^*\|^2 \\ &= \|\Pi F \tilde{J} - \Pi F J^*\|^2 + \|\Pi J^* - J^*\|^2 \\ &\leq \kappa^2 \|\tilde{J} - J^*\|^2 + \|\Pi J^* - J^*\|^2, \end{aligned}$$

and it follows that

$$\|\tilde{J} - J^*\| \leq \frac{1}{\sqrt{1-\kappa^2}} \|\Pi J^* - J^*\|.$$

**Q.E.D.**

This theorem implies that the iterates $\tilde{J}_m$ converge to some $\tilde{J} \in \mathcal{H}$. Furthermore, this limit $\tilde{J}$ provides an approximation to $J^*$ in a sense that we will now describe. The term $\|\Pi J^* - J^*\|$ represents the error associated with the projection $\Pi J^*$. By the projection theorem, this error is minimal (if we are constrained to selecting approximations from $\mathcal{H}$). The bound given in Theorem 3.9 therefore establishes that the error associated with $\tilde{J}$ is within a constant factor of the best possible.

Let us close this section with an example that captures the spirit of the approximation technique we have described.

> **Example 3.4** Suppose that we wish to approximate the fixed point $J^* \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ of a contraction $F$ via a weighted combination of a set of basis functions $\phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. The need for an approximation here arises in part due to the impracticality of storing one value $J^*(x)$ per point $x$ in the domain $\Re^d$. As in Example 3.3, we might aim at generating an approximation by selecting a set of basis functions

$\phi_1, \ldots, \phi_K \in \mathcal{J}$ and obtaining weights $r(1), \ldots, r(K) \in \Re$ that minimize

$$\left\| \sum_{k=1}^{K} r(k)\phi_k - J^* \right\|_\pi .$$

This would result in a projection $\Pi J^*$ on the subspace

$$\mathcal{H} = \left\{ \sum_{k=1}^{K} r(k)\phi_k \middle| r(1), \ldots, r(K) \in \Re \right\}.$$

However, let us consider a situation where $J^*$ is unavailable, but instead, given any $J \in \mathcal{H}$, we can compute $\Pi F J$. The iterative algorithm $\tilde{J}_{m+1} = \Pi F \tilde{J}_m$ then provides a viable approach to obtaining an approximation. Though the final approximation $\tilde{J}$ may differ from the optimal $\Pi J^*$, the bound of Theorem 3.9 assures that the resulting error is within a constant factor of the best possible.

## 3.4 Stochastic Approximation of Fixed Points

The iterative algorithm of the previous section alleviates the need to store computationally unmanageable functions generated in successive approximations. In particular, instead of storing one value per point in the domain, we store basis function weights $r(1), \ldots, r(K)$. Unfortunately, the computation of $\Pi F \tilde{J}_m$ at each iteration is often intractable.

In this section, we develop a stochastic algorithm that converges to the same approximation $\tilde{J}$ as does the deterministic algorithm from the previous section. However, the stochastic algorithm often alleviates the prohibitive computational requirements associated with the deterministic algorithm. Temporal–difference learning and related methods that we will analyze in subsequent chapters are instances of this stochastic algorithm.

The next subsection makes a digression to present a general class of stochastic algorithms, together with a convergence theorem from [7]. In Subsection 3.4.2, we discuss a specialized subset of such algorithms that includes those that will be studied in later chapters. We also prove a theorem that ensures validity of certain key assumptions made by the convergence theorem of Subsection 3.4.1.

### 3.4.1 Stochastic Approximation

Many stochastic approximation algorithms can be thought of as approaches to approximating solutions of ordinary differential equations. We will motivate and introduce stochastic approximation from this point of view.

Consider an ordinary differential equation of the form

$$\dot{r}_t = \overline{s}(r_t),$$

where $r_t \in \Re^K$ for all $t \geq 0$, and $\overline{s} : \Re^K \mapsto \Re^K$ satisfies

$$(r - r^*)'\overline{s}(r) < 0,$$

for some $r^* \in \Re^K$ and all $r \neq r^*$. It is well–known that, if $\overline{s}$ satisfies suitable regularity conditions, this ordinary differential equation is stable. In particular, for any $r_0 \in \Re^K$, we have $\lim_{t\to\infty} r_t = r^*$.

The solution to the differential equation can be approximated by a difference equation of the form

$$r_{t+1} = r_t + \gamma_t \overline{s}(r_t),$$

where each $\gamma_t$ is a scalar "step size." Under suitable regularity conditions and with an appropriate selection of step sizes, solutions of this difference equation also converge to $r^*$ (see, e.g., [9]).

In certain situations of practical interest, it is difficult to compute $\overline{s}(r_t)$, but we instead have access to a "noisy estimate" $s(y_t, r_t)$. For example, the estimate might be given by

$$s(y_t, r_t) = \overline{s}(r_t) + y_t,$$

where $y_t$ is a zero–mean random variable. An iterative method of the form

$$r_{t+1} = r_t + \gamma_t s(y_t, r_t),$$

is called a stochastic approximation algorithm, and under suitable conditions, the iterates $r_t$ once again converge to $r^*$.

We will present without proof a general result that provides a set of conditions under which a stochastic approximation algorithm converges. The result is a special case of Theorem 17 on page 239 of the book by Benveniste, Métivier, and Priouret [7]. The theorem makes use of the term *Markov*, which we will define now. Let $\{y_t | t = 0, 1, 2, \ldots\}$ be a stochastic process taking on values in a state space $\Re^N$ defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$. Denote the $\sigma$–field generated by random variables $y_0, \ldots, y_t$ by $\mathcal{F}_t$, and take $\mathcal{F}$ to be the smallest $\sigma$–field containing $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$ as sub–$\sigma$–fields. The process is Markov if there exists a scalar function $P$ on $\Re^N \times \mathcal{B}(\Re^N)$ such that, for any $A \in \mathcal{B}(\Re^N)$,

$$\mathcal{P}\{y_{t+1} \in A | \mathcal{F}_t\} = P(y_t, A),$$

and $P(\cdot, A)$ is measurable.

We now present the convergence result. As mentioned earlier, it is a special case of a theorem in [7]. We do not state that theorem in full generality because the list of assumptions is long and would require a lot in terms of new notation. However, we note that in our setting, the potential function $U(\cdot)$ that would be required to satisfy the assumptions of the theorem from [7] is given by $U(r) = \|r - r^*\|_2^2$, where $\|\cdot\|_2$ is used to denote the standard Euclidean norm on finite dimensional spaces.

**Theorem 3.10** *Let $\{y_t | t = 0, 1, 2, \ldots\}$ be a Markov process taking on values in $\Re^N$. Consider a second process $\{r_t | t = 0, 1, 2, \ldots\}$ taking values in $\Re^K$, initialized with an*

*arbitrary vector* $r_0$ *and evolving according to*

$$r_{t+1} = r_t + \gamma_t s(y_t, r_t),$$

*for some* $s : \Re^N \times \Re^K \mapsto \Re^K$. *Let the following assumptions hold for some function* $\overline{s} : \Re^K \mapsto \Re^K$.

1. *There exists some* $r^* \in \Re^K$ *such that* $(r - r^*)'\overline{s}(r) < 0$, *for all* $r \neq r^*$, *and* $\overline{s}(r^*) = 0$.

2. *For any* $q > 0$, *there exists a scalar* $\mu_q$ *such that*

$$\mathrm{E}[\|y_t\|_2^q | y_0] \leq \mu_q(1 + \|y_0\|_2^q),$$

   *for all* $t = 0, 1, 2, \ldots$ *and* $y \in \Re^N$.

3. *There exist scalars* $C$ *and* $q$ *such that*

$$\|s(y, r)\|_2 \leq C(1 + \|r\|_2)(1 + \|y\|_2^q),$$

   *for all* $r \in \Re^K$ *and* $y \in \Re^N$.

4. *There exist scalars* $C$ *and* $q$ *such that*

$$\sum_{t=0}^{\infty} \left\| \mathrm{E}\left[ s(y_t, r) - \overline{s}(r) \middle| y_0 \right] \right\|_2 \leq C(1 + \|r\|_2)(1 + \|y_0\|_2^q),$$

   *for all* $r \in \Re^K$ *and* $y \in \Re^N$, *and*

5. *Let* $\nu : \Re^N \times \Re^K \mapsto \Re^K$ *be defined by*

$$\nu(y, r) = \sum_{t=0}^{\infty} \mathrm{E}\left[ s(y_t, r) - \overline{s}(r) \middle| y_0 = y \right],$$

   *for all* $r \in \Re^K$ *and* $y \in \Re^N$. *There exist scalars* $C$ *and* $q$ *such that*

$$\|\nu(y, r) - \nu(y, \overline{r})\|_2 \leq C\|r - \overline{r}\|_2(1 + \|y\|_2^q),$$

   *for all* $r, \overline{r} \in \Re^K$ *and* $y \in \Re^N$.

6. *The (predetermined) step size sequence* $\gamma_t$ *is nonincreasing and satisfies* $\sum_{t=0}^{\infty} \gamma_t = \infty$ *and* $\sum_{t=0}^{\infty} \gamma_t^2 < \infty$.

*Then,* $r_t$ *almost surely converges to* $r^*$.

Let us comment briefly on the conditions listed in the theorem. Condition (1) was motivated earlier by viewing the algorithm as one that approximates the solution to a stable ordinary differential equation. Condition (2) concerns stochastic stability of $y_t$. Combined with this assumption, Condition (3), which restricts the rate of growth

of $s$, ensures a similar sort of stability for the process $s(y_t, r_t)$. Condition (4) is a "mixing" condition, ensuring that $s(y_t, r)$ approaches the "steady state" version $\overline{s}(r)$ at a sufficiently rapid rate. Condition (5) restricts the extent to which $\nu(y, r)$ can change as $r$ varies. In Condition (6), the fact that the step sizes have an infinite sum ensures that the algorithm does not converge to a point different from $r^*$. When the sum of the squares is finite, the step sizes diminish at a rate that causes the "noise" $\overline{s}(r_t) - s(y_t, r_t)$ to be "averaged out."

### 3.4.2 Approximation of Fixed Points

Temporal–difference learning and related algorithms studied in subsequent chapters are special cases of the stochastic approximation algorithm addressed by Theorem 3.10. However, there is additional structure common to the algorithms we will study, as they all aim at approximating contraction fixed points. In particular, we will consider algorithms for which the function $\overline{s}$ that represents the "steady–state" version of $s(y_t, \cdot)$ is given by

$$\overline{s}_k(r) = \Big\langle \phi_k, F\Phi r - \Phi r \Big\rangle,$$

where $F$ is a contraction on a Hilbert space $(\mathcal{J}, \langle \cdot, \cdot \rangle)$, $\phi_1, \ldots, \phi_K \in \mathcal{J}$ are a set of basis functions, and $\Phi : \Re^K \mapsto \mathcal{J}$ is defined by $\Phi r = \sum_{k=1}^K r(k)\phi_k$ for any $r \in \Re^K$. Hence, the algorithm can be thought of as an approximation to the solution of an ordinary differential equation:

$$\dot{r}_t(k) = \Big\langle \phi_k, F\Phi r_t - \Phi r_t \Big\rangle.$$

To motivate the relevance of this ordinary differential equation, let us consider a special case.

**Example 3.5** Let $(\mathcal{J}, \langle \cdot, \cdot \rangle)$ be a Hilbert space. It is well known that when the basis functions $\phi_1, \ldots, \phi_K \in \mathcal{J}$ are orthonormal (i.e., $\|\phi_k\| = 1$ for each $k$ and $\langle \phi_j, \phi_k \rangle = 0$ for each $j \neq k$), the operator $\Pi$ that projects onto their span is given by

$$\Pi J = \sum_{k=1}^K \phi_k \langle \phi_k, J \rangle,$$

for any $J \in \mathcal{J}$.

Now let us consider the ordinary differential equation

$$\dot{r}_t(k) = \Big\langle \phi_k, F\Phi r_t - \Phi r_t \Big\rangle,$$

assuming that the basis functions are orthonormal. We then have

$$\Phi \dot{r}_t = \sum_{k=1}^K \phi_k \dot{r}_t(k)$$

$$
\begin{aligned}
&= \sum_{k=1}^{K} \phi_k \Big\langle \phi_k, F\Phi r_t - \Phi r_t \Big\rangle \\
&= \Pi F \Phi r_t - \Phi r_t.
\end{aligned}
$$

Hence, given a current approximation $\Phi r_t$, the dynamics of this ordinary differential equation guides the approximation towards $\Pi F \Phi r_t$.

We now state and prove a theorem establishing that Assumption 3.10(1), relating to the stability of the ordinary differential equation, is satisfied by the definition of $\overline{s}$ that we have in mind. In addition, we show that the limiting weight vector $r^*$ generates a function $\tilde{J} = \Phi r^*$ that is the unique fixed point of $\Pi F$.

**Theorem 3.11** *Let $(\mathcal{J}, \langle \cdot, \cdot \rangle)$ be a Hilbert space, let $F : \mathcal{J} \mapsto \mathcal{J}$ be a contraction, let $\phi_1, \ldots, \phi_K \in \mathcal{J}$ be a set of linearly independent functions, and let $\Pi$ be the projection operator that projects from $\mathcal{J}$ onto the span of $\phi_1, \ldots, \phi_K$. Let $\overline{s} : \Re^K \mapsto \Re^K$ be given by*

$$
\overline{s}_k(r) = \Big\langle \phi_k, F\Phi r - \Phi r \Big\rangle,
$$

*for $k = 1, \ldots, K$. Then,*

1. *There exists a unique vector $r^* \in \Re^K$ such that $\Phi r^* = \Pi F \Phi r^*$.*
2. *$(r - r^*)' \overline{s}(r) < 0$ for all $r \neq r^*$.*
3. *$\overline{s}(r^*) = 0$.*
4. *Letting $\kappa$ be the contraction factor of $\Pi F$, $(r - r^*)' \overline{s}(r) \leq (\kappa - 1) \|\Phi r - \Phi r^*\|^2$, for all $r \in \Re^K$.*

**Proof:** From Theorem 3.9 we know that $\Pi F$ is a contraction and has a unique fixed point in the span of $\phi_1, \ldots, \phi_K$. Since $\phi_1, \ldots, \phi_K$ are linearly independent, there is a unique vector $r^*$ that generates this fixed point. This establishes the first part of the theorem.

For the second part,

$$
\begin{aligned}
(r - r^*)' \overline{s}(r) &= \sum_{k=1}^{K} \Big\langle \phi_k, F\Phi r - \Phi r \Big\rangle (r_k - r_k^*) \\
&= \Big\langle \Phi r - \Phi r^*, F\Phi r - \Phi r \Big\rangle \\
&= \Big\langle \Phi r - \Phi r^*, \Pi F \Phi r - \Phi r \Big\rangle,
\end{aligned}
$$

where the final equality follows because $\Pi\Phi = \Phi$ (since $\Pi$ projects onto the range of $\Phi$) and $\Pi$ is self–adjoint. Since $\Phi r^*$ is the fixed point of $\Pi F$, which is a contraction by Theorem 3.9,

$$
\|\Pi F \Phi r - \Phi r^*\| \leq \kappa \|\Phi r - \Phi r^*\|,
$$

for all $r$, where $\kappa$ is the contraction factor of $\Pi F$. Using the Cauchy–Schwartz inequality together with this fact,

$$
\Big\langle \Phi r - \Phi r^*, \Pi F \Phi r - \Phi r \Big\rangle = \Big\langle \Phi r - \Phi r^*, (\Pi F \Phi r - \Phi r^*) + (\Phi r^* - \Phi r) \Big\rangle
$$

$$\begin{aligned} &\leq \|\Phi r - \Phi r^*\| \cdot \|\Pi F \Phi r - \Phi r^*\| - \|\Phi r^* - \Phi r\|^2 \\ &\leq (\kappa - 1)\|\Phi r - \Phi r^*\|^2. \end{aligned}$$

Since $\phi_1, \ldots, \phi_K$ are linearly independent, $\|\Phi r - \Phi r^*\| > 0$ for all $r \neq r^*$. Combining this with the fact that $\kappa < 1$, we have $(r - r^*)'\overline{s}(r) < 0$ for all $r \neq r^*$.

To complete the proof, we have

$$\overline{s}_k(r^*) = \left\langle \phi_k, F\Phi r^* - \Phi r^* \right\rangle = \left\langle \phi_k, \Pi F \Phi r^* - \Phi r^* \right\rangle = 0.$$

**Q.E.D.**

Let us close with an example that illustrates one situation where the type of algorithm studied in this section is applicable.

**Example 3.6** Consider the Hilbert space $L_2(\Re^N, \mathcal{B}(\Re^N), \pi)$ for probability measure $\pi$. Suppose that we would like to approximate the fixed point of a contraction $F$ via a weighted combination of basis functions $\phi_1, \ldots, \phi_K \in \mathcal{J}$, and we have the following capabilities:

1. For any $r \in \Re^K$ and $y \in \Re^N$ we can efficiently compute $(F\Phi r)(y)$.
2. We can generate samples of a random variable distributed according to $\pi$.

Then, we could implement the iteration

$$r_{t+1} = r_t + \gamma_t \phi(y_t)\Big((F\Phi r_t)(y_t) - (\Phi r_t)(y_t)\Big),$$

where each $y_t$ is independently sampled according to the probability distribution $\pi$. The algorithm is a special case of the stochastic approximation algorithm from Subsection 3.4.1 with $\overline{s}$ given by

$$\begin{aligned} \overline{s}_k(r) &= \mathrm{E}\left[\phi_k(y_0)\Big((F\Phi r)(y_0) - (\Phi r)(y_0)\Big)\right] \\ &= \int \phi_k(y_0)\Big((F\Phi r)(y_0) - (\Phi r)(y_0)\Big)\pi(dx) \\ &= \left\langle \phi_k, F\Phi r - \Phi r \right\rangle_\pi, \end{aligned}$$

which is exactly of the form addressed by Theorem 3.11. Notice that this iteration has the same limit as the iteration $\tilde{J}_{m+1} = \Pi F \tilde{J}_m$, but it gets there without explicitly computing a projection.

## 3.5 Closing Remarks

Let us conclude this chapter by overviewing how the results we have developed will be applied in the remainder of the thesis. Three theorems will play important roles: Theorem 3.9 (concerning compositions of projections and contractions), Theorem 3.10 (which is a special case of a result from [7]), and Theorem 3.11 (establishing the convergence of an ordinary differential equation). The methods considered in the next

two chapters, including temporal–difference learning and approximation algorithms for optimal stopping problems, take the form of the stochastic iteration addressed in Theorem 3.10. In order to apply the theorem, we must establish a certain convergence criterion for the related ordinary differential equation, and this is done through use of Theorem 3.11. Finally, Theorem 3.9 provides an error bound for the final approximation. In Chapter 7, we propose a variant of temporal–difference learning that approximates differential reward functions, which are appropriate when dealing with an averaged – as opposed to discounted – reward criterion. This algorithm is of a structure similar to but different from that which would be amenable to Theorems 3.10 and 3.11. These results nevertheless play a significant role in the analysis of this new algorithm.

# Chapter 4

# An Analysis of Temporal–Difference Learning

In this chapter, we present an analysis of temporal–difference learning for autonomous systems, as discussed in Chapter 2. We will begin by defining some terms that characterize the class of autonomous systems under consideration. We then provide in Section 4.2 a formal definition of the algorithm, together with technical assumptions and our main convergence result. In Section 4.3, we recast temporal-difference learning in a way that sheds light into its mathematical structure. Section 4.4 presents the proof of our convergence result. The result is valid for finite–dimensional Euclidean state spaces (or subsets such as countable or finite spaces) under some technical assumptions. In Section 4.5, we show that these technical assumptions are automatically valid for the case of irreducible aperiodic finite–state Markov chains. In Section 4.6, we argue that the class of infinite–state Markov chains that satisfy our assumptions is broad enough to encompass practical situations. The use of a simulated trajectory is of fundamental import to the algorithm's convergence. Section 4.7 contains a converse convergence result that formalizes this point. Section 4.8 considers a generalization of temporal–difference learning that accommodates the use of nonlinear parameterizations. Though this algorithm has been successful in certain practical situations, we show through one example that it can lead to divergence. We close the chapter by discussing how our results fit into the context of other research in the field.

## 4.1 Preliminary Definitions

We will focus on processes that are Markov, stationary, and mixing. In this section, we define these three terms. Our definitions are customized to our purposes, and many texts provide more general definitions together with extensive treatments of such properties of stochastic processes (see, e.g., [15, 60]). Before defining the three terms of interest, let us introduce the transition probability function, which will be used in the subsequent definitions.

**Definition 4.1 (transition probability function)** *A function* $P : \Re^d \times \mathcal{B}(\Re^d) \mapsto [0, 1]$ *is a transition probability function if it satisfies the following conditions.*

1. *For any* $x \in \Re^d$, $P(x, \cdot)$ *is a probability measure.*

2. *For any* $A \in \mathcal{B}(\Re^d)$, $P(\cdot, A)$ *is measurable with respect to* $\mathcal{B}(\Re^d)$.

Given a transition probability function $P$, for each positive integer $t$, we will denote by $P_t$ a "$t$–step transition probability function" defined recursively by $P_1 = P$ and

$$P_t(x, A) = \int P_{t-1}(x, dy)P(y, A),$$

for all $x \in \Re^d$ and $A \in \mathcal{B}(\Re^d)$.

We consider a stochastic process $\{x_t | t = 0, 1, 2, \ldots\}$ taking on values in a state space $\Re^d$ defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$. We denote the $\sigma$–field generated by random variables $x_0, \ldots, x_t$ by $\mathcal{F}_t$, and we take $\mathcal{F}$ to be the smallest $\sigma$–field containing $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$ as sub–$\sigma$–fields. The following definition provides a condition under which the process is considered to be Markov.

**Definition 4.2 (Markov)** *The process* $\{x_t | t = 0, 1, 2, \ldots\}$ *is Markov if there exists a transition probability function* $P$ *such that*

$$\mathcal{P}\{x_{t+1} \in A | \mathcal{F}_t\} = P(A, x_t),$$

*for all* $A \in \mathcal{B}(\Re^d)$.

The probability distribution $\mathcal{P}$ for a Markov process can be defined by an initial distribution $\pi(A) = \mathcal{P}(x_0 \in A)$ and a transition function $P$. In particular, finite–dimensional distributions are given by

$$\begin{aligned}\mathcal{P}\{x_{t_1} \in A_1, \ldots x_{t_m} \in A_m\} \;=\; & \int_{y_0 \in \Re^n} \int_{y_1 \in A_1} \cdots \int_{y_{m-1} \in A_{m-1}} \pi(dy_0) P_{t_1}(y_0, dy_1) P_{t_2 - t_1}(y_1, dy_2) \\ & \cdots P_{t_{m-1} - t_{m-2}}(y_{m-2}, dy_{m-1}) P_{t_m - t_{m-1}}(y_{m-1}, A_m),\end{aligned}$$

for all nonnegative integers $t_1, \ldots, t_m$ and sets $A_1, \ldots, A_m \in \mathcal{B}(\Re^d)$. By Kolmogorov's extension theorem, infinite–dimensional distributions are implicitly defined by the finite–dimensional ones.

We now move on to define stationarity.

**Definition 4.3 (stationary)** *The process* $\{x_t | t = 0, 1, 2, \ldots\}$ *is stationary if*

$$\mathcal{P}\{x_{t_1} \in A_1, \ldots, x_{t_m} \in A_m\} = \mathcal{P}\{x_{t+t_1} \in A_1, \ldots, x_{t+t_m} \in A_m\}.$$

*for all nonnegative integers* $t, t_1, \ldots, t_m$ *and sets* $A_1, \ldots, A_m \in \mathcal{B}(\Re^d)$.

One consequence of stationarity is that

$$E[J(x_t)] = E[J(x_0)],$$

for all nonnegative integers $t$ and functions $J : \Re^d \mapsto \Re$ for which the expectation is well–defined.

Finally, we define a notion of mixing.

**Definition 4.4 (mixing)** *A stationary process* $\{x_t | t = 0, 1, 2, \ldots\}$ *is mixing if*

$$\lim_{t \to \infty} \mathcal{P}\{x_0 \in A_1, x_t \in A_2\} = \mathcal{P}\{x_0 \in A_1\}\mathcal{P}\{x_0 \in A_2\},$$

*for all sets* $A_1, A_2 \in \mathcal{B}(\Re^d)$.

One property that is implied by *mixing* is that the $t$–step transition probabilities converge to a "steady–state." In particular, there exists a distribution $\pi$ such that

$$\lim_{t \to \infty} P_t(x, A) = \pi(A),$$

for all $A \in \mathcal{B}(\Re^d)$ and almost all $x \in \Re^d$. Since the process is stationary, this distribution also satisfies

$$\pi(A) = \mathcal{P}\{x_t \in A\},$$

for all $A \in \mathcal{B}(\Re^d)$ and all nonnegative integers $t$.

Let us close with a simple example of a process that is Markov, stationary, and mixing.

> **Example 4.1** Consider an aperiodic irreducible finite–state Markov chain with a state space $\{1, \ldots, n\}$ and a transition probability matrix $Q \in \Re^{n \times n}$. It is well known that this transition matrix possesses a unique invariant distribution $\pi \in \Re^n$, which is proportional to the left eigenvector with the largest eigenvalue (see, e.g., [28]). Since $\pi$ is an invariant distribution and the process is Markov, the process is stationary if we let the initial state $x_0$ be distributed according to $\pi$. Furthermore, it is well known that, for any initial state $x$, the state probabilities $\mathcal{P}\{x_t = y | x_0 = x\}$ converge to $\pi(y)$, and it easily follows that the process is mixing.

## 4.2 Definition and Convergence Theorem

As in the previous section, we consider a stochastic process $\{x_t | t = 0, 1, 2, \ldots\}$ taking on values in a state space $\Re^d$ defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$. We denote the $\sigma$–field generated by random variables $x_0, \ldots, x_t$ by $\mathcal{F}_t$, and we take $\mathcal{F}$ to be the smallest $\sigma$–field containing $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$ as sub–$\sigma$–fields. We make the following assumption concerning the dynamics of this process.

**Assumption 4.5** *The process* $\{x_t | t = 0, 1, 2, \ldots\}$ *is Markov, stationary, and mixing.*

Since the process is stationary, we can define a distribution $\pi$ satisfying $\pi(A) = \mathcal{P}\{x_t = A\}$ for all nonnegative integers $t$. Central to our analysis will be the Hilbert space $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, which is endowed with an inner product $\langle J_1, J_2 \rangle_\pi = \int J_1(x) J_2(x) \pi(dx)$ and a norm $\|J\|_\pi = \langle J, J \rangle_\pi^{1/2}$.

Let $\alpha \in [0, 1)$ be a discount factor and let $g \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ be a reward function. The value function $J^* : \Re^d \mapsto \Re$ is then defined by

$$J^*(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \alpha^t g(x_t) \Big| x_0 = x\right],$$

(we will establish later that $J^*$ is well–defined and in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$).

Let $\phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ be a set of basis functions, and let $\Pi$ denote the projection operator that projects onto their span. We make the following assumption concerning the basis functions.

**Assumption 4.6** *The basis functions $\phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ are linearly independent.*

It is convenient to define some additional notation. Let a vector–valued function $\phi : \Re^d \mapsto \Re^K$, be defined by $\phi(x) = (\phi_1(x), \ldots, \phi_K(x))'$. Also, let an operator $\Phi : L_2(\Re^d, \mathcal{B}(\Re^d), \pi) \mapsto L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ be defined by

$$\Phi r = \sum_{k=1}^{K} r(k)\phi_k,$$

for any $r = (r(1), \ldots, r(K))'$.

Temporal–difference learning is initialized with a weight vector $r_0 \in \Re^K$ and recursively generates a sequence $\{r_t | t = 1, 2, 3, \ldots\}$. For each $t = 0, 1, 2, \ldots$, given $r_t$, a temporal difference $d_t$ is defined by

$$d_t = g(x_t) + \alpha(\Phi r_t)(x_{t+1}) - (\Phi r_t)(x_t).$$

This temporal difference is then used in generating $r_{t+1}$ according to

$$r_{t+1} = r_t + \gamma_t d_t z_t,$$

where $\{\gamma_t | t = 0, 1, 2, \ldots\}$ is a sequence of scalar step sizes and $\{z_t | t = 0, 1, 2, \ldots\}$ is a sequence of "eligibility vectors" taking on values in $\Re^K$, defined by

$$z_t = \sum_{\tau=0}^{t} (\alpha\lambda)^{t-\tau} \phi(x_\tau).$$

We make the following assumption concerning the step sizes.

**Assumption 4.7** *The sequence $\{\gamma_t | t = 0, 1, 2, \ldots\}$ is prespecified (deterministic), nonincreasing, and satisfies*

$$\sum_{t=0}^{\infty} \gamma_t = \infty \qquad \text{and} \qquad \sum_{t=0}^{\infty} \gamma_t^2 < \infty.$$

One final assumption addresses stability requirements for application of the stochastic approximation result discussed in the previous chapter. As will be shown in Section 4.5, this assumption is always satisfied when the state space is finite. It is also satisfied in many situations of practical interest when the state space is infinite. Further discussion can be found in Section 4.6.

**Assumption 4.8** *The following conditions hold:*

1. *There exist positive scalars $C$ and $q$ such that, for all $x \in \Re^d$,*

$$|g(x)| \leq C(1 + \|x\|_2^q) \qquad \text{and} \qquad \|\phi(x)\|_2 \leq C(1 + \|x\|_2^q).$$

2. *For any $q > 0$, there exists a scalar $\mu_q$ such that for all $x \in \Re^d$ and $t = 0, 1, 2, \ldots$,*

$$\mathrm{E}\Big[\|x_t\|_2^q | x_0\Big] \leq \mu_q(1 + \|x_0\|_2^q) \qquad \text{and} \qquad \mathrm{E}\Big[\|\phi(x_t)\|_2^q | x_0\Big] \leq \mu_q(1 + \|\phi(x_0)\|_2^q).$$

3. *There exists scalars $C$ and $q$ such that, for all $x \in \Re^d$ and $m = 0, 1, 2, \ldots$,*

$$\sum_{t=0}^{\infty} \|\mathrm{E}[\phi(x_t)\phi'(x_{t+m})|x_0 = x] - \mathrm{E}[\phi(x_0)\phi'(x_m)]\|_2 \leq C(1 + \|x\|_2^q),$$

*and*

$$\sum_{t=0}^{\infty} \|\mathrm{E}[\phi(x_t)g(x_{t+m})|x_0 = x] - \mathrm{E}[\phi(x_0)g(x_m)]\|_2 \leq C(1 + \|x\|_2^q).$$

(We use $\|\cdot\|_2$ to denote the standard Euclidean norm on finite dimensional vectors and the Euclidean–induced norm on finite matrices.)

To simplify the statement of our theorem, let us define an operator $T^{(\lambda)}$ that acts on $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ by

$$(T^{(\lambda)}J)(x) = (1 - \lambda) \sum_{m=0}^{\infty} \lambda^m \mathrm{E}\left[\sum_{t=0}^{m} \alpha^t g(x_t) + \alpha^{m+1} J(x_{m+1}) \Big| x_0 = x\right],$$

for $\lambda \in [0, 1)$, and

$$(T^{(1)}J)(x) = \mathrm{E}\left[\sum_{t=0}^{\infty} \alpha^t g(x_t) \Big| x_0 = x\right] = J^*(x),$$

for $\lambda = 1$, so that $T^{(\lambda)}J$ converges to $T^{(1)}J$ as $\lambda$ approaches 1 (under some technical conditions). The fact that $T^{(\lambda)}$ maps $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ to $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ will be established in a later section.

**Theorem 4.9** *Under Assumptions 4.5–4.8, for any $\lambda \in [0, 1]$,*

1. *The value function $J^*$ is in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$.*

2. *The sequence $\{r_t | t = 0, 1, 2, \ldots\}$ converges almost surely.*

3. *The limit of convergence $r^*$ is the unique solution of the equation*

$$\Pi T^{(\lambda)} \Phi r^* = \Phi r^*.$$

*4. The limit of convergence $r^*$ satisfies*

$$\|\Phi r^* - J^*\|_\pi \leq \frac{1}{\sqrt{1-\kappa^2}}\|\Pi J^* - J^*\|_\pi,$$

*where $\kappa$ is the contraction factor of $\Pi T^{(\lambda)}$ and satisfies*

$$\kappa \leq \frac{\alpha(1-\lambda)}{1-\lambda\alpha} \leq \alpha.$$

## 4.3 Understanding Temporal-Difference Learning

One interpretation of temporal–difference learning is as an algorithm that "looks back in time and corrects previous predictions." In this context, the eligibility vector keeps track of how the parameter vector should be adjusted in order to appropriately modify prior predictions when a temporal difference is observed. In this section, we take a different perspective that involves viewing the algorithm as a stochastic approximation method and examining its asymptotic behavior. In the remainder of this section, we introduce this view of TD($\lambda$) and provide an overview of the analysis that it leads to. Our goal is to convey some intuition about how the algorithm works, and in this spirit, we maintain the discussion at an informal level, omitting technical assumptions and other details required to formally prove the statements we make. A formal analysis will be provided in Section 4.4.

### 4.3.1 The TD($\lambda$) Operator

Recall that the TD($\lambda$) operator is defined by

$$(T^{(\lambda)}J)(x) = (1-\lambda)\sum_{m=0}^{\infty}\lambda^m \mathrm{E}\left[\sum_{t=0}^{m}\alpha^t g(x_t) + \alpha^{m+1}J(x_{m+1})\Big|x_0 = x\right],$$

for $\lambda \in [0,1)$, and

$$(T^{(1)}J)(x) = \mathrm{E}\left[\sum_{t=0}^{\infty}\alpha^t g(x_t)\Big|x_0 = x\right] = J^*(x),$$

for $\lambda = 1$. To interpret this operator in a meaningful manner, note that, for each $m$, the term

$$\mathrm{E}\left[\sum_{t=0}^{m}\alpha^t g(x_t) + \alpha^{m+1}J(x_{m+1})\Big|x_0 = x\right]$$

is the expected reward over $m$ transitions plus an approximation to the remaining reward, based on $J$. This sum is sometimes called the "$m$–stage truncated value." Intuitively, if $J$ is an approximation to the value function, the $m$–stage truncated value can be viewed as an improved approximation. Since $T^{(\lambda)}J$ is a weighted average over the $m$–stage truncated values, $T^{(\lambda)}J$ can also be viewed as an improved approximation to $J^*$. In fact, we will prove later that $T^{(\lambda)}$ is a contraction on $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, whose

fixed point is $J^*$. Hence, $T^{(\lambda)}J$ is always closer to $J^*$ than $J$ is, in the sense of the norm $\|\cdot\|_\pi$.

### 4.3.2 Dynamics of the Algorithm

To clarify the fundamental structure of TD($\lambda$), we construct a process $y_t = (x_t, x_{t+1}, z_t)$. It is easy to see that $y_t$ is a Markov process. In particular, $x_{t+1}$ and $z_{t+1}$ are deterministic functions of $y_t$ and the distribution of $x_{t+2}$ only depends on $x_{t+1}$. Note that at each time $t$, the random vector $x_t$, together with the current parameter vector $r_t$, provides all necessary information for computing $r_{t+1}$. By defining a function $s$ with

$$s(r, y) = (g(x) + \alpha(\Phi r)(\overline{x}) - (\Phi r)(x))z,$$

where $y = (x, \overline{x}, z)$, we can rewrite the TD($\lambda$) algorithm as

$$r_{t+1} = r_t + \gamma_t s(r_t, y_t).$$

As we will show later, for any $r$, $s(r, y_t)$ has a well–defined "steady-state" expectation, given by

$$\overline{s}(r) = \lim_{t\to\infty} \mathrm{E}[s(r, y_t)].$$

Intuitively, TD($\lambda$) is a stochastic approximation algorithm with dynamics related to the ordinary differential equation

$$\dot{r}_t = \overline{s}(r_t).$$

It turns out that

$$\overline{s}_k(r) = \left\langle \phi_k, T^{(\lambda)}\Phi r - \Phi r \right\rangle_\pi,$$

and $T^{(\lambda)}$ is a contraction on $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. Consequently, results from Chapter 3 can be used to show that this ordinary differential equation is stable and that, under some technical conditions, TD($\lambda$) converges.

## 4.4 Proof of Theorem 4.5

This section offers a formal analysis of TD($\lambda$) in the form of a proof of Theorem 4.9. We first review a couple standard results that will be used at several points in our analysis. In Section 4.4.2, we prove a few lemmas that characterize the value function, the TD($\lambda$) operator, and the "steady–state" behavior of the updates. Finally, Section 4.4.3 integrates these lemmas with the results of Chapter 3 in order to prove Theorem 4.9.

### 4.4.1 Some Mathematical Background

In this subsection, we review some standard results from probability theory. The stated theorems are adapted from [27].

One result that we will use is Jensen's inequality, which states that the expectation of a convex function of a random variable is greater than or equal to the convex function of the expectation of the random variable.

**Theorem 4.10 (Jensen's Inequality)** *Let $\pi$ be a probability measure on $(\Re^d, \mathcal{B}(\Re^d))$, let $w$ be a random variable defined on this space with $E[||w||] < \infty$, and let $f$ be a convex function on $\Re^d$ such that $f(w)$ is a random variable. Then, $E[f(w)] \geq f(E[w])$.*

In our applications of Jensen's inequality, the function of interest will usually be $f(w) = w^2$. In this case, Jensen's inequality reduces to a statement that the second moment of a random variable is greater than or equal to the square of its expectation.

Another result that we will occasionally use is the Tonelli–Fubini Theorem.

**Theorem 4.11 (Tonelli–Fubini)** *Let $(X, \mathcal{X}, \mathcal{P}_1)$ and $(Y, \mathcal{Y}, \mathcal{P}_2)$ be probability spaces. Let $J : X \times Y \mapsto \Re$ be measurable with respect to $\mathcal{X} \times \mathcal{Y}$. Then,*

$$\int \left( \int J(x,y) \mathcal{P}_1(dx) \right) \mathcal{P}_2(dy) = \int \left( \int J(x,y) \mathcal{P}_2(dy) \right) \mathcal{P}_1(dx),$$

*if (1) $J$ is nonnegative or (2) $J$ is absolutely integrable; i.e.,*

$$\int \left( \int |J(x,y)| \mathcal{P}_1(dx) \right) \mathcal{P}_2(dy) < \infty.$$

This theorem will be used when we wish to switch the ordering of integrals, summations, or expectations.

### 4.4.2 Preliminary Lemmas

By the Markov property, there exists a transition probability function $P$ on $\Re^d \times \mathcal{B}(\Re^d)$ such that, for any $A \in \mathcal{B}(\Re^d)$,

$$\mathcal{P}\{x_{t+1} \in A | \mathcal{F}_t\} = P(x_t, A),$$

and $P(\cdot, A)$ is measurable. We will also use $P$ to denote an operator given by

$$(PJ)(x) = \int J(y) P(x, dy).$$

Note that, for any nonnegative integers $m$ and $t$,

$$(P^m J)(x) = E[J(x_{t+m}) | x_t = x].$$

We begin by proving a fundamental lemma pertaining to the operator $P$.

**Lemma 4.12** *Under Assumption 4.5, for any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, we have $\|PJ\|_\pi \leq \|J\|_\pi$.*

**Proof:** The proof involves Jensen's inequality and stationarity. In particular, for any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$\begin{aligned} \|PJ\|_\pi^2 &= \mathrm{E}\left[(PJ)^2(x_0)\right] \\ &= \mathrm{E}\left[(\mathrm{E}[J(x_1)|x_0])^2\right] \\ &\leq \mathrm{E}\left[\mathrm{E}[J^2(x_1)|x_0]\right] \\ &= \mathrm{E}\left[J^2(x_0)\right] \\ &= \|J\|_\pi^2. \end{aligned}$$

**Q.E.D.**

Our first use of Lemma 4.12 will be in showing that $J^*$ is in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$.

**Lemma 4.13** *Under Assumption 4.5, $J^*$ is in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, and*

$$J^* = \sum_{t=0}^{\infty} (\alpha P)^t g.$$

**Proof:** To establish that $J^*$ is well–defined (for almost all $x$),

$$E\left[\sum_{t=0}^{\infty} \alpha^t |g(x_t)|\right] = \frac{1}{1-\alpha} E[|g(x_0)|] \leq \frac{1}{1-\alpha} \left(E[g^2(x_0)]\right)^{1/2} = \frac{1}{1-\alpha} \|g\|_\pi,$$

where the Tonelli–Fubini theorem, stationarity, and Jensen's inequality have been used. We can apply the Tonelli–Fubini theorem again to obtain

$$J^*(x) = E\left[\sum_{t=0}^{\infty} \alpha^t g(x_t) \Big| x_0 = x\right] = \sum_{t=0}^{\infty} \alpha^t E\left[g(x_t)|x_0 = x\right] = \sum_{t=0}^{\infty} (\alpha^t P^t g)(x).$$

It then follows from Lemma 4.12 that

$$\|J^*\|_\pi \leq \frac{1}{1-\alpha} \|g\|_\pi,$$

and therefore, $J^*$ is in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. **Q.E.D.**

The next lemma provides an alternative characterization of the TD($\lambda$) operator and states that it is a contraction whose fixed point is the value function $J^*$.

**Lemma 4.14** *Let Assumption 4.5 hold. Then,*

1. *For any $\lambda \in [0, 1)$ and any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,*

$$T^{(\lambda)} J = (1-\lambda) \sum_{m=0}^{\infty} \lambda^m \left( \sum_{t=0}^{m} (\alpha P)^t g + (\alpha P)^{m+1} J \right),$$

*which is in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$.*

2. *For any* $\lambda \in [0,1]$ *and any* $J_1, J_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$\|T^{(\lambda)}J_1 - T^{(\lambda)}J_2\|_\pi \leq \frac{\alpha(1-\lambda)}{1-\alpha\lambda}\|J_1 - J_2\|_\pi \leq \alpha\|J_1 - J_2\|_\pi.$$

3. *For any* $\lambda \in [0,1]$, *the value function* $J^*$ *is the unique fixed point of* $T^{(\lambda)}$.

**Proof:** For any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$\begin{aligned}
(T^{(\lambda)}J)(x) &= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m \mathrm{E}\left[\sum_{t=0}^{m}\alpha^t g(x_t) + \alpha^{m+1}J(x_{m+1})\Big| x_0 = x\right] \\
&= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\left(\sum_{t=0}^{m}\alpha^t \mathrm{E}[g(x_t) \mid x_0 = x] + \alpha^{m+1}\mathrm{E}[J(x_{m+1})|x_0 = x]\right) \\
&= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\left(\sum_{t=0}^{m}\alpha^t (P^t g)(x) + \alpha^{m+1}(P^{m+1}J)(x)\right).
\end{aligned}$$

and the first proposition of the lemma follows. The fact that this operator maps $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ onto $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ follows easily from Lemma 4.12.

For the case of $\lambda = 1$, establishing that the operator is a contraction is trivial, and the contraction factor is 0. For $\lambda \in [0,1)$, the result follows from Lemma 4.12. In particular, for any $J_1, J_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$\begin{aligned}
\|T^{(\lambda)}J_1 - T^{(\lambda)}J_2\|_\pi &= \Big\|(1-\lambda)\sum_{m=0}^{\infty}\lambda^m(\alpha P)^{m+1}(J_1 - J_2)\Big\|_\pi \\
&\leq (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\alpha^{m+1}\|J_1 - J_2\|_\pi \\
&= \frac{\alpha(1-\lambda)}{1-\alpha\lambda}\|J_1 - J_2\|_\pi.
\end{aligned}$$

For the case of $\lambda = 1$, $J^*$ is the unique fixed point by definition. For $\lambda \in [0,1)$, the fact that $J^*$ is a fixed point follows from Lemma 4.13 and some simple algebra:

$$\begin{aligned}
T^{(\lambda)}J^* &= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\Big(\sum_{t=0}^{m}(\alpha P)^t g + (\alpha P)^{m+1}J^*\Big) \\
&= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\Big(\sum_{t=0}^{m}(\alpha P)^t g + (\alpha P)^{m+1}\sum_{t=0}^{\infty}(\alpha P)^t g\Big) \\
&= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\Big(\sum_{t=0}^{\infty}(\alpha P)^t g\Big) \\
&= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m J^* \\
&= J^*.
\end{aligned}$$

The contraction property implies that the fixed point is unique. **Q.E.D.**

Recall that, for $y_t = (x_t, x_{t+1}, z_t)$, the TD($\lambda$) iteration takes the form

$$r_{t+1} = r_t + \gamma_t s(r_t, y_t),$$

where the update direction $s$ is defined by

$$s(r, y) = (g(x) + \alpha(\Phi r)(\overline{x}) - (\Phi r)(x))z,$$

for $y = (x, \overline{x}, z)$. The following lemma characterizes the "steady–state" behavior of the updates. We will employ the notation $\mathop{\mathrm{E}}_{t\to\infty}[\cdot]$ as shorthand for $\lim_{t\to\infty} \mathrm{E}[\cdot]$.

**Lemma 4.15** *Under Assumption 4.5, for any $r \in \Re^K$, $k = 1, \ldots, K$, and $\lambda \in [0, 1]$,*

$$\mathop{\mathrm{E}}_{t\to\infty}[s_k(r, y_t)] = \left\langle \phi_k, T^{(\lambda)}\Phi r - \Phi r \right\rangle_\pi.$$

**Proof:** For any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ and $k = 1, \ldots, K$,

$$\begin{aligned}
\mathop{\mathrm{E}}_{t\to\infty}[z_t(k)J(x_t)] &= \lim_{t\to\infty} \mathrm{E}\left[\sum_{\tau=0}^{t} (\alpha\lambda)^{t-\tau}\phi_k(x_\tau)J(x_t)\right] \\
&= \lim_{t\to\infty} \mathrm{E}\left[\sum_{m=0}^{t} (\alpha\lambda)^{m}\phi_k(x_{t-m})J(x_t)\right] \\
&= \lim_{t\to\infty} \sum_{m=0}^{t} (\alpha\lambda)^{m}\mathrm{E}\left[\phi_k(x_{t-m})J(x_t)\right] \\
&= \sum_{m=0}^{\infty} (\alpha\lambda)^{m}\mathrm{E}\left[\phi_k(x_0)J(x_m)\right]
\end{aligned}$$

where $m = t - \tau$ and the final expression follows from stationarity. Observe that for any $J_1, J_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$\mathrm{E}[J_1(x_0)J_2(x_m)] = \mathrm{E}[J_1(x_0)(P^m J_2)(x_0)] = \langle J_1, P^m J_2\rangle_\pi \leq \|J_1\|_\pi \|J_2\|_\pi,$$

by Lemma 4.12 and the Cauchy–Schwartz inequality. It follows that, for any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$\mathop{\mathrm{E}}_{t\to\infty}[z_t(k)J(x_t)] = \sum_{m=0}^{\infty} (\alpha\lambda)^m \langle \phi_k, P^m J\rangle_\pi,$$

and the magnitude of this expression is finite. By specializing to the case of $J = g + \alpha P\Phi r - \Phi r$, for any $r \in \Re^K$ and $k = 1, \ldots, K$,

$$\begin{aligned}
\mathop{\mathrm{E}}_{t\to\infty}[s_k(r, y_t)] &= \mathop{\mathrm{E}}_{t\to\infty}\left[z_t(k)\Big(g(x_t) + \alpha(\Phi r)(x_{t+1}) - (\Phi r)(x_t)\Big)\right] \\
&= \mathop{\mathrm{E}}_{t\to\infty}\left[z_t(k)\Big(g(x_t) + \alpha(P\Phi r)(x_t) - (\Phi r)(x_t)\Big)\right] \\
&= \sum_{m=0}^{\infty} (\alpha\lambda)^m \Big\langle \phi_k, P^m(g + \alpha P\Phi r - \Phi r)\Big\rangle_\pi.
\end{aligned}$$

In the case of $\lambda = 1$, it follows that

$$\lim_{t\to\infty} \mathrm{E}[s_k(r, y_t)] = \left\langle \phi_k, \sum_{m=0}^{\infty} \alpha^m P^m (g + \alpha P \Phi r - \Phi r) \right\rangle_\pi = \langle \phi_k, J^* - \Phi r \rangle_\pi.$$

Note that for any $\lambda \in [0, 1)$ and $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$\sum_{m=0}^{\infty} (\lambda\alpha)^m P^m J = (1-\lambda) \sum_{m=0}^{\infty} \lambda^m \sum_{t=0}^{m} \alpha^t P^t J.$$

We therefore have, for any $\lambda \in [0, 1)$, $r \in \Re^K$, and $k = 1, \ldots, K$,

$$\begin{aligned} \lim_{t\to\infty} \mathrm{E}[s_k(r, y_t)] &= \left\langle \phi_k, (1-\lambda) \sum_{m=0}^{\infty} \lambda^m \sum_{t=0}^{m} \alpha^t P^t \Big( g + \alpha P \Phi r - (\Phi r) \Big) \right\rangle_\pi \\ &= \left\langle \phi_k, T^{(\lambda)} \Phi r - \Phi r \right\rangle_\pi. \end{aligned}$$

**Q.E.D.**

### 4.4.3 Proof of the Theorem

As discussed earlier, the process $y_t$ is Markov, and the TD($\lambda$) iteration is given by

$$r_{t+1} = r_t + \gamma_t s(y_t, r_t),$$

which is of the form of the stochastic approximation algorithm studied in Chapter 3. Under certain technical conditions, convergence of such an iteration is established by Theorem 3.10. We will show that these conditions are valid in the case of TD($\lambda$). We let the function $\overline{s}$ required by the conditions be given by

$$\overline{s}(r) = \left\langle \phi_k, T^{(\lambda)} \Phi r - \Phi r \right\rangle_\pi.$$

Also, though we have considered until this point a process $y_t = (x_t, x_{t+1}, z_t)$, for the purposes of validating conditions of Theorem 3.10, we let $y_t = (x_t, x_{t+1}, z_t, \phi(x_{t+1}))$ (the process takes on values in $\Re^N$ with $N = 2d + 2K$). Note that there is a one-to-one mapping between the two versions of $y_t$, so the update direction $s$ is still a function of $r_t$ and $y_t$. We now address the six conditions of Theorem 3.10.

1. Given that the basis functions are independent (Assumption 4.6) and $T^{(\lambda)}$ is a contraction (Lemma 4.14), Theorem 3.11 establishes validity of condition (1). Note that Theorem 3.11 states, in addition, that the limit of convergence $r^* \in \Re^K$ is the unique solution to $\Phi r^* = \Pi T^{(\lambda)} \Phi r^*$, and that

$$\|\Phi r^* - J^*\|_\pi \leq \frac{1}{\sqrt{1-\kappa^2}} \|\Pi J^* - J^*\|_\pi,$$

where $\kappa$ is the contraction factor of $\Pi T^{(\lambda)}$, which by Theorem 4.9, satisfies

$$\kappa \leq \frac{\alpha(1-\lambda)}{1-\lambda\alpha} \leq \alpha.$$

2. Using Assumption 4.8(2), for any $t = 1, 2, \ldots$ and any scalar $q > 0$, it is easy to show that there exists a scalar $C$ such that

$$\mathrm{E}[\|z_t\|_2^q | y_0] \leq \mathrm{E}\left[\left(\sum_{\tau=0}^{t}(\alpha\lambda)^{t-\tau}\|\phi(x_\tau)\|_2\right)^q \Big| y_0\right] \leq \frac{1}{1-\alpha\lambda}C(1+\|y_0\|_2^q).$$

Making further use of Assumption 4.8(2), for any $q > 1$, there exist scalars $C_1$ and $C_2$ such that

$$\begin{aligned}
\mathrm{E}[\|y_t\|_2^q | y_0] &\leq \mathrm{E}\left[(\|x_t\|_2 + \|x_{t+1}\|_2 + \|z_t\|_2 + \|\phi(x_{t+1})\|_2)^q \Big| y_0\right] \\
&\leq C_1\mathrm{E}\left[\|x_t\|_2^q + \|x_{t+1}\|_2^q + \|z_t\|_2^q + \|\phi(x_{t+1})\|_2^q \Big| y_0\right] \\
&\leq C_2\left(1 + \|x_0\|_2^q + \|x_1\|_2^q + \|z_0\|_2^q + \|\phi(x_1)\|_2^q\right) \\
&\leq C_2(1+\|y_0\|_2^q).
\end{aligned}$$

For the case of $q \in (0,1)$, by Jensen's inequality,

$$\mathrm{E}[\|y_t\|_2^q | y_0 = y] \leq \left(\mathrm{E}[\|y_t\|_2^{q(1/q)} | y_0]\right)^q \leq C(1+\|y_0\|_2^q),$$

for some scalars $C$ and $q$. Hence, the condition is valid for all $q > 0$.

3. For any $r$ and $y = (x, \overline{x}, z, \cdot)$, by the Cauchy–Schwartz Inequality (on $\Re^K$).

$$\|s(r,y)\|_2 \leq \Big(|g(x)| + \|r\|_2(\|\phi(x)\|_2 + \|\phi(\overline{x})\|_2)\Big)\|z\|_2.$$

Then, by Assumption 4.8(1), there exist positive scalars $C$ and $q$ such that,

$$\|s(r,y)\|_2 \leq C\Big(1 + \|x\|_2^q + \|r\|_2(1 + \|x\|_2^q + \|\overline{x}\|_2^q)\Big)\|z\|_2.$$

Validity of the condition easily follows.

4. By Lemma 4.15, $\overline{s}(r) = \underset{t\to\infty}{\mathrm{E}}[s(r, y_t)]$. We therefore study differences between $\mathrm{E}[s(r, y_t)|y_0]$ and $\underset{t\to\infty}{\mathrm{E}}[s(r, y_t)]$. Let us concentrate on a term

$$\Delta = \sum_{m=0}^{\infty}\left\|\mathrm{E}[z_m(\Phi r)(x_m)|x_0 = x] - \underset{t\to\infty}{\mathrm{E}}[z_t(\Phi r)(x_t)]\right\|_2,$$

for some fixed $x$. By the triangle inequality,

$$\Delta \leq \sum_{m=0}^{\infty}\sum_{\tau=0}^{m}(\alpha\lambda)^{m-\tau}\left\|\mathrm{E}[\phi(x_\tau)(\Phi r)(x_m)|x_0 = x] - \mathrm{E}[\phi(x_\tau)(\Phi r)(x_m)]\right\|_2$$

$$+\sum_{m=0}^{\infty}\left(\sum_{k=1}^{K}\left(\sum_{\tau=m+1}^{\infty}(\alpha\lambda)^{\tau}\langle\phi_k, P^{\tau}\Phi r\rangle_{\pi}\right)^2\right)^{1/2}.$$

Let

$$\Delta_1 = \sum_{m=0}^{\infty}\sum_{\tau=0}^{m}(\alpha\lambda)^{m-\tau}\left\|\mathrm{E}[\phi(x_\tau)(\Phi r)(x_m)|x_0=x] - \mathrm{E}[\phi(x_\tau)(\Phi r)(x_m)]\right\|_2,$$

and

$$\Delta_2 = \sum_{m=0}^{\infty}\left(\sum_{k=1}^{K}\left(\sum_{\tau=m+1}^{\infty}(\alpha\lambda)^{\tau}\langle\phi_k, P^{\tau}\Phi r\rangle_{\pi}\right)^2\right)^{1/2}.$$

Letting $n = m - \tau$,

$$\begin{aligned}
\Delta_1 &= \lim_{t\to\infty}\sum_{m=0}^{t}\sum_{\tau=0}^{m}(\alpha\lambda)^{m-\tau}\left\|\mathrm{E}[\phi(x_\tau)(\Phi r)(x_m)|x_0=x] - \mathrm{E}[\phi(x_\tau)(\Phi r)(x_m)]\right\|_2 \\
&= \lim_{t\to\infty}\sum_{\tau=0}^{t}\sum_{m=\tau}^{t}(\alpha\lambda)^{m-\tau}\left\|\mathrm{E}[\phi(x_\tau)(\Phi r)(x_m)|x_0=x] - \mathrm{E}[\phi(x_\tau)(\Phi r)(x_m)]\right\|_2 \\
&= \lim_{t\to\infty}\sum_{\tau=0}^{t}\sum_{n=0}^{t-\tau}(\alpha\lambda)^{n}\left\|\mathrm{E}[\phi(x_\tau)(\Phi r)(x_{\tau+n})|x_0=x] - \mathrm{E}[\phi(x_\tau)(\Phi r)(x_{\tau+n})]\right\|_2 \\
&\le \sum_{\tau=0}^{\infty}\sum_{n=0}^{\infty}(\alpha\lambda)^{n}\left\|\mathrm{E}[\phi(x_\tau)(\Phi r)(x_{\tau+n})|x_0=x] - \mathrm{E}[\phi(x_\tau)(\Phi r)(x_{\tau+n})]\right\|_2 \\
&= \sum_{n=0}^{\infty}(\alpha\lambda)^{n}\sum_{\tau=0}^{\infty}\left\|\mathrm{E}[\phi(x_\tau)(\Phi r)(x_{\tau+n})|x_0=x] - \mathrm{E}[\phi(x_\tau)(\Phi r)(x_{\tau+n})]\right\|_2 \\
&= \frac{1}{1-\alpha\lambda}\sup_{n\ge 0}\sum_{\tau=0}^{\infty}\left\|\mathrm{E}[\phi(x_\tau)(\Phi r)(x_{\tau+n})|x_0=x] - \mathrm{E}[\phi(x_\tau)(\Phi r)(x_{\tau+n})]\right\|_2.
\end{aligned}$$

From Assumption 4.8(3), it then follows that there exist scalars $C$ and $q$ such that

$$\Delta_1 \le \frac{C}{1-\alpha\lambda}(1 + \|x\|_2^q)\|r\|_2.$$

By Lemma 4.12 and the Cauchy–Schwartz inequality, for any $k = 1, \ldots, K$,

$$|\langle\phi_k, P^{\tau}\Phi r\rangle_{\pi}| \le \|\phi_k\|_{\pi}\|\Phi r\|_{\pi} \le C(1 + \|r\|_2),$$

for some scalar $C$, and therefore, there exists a scalar $C$ such that

$$\Delta_2 \le C(1 + \|r\|_2).$$

It follows that there exist scalars $C$ and $q$ such that

$$\Delta \le C(1 + \|x\|_2^q)(1 + \|r\|_2).$$

Using roughly the same line of reasoning, similar bounds can be established for other terms involved in

$$\sum_{t=0}^{\infty} \left\| \mathrm{E}\left[ s(y_t, r) - \overline{s}(r) \middle| y_0 \right] \right\|_2 ,$$

and these bounds can be combined to validate the condition. We omit these tedious details.

5. Note that both $s$ and $\overline{s}$ are affine functions of $r$. Combining this fact with reasoning similar to that employed in verifying condition (4), it is not difficult to show that that there exists scalars $C$ and $q$ such that

$$\|\nu(y, r) - \nu(y, \overline{r})\|_2 \leq C\|r - \overline{r}\|_2(1 + \|y\|_2^q).$$

Once again, we omit the details.

6. This condition is the same as Assumption 4.7.

Given that all the conditions of Theorem 3.10 are valid, the algorithm converges to a unique vector $r^*$ (statement (2) of the theorem). The properties of this vector (statements (3)–(4)), are established in the comments pertaining to Condition (1) (see above item (1) above). The fact that $J^*$ is in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ (statement (1)) is established by Lemma 4.13.

## 4.5 The Case of a Finite State Space

In this section, we show that Assumptions 4.5 and 4.8 are automatically true whenever we are dealing with an irreducible aperiodic finite–state Markov chain. This tremendously simplifies the conditions required to apply Theorem 4.9, reducing them to a requirement that the basis functions be linearly independent (Assumption 4.6). Actually, even this assumption can be relaxed if Theorem 4.9 is stated in a more general way. This assumption was adopted for the sake of simplicity in the proof.

Let us now assume that $\{x_t | t = 0, 1, 2, \ldots\}$ is generated by an irreducible aperiodic finite-state Markov chain taking on values in a state space $S = \{1, \ldots, n\}$ (this is just a subset of $\Re$, so our results apply). It is well–known that Assumptions 4.5 is satisfied in this case (see, e.g., [28]). Also, Assumption 4.8(1) is trivially satisfied because the functions $g$ and $\phi_1, \ldots, \phi_K$ are bounded over the finite state space. We will therefore focus on showing that Assumptions 4.8(2)–(3) are valid.

It is well known that for any irreducible aperiodic finite–state Markov chain, there exist scalars $\rho < 1$ and $C$ such that

$$|\mathcal{P}\{x_t = \overline{x} | x_0 = x\} - \pi(\overline{x})| \leq C\rho^t, \qquad \forall x \in S.$$

For each $x \in S$, we define a $K \times K$ diagonal matrix $D$ with diagonal elements equal to the steady–state probabilities $\pi(1), \ldots, \pi(n)$ and a sequence of $K \times K$ diagonal

matrices $D_{x,t}$ with the $\overline{x}$th diagonal element equal to $\mathcal{P}\{x_t = \overline{x}|x_0 = x\}$. Note that

$$\|D_{x,t} - D\|_2 \leq C\rho^t,$$

for all $x \in S$. It is easy to show that

$$\mathrm{E}[\phi(x_t)\phi'(x_{t+m})|x_0 = x] = \Phi' D_{x,t} P^m \Phi.$$

(Note that the operators $P$ and $\Phi$ are matrices when the state space is finite.) We then have

$$\mathrm{E}[\phi(x_t)\phi'(x_{t+m})|x_0 = x] - E_0[\phi(x_t)\phi'(x_{t+m})] = \Phi'(D_{x,t} - D)P^m\Phi.$$

Note that all entries of $P^m$ are bounded by 1 and therefore there exists a constant $G$ such that $\|P^m\|_2 \leq G$ for all $m$. We then have

$$\begin{aligned}
\sum_{t=0}^{\infty} \|\Phi'(D_{x,t} - D)P^m\Phi\|_2 &\leq \sum_{t=0}^{\infty} K^2 \max_{k,j} |\phi_k'(D_{x,t} - D)P^m\phi_j| \\
&\leq K^2 \max_k \|\phi_k\|_2 G \max_j \|\phi_j\|_2 \sum_{t=0}^{\infty} \|D_{x,t} - D\|_2 \\
&\leq GK^2 \max_k \|\phi_k\|_2^2 \frac{C}{1-\rho}.
\end{aligned}$$

The first part of Assumption 4.8(3) is therefore satisfied by a constant bound (i.e., let $q = 0$). An analogous argument, which we omit, can be used to establish that the same is true for the second part of Assumption 4.8(3). Assumption 4.8(2) is trivially satisfied.

## 4.6 Infinite State Spaces

The purpose of this section is to shed some light on the nature of Assumption 4.8 and to suggest that our results apply to infinite–state Markov processes of practical interest.

Let us first assume that the state space is a bounded subset $S \subset \Re^d$ and that the reward function $g$ and basis functions $\phi_1, \ldots, \phi_K$ are continuous on $\Re^d$. Then, the functions $g$ and $\phi_1, \ldots, \phi_K$ are bounded over the state space, and Assumption 4.8(1) is satisfied. Assumption 4.8(3) basically refers to the speed with which the process reaches steady-state. Let $\pi_{x,t}$ be a probability measure defined by

$$\pi_{x,t}(A) = \mathcal{P}\{x_t \in A|x_0 = x\}.$$

Then, Assumption 4.8(3) is satisfied if we require that there exists a scalar $C$ such that

$$\sum_{t=0}^{\infty} |\pi_{x,t}(A) - \pi(A)| \leq C,$$

for all $A \subseteq S$ and $x \in S$. In other words, we want the $t$–step transition probabilities to converge fast enough to the steady–state probabilities (for example, $|\pi_{x,t}(A) - \pi(A)|$ could decay at a rate of $1/t^2$). In addition, we require that this convergence be uniform in the initial state.

As a special case, suppose that there is a distinguished state, say state $\overline{x}$, and that for some $\delta > 0$,

$$\mathcal{P}\{x_{t+1} = \overline{x} | x_t = x\} \geq \delta \qquad \forall x.$$

Then, $\pi_{x,t}(x)$ converges to $\pi$ exponentially fast, and uniformly in $x$, and Assumption 4.8(3) is satisfied with a constant bound. Assumption 4.8(2) is again trivially satisfied, since the state space is bounded.

Let us now consider the case where the state space is an unbounded set $S \subseteq \Re^d$. For many stochastic processes of practical interest (e.g., those that satisfy a large deviations principle), the tails of the probability distribution $\pi$ exhibit exponential decay; let us assume that this is the case.

Assumption 4.8(2) is essentially a stability condition; it states that $\|x_t\|_2^q$ and $\|\phi(x_t)\|_2^q$ are not expected to grow too rapidly, and this is satisfied by most stable Markov processes of practical interest. Note that by taking the expectation with respect to the stationary distribution we obtain $\mathrm{E}[\|x_0\|_2^q] < \infty$ and $\mathrm{E}[\|\phi(x_0)\|_2^q] < \infty$ for all $q > 0$, which in essence says that the tails of the steady-state distribution $\pi$ decay faster than certain polynomials (e.g., exponentially).

Assumption 4.8(3) is the most complex one. Recall that it deals with the speed of convergence of certain functions of the Markov process to steady-state. Whether it is satisfied has to do with the interplay between the speed of convergence of $\pi_{x,t}$ to $\pi$ and the growth rate of the functions $\phi_k$ and $g$. Note that the assumption allows the rate of convergence to get worse as $\|x\|_2$ increases; this is captured by the fact that the bounds are polynomial in $\|x\|_2$.

We close with a concrete illustration, related to queueing theory. Let $\{x_t | t = 0, 1, 2, \ldots\}$ be a Markov process that takes values in the nonnegative integers, and let its dynamics be

$$x_{t+1} = \max\{0,\ x_t + w_t - 1\},$$

where the $w_t$ are independent identically distributed nonnegative integer random variables with a "nice" distribution; e.g., assume that the tail of the distribution of $w_t$ asymptotically decays at an exponential rate. (This Markov chain corresponds to an M/G/1 queue which is observed at service completion times, with $w_t$ being the number of new arrivals while serving a customer.) Assuming that $\mathrm{E}[w_t] < 1$, this chain has a "downward drift," is "stable," and has a unique invariant distribution [90]. Furthermore, there exists some $\delta > 0$ such that $\pi(x) \leq e^{-x\delta}$, for $x$ sufficiently large. Let $g(x) = x$, so that the reward function basically counts the number of customers in queue. Let us introduce the basis functions $\phi_k(x) = x^k$, $k = 0, 1, 2, 3$. Then, Assumptions 4.8(1) is satisfied. Assumption 4.8(2) can be shown to be true by exploiting the downward drift property.

Let us now discuss Assumption 4.8(3). The key is again the speed of convergence of $\pi_{x,t}$ to $\pi$. Starting from a large state $x$, the Markov chain has a negative drift, and requires $O(x)$ steps to enter (with high probability) the vicinity of state 0 [71, 43].

Once the vicinity of state 0 is reached, it quickly reaches steady-state. Thus, if we concentrate on $\phi_3(x) = x^3$, the difference $\mathrm{E}[\phi_3(x_t)\phi_3'(x_{t+m})|x_0 = x] - \mathrm{E}[\phi_3(x_0)\phi_3'(x_m)]$ is of the order of $x^6$ for $O(x)$ time steps and afterwards decays at a fast rate. This suggests that Assumption 4.8(3) is satisfied.

Our discussion in the preceding example was far from rigorous. Our objective was not so much to prove that our assumptions are satisfied by specific examples, but rather to demonstrate that their content is plausible. Furthermore, while the M/G/1 queue is too simple an example, we expect that stable queueing networks that have downward drifting Lyapunov functions, should also generically satisfy our assumptions. In fact, it has been pointed out by Meyn [52] that Assumption 4.8 is weaker than certain standard notions of stochastic stability (see, e.g., [53]), such as that involving (deterministic) stability of a fluid model corresponding to a queueing system.

## 4.7 The Importance of Simulated Trajectories

In Section 3.3, we studied an iterative algorithm of the form $J_{t+1} = \Pi F J_t$. The convergence proof relied on the fact that $\Pi$ was a projection in the same Hilbert space upon which $F$ was a contraction, the consequence being that the composition $\Pi F$ was also a contraction. One possible variant of this algorithm involves a projection $\overline{\Pi}$ with respect to a different Hilbert space norm. In particular, the algorithm would take on the form $J_{t+1} = \overline{\Pi} F J_t$, where $F$ is a contraction on $L_2(\Re, \mathcal{B}(\Re), \pi)$ whereas $\overline{\Pi}$ is a projection defined with respect to the norm of a Hilbert space $L_2(\Re, \mathcal{B}(\Re), \overline{\pi})$, where the probability measure $\overline{\pi}$ is different from $\pi$. It turns out that, unlike $\Pi F$, which is guaranteed to be a contraction on $L_2(\Re, \mathcal{B}(\Re), \pi)$, $\overline{\Pi} F$ need not be a contraction on any Hilbert space, and the algorithm $J_{t+1} = \overline{\Pi} F J_t$ may even diverge. To visualize the possibility of divergence, consider the two situations illustrated in Figure 4-1. In both cases, the line represents the subspace spanned by the basis functions, and it is assumed that $J^*$ is within the span. The ellipse centered at $J^*$ represents all points as far away from $J^*$ as $J_t$ (distance is measured in terms of $\|\cdot\|_\pi$). The reason we have drawn an ellipse is that we are dealing with a weighted Euclidean norm. Since $F$ is a contraction with respect to this norm, $FJ_t$ is closer to $J^*$ than $J_t$, and is therefore inside the ellipse. Now when we project according to $\Pi$, we draw the smallest ellipse that is centered at $FJ_t$ and touches the span of the basis functions, and the point of contact gives us $J_{t+1} = \Pi F J_t$. The shape of the ellipse is determined by the norm with respect to which we project, and since $\Pi$ is a projection with respect to $\|\cdot\|_\pi$, the ellipse is similar to the first. On the other hand, since $\overline{\Pi}$ is a projection with respect to a different norm, the ellipse in the second diagram is of a different shape. Consequently, it may take us further away from $J^*$ (i.e., outside the big ellipse).

It should be evident in our analysis that the convergence of the TD($\lambda$) is intimately tied to the fact that $\Pi T^{(\lambda)}$ is a contraction. In this section, we consider a variant of TD(0) that has been proposed and implemented in the literature. This variant does not simulate an entire trajectory, and instead, it involves a sampling distribution $\overline{\pi}$ selected by a user. This algorithm turns out to be related to the composition

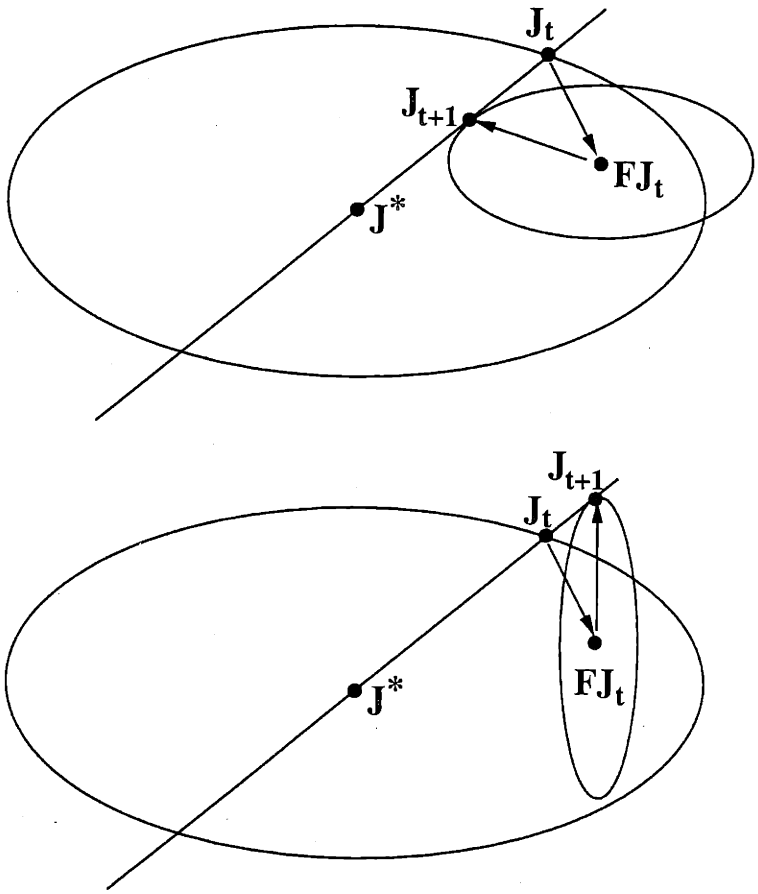


Figure 4-1: Convergence of $J_{t+1} = \Pi F J_t$ versus divergence of $J_{t+1} = \overline{\Pi} F J_t$.

of a projection operator $\overline{\Pi}$ with the operator $T^{(\lambda)}$. The latter is a contraction on $L_2(\Re, \mathcal{B}(\Re), \pi)$, whereas the projection is defined with respect to $\|\cdot\|_{\overline{\pi}}$. We will provide a result establishing that, like the deterministic algorithm $J_{t+1} = \overline{\Pi}T^{(\lambda)}J_t$, this stochastic algorithm can diverge.

Consider a variant of TD(0) where states $a_t$ are sampled independently from a distribution $\overline{\pi}$ and successor states $b_t$ are generated by sampling according to $\text{Prob}\{b_t = b|a_t = a\} = \text{Prob}\{x_{t+1} = b|x_t = a\}$. Each iteration of the algorithm takes the form

$$r_{t+1} = r_t + \gamma_t \phi(a_t)\Big(g(a_t) + \alpha\phi'(b_t)r_t - \phi'(a_t)r_t\Big).$$

Let us refer to this algorithm as $\overline{\pi}$–sampled TD(0). Note that this algorithm is closely related to the original TD(0) algorithm. In particular, if $a_t = x_t$ and $b_t = x_{t+1}$, we are back to the original algorithm. It is easy to show, using a subset of the arguments required to prove Theorem 4.9, that this algorithm converges when $\overline{\pi} = \pi$ and Assumptions 4.5–4.8 are satisfied. However, results can be very different when $\overline{\pi}$ is arbitrary. This is captured by the following theorem, which for simplicity, addresses only finite state spaces.

**Theorem 4.16** *Let $\overline{\pi}$ be a probability distribution over a finite state space $S$ with at least two elements. Let the discount factor $\alpha$ be within the open interval $(\frac{5}{6}, 1)$. Let the sequence $\{\gamma_t | t = 0, 1, 2, \ldots\}$ satisfy Assumption 4.7. Then, there exists a stochastic matrix $P$, a reward function $g(\cdot)$, and a matrix $\Phi$, such that Assumptions 4.5 and 4.6 are satisfied and execution of the $\overline{\pi}$–sampled TD(0) algorithm leads to*

$$\lim_{t\to\infty} \|\text{E}[r_t|r_0 = r]\|_2 = \infty, \qquad \forall r \neq r^*,$$

*for some unique vector $r^*$.*

**Proof:** Without loss of generality, let $S = \{1, \ldots, n\}$ and assume throughout this proof that $\overline{\pi}(1) > 0$ and $\overline{\pi}(1) \geq \overline{\pi}(2)$. We define a probability distribution $\pi$ satisfying $1 > \pi(2) > \frac{5}{6\alpha}$ and $\pi(x) > 0$ for all $x \in S$. The fact that $\alpha > \frac{5}{6}$ ensures that such a probability distribution exists. We define the transition probability matrix $P$ with each row equal to $(\pi(1), \ldots, \pi(n))$. Finally, we define the reward function to be $g(x) = 0$, for all $x$. Assumption 4.5 is trivially satisfied by our choice of $P$ and $g$, and the invariant distribution of the Markov chain is $\pi$. Note that $J^* = 0$, since rewards are zero.

Consider a single basis function given by

$$\phi(x) = \begin{cases} 1, & \text{if } x = 1, \\ 2, & \text{if } x = 2, \\ 0, & \text{otherwise.} \end{cases}$$

Hence, $r_t$ is scalar, and $\Phi$ can be thought of as an $n \times 1$ matrix. Also, Assumption 4.6 is trivially satisfied. We let $r^* = 0$, so that $J^* = \Phi r^*$.

In general, we can express $\mathrm{E}[r_t|r_0 = r]$ in terms of a recurrence of the form

$$\begin{aligned}\mathrm{E}[r_{t+1}|r_0 = r] &= \mathrm{E}[r_t|r_0 = r] + \gamma_t \mathrm{E}\Big[\phi(a_t)(g(a_t) + \alpha\phi(b_t)r_t - \phi(a_t)r_t)|r_0 = r\Big] \\ &= \mathrm{E}[r_t|r_0 = r] + \gamma_t \Phi' Q(\alpha P\Phi - \Phi)\mathrm{E}[r_t|r_0 = r],\end{aligned}$$

where $Q$ is the diagonal matrix with diagonal elements $\overline{\pi}(1), \ldots \overline{\pi}(n)$.

Specializing to our choice of parameters, the recurrence becomes

$$\begin{aligned}\mathrm{E}[r_{t+1}|r_0 = r] &= \mathrm{E}[r_t|r_0 = r] \\ &\quad + \gamma_t \Big[\overline{\pi}(1) \;\; 2\overline{\pi}(2)\Big] \left(\alpha \begin{bmatrix} \pi(1) + 2\pi(2) \\ \pi(1) + 2\pi(2) \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) \mathrm{E}[r_t|r_0 = r] \\ &= \mathrm{E}[r_t|r_0 = r] + \gamma_t \Big((\alpha(\pi(1) + 2\pi(2)) - 1)\overline{\pi}(1) \\ &\quad + 2(\alpha(\pi(1) + 2\pi(2)) - 2)\overline{\pi}(2)\Big)\mathrm{E}[r_t|r_0].\end{aligned}$$

For shorthand notation, let $\Delta$ be defined by

$$\Delta = (\alpha\pi(1) + 2\alpha\pi(2) - 1)\overline{\pi}(1) + 2(\alpha\pi(1) + 2\alpha\pi(2) - 2)\overline{\pi}(2).$$

Since $\alpha\pi(1) + 2\alpha\pi(2) < 2$ and $\overline{\pi}(1) \geq \overline{\pi}(2)$, we have

$$\begin{aligned}\Delta &\geq (\alpha\pi(1) + 2\alpha\pi(2) - 1)\overline{\pi}(1) \\ &\quad + 2(\alpha\pi(1) + 2\alpha\pi(2) - 2)\overline{\pi}(1) \\ &= (3\alpha\pi(1) + 6\alpha\pi(2) - 5)\overline{\pi}(1) \\ &\geq (6\alpha\pi(2) - 5)\overline{\pi}(1),\end{aligned}$$

and since $\pi(2) > \frac{5}{6\alpha}$, there exists some $\epsilon > 0$ such that

$$\begin{aligned}\Delta &\geq (5 + \epsilon - 5)\overline{\pi}(1) \\ &= \epsilon\overline{\pi}(1).\end{aligned}$$

It follows that

$$\|\mathrm{E}[r_{t+1}|r_0 = r]\|_2 \geq (1 + \gamma_t\epsilon\overline{\pi}(1))\|\mathrm{E}[r_t|r_0 = r]\|_2,$$

and since $\sum_{t=0}^{\infty} \gamma_t = \infty$, we have

$$\lim_{t\to\infty} \|\mathrm{E}[r_{t+1}|r_0 = r]\|_2 = \infty,$$

if $r \neq r^*$. **Q.E.D.**

Let us close this section by reflecting on the implications of this result. It demonstrates that, if the sampling distribution $\overline{\pi}$ is chosen independently of the dynamics of the Markov process, there is no convergence guarantee. Clearly, this does not imply that divergence will always occur when such a random sampling scheme is employed in practice. In fact, for any problem, there is a set of sampling distributions that lead

to convergence. Our current understanding indicates that $\pi$ is an element of this set, so it seems sensible to take advantage of this knowledge by setting $\overline{\pi} = \pi$. However, since it is often difficult to model the steady–state distribution, one generally must resort to simulation in order to generate the desired samples.

## 4.8 Divergence with Nonlinear Parameterizations

We have focused on approximations $\tilde{J}(x, r)$ that are linear in the parameter vector $r$. It is sometimes natural to consider nonlinear parameterizations, such as neural networks. In this case, we consider an extension of TD($\lambda$) that has been applied in the literature. It involves an iteration of the form

$$r_{t+1} = r_t + \gamma_t d_t z_t,$$

with the temporal–difference defined by

$$d_t = g(x_t) + \alpha \tilde{J}(x_{t+1}, r_t) - \tilde{J}(x_t, r_t),$$

and the eligibility vector defined by

$$z_t = \sum_{\tau=0}^{t} (\alpha\lambda)^{t-\tau} \nabla_r \tilde{J}(x_\tau, r_\tau).$$

Note that, when the parameterization is linear (i.e., $\tilde{J}(x, r) = \sum_{k=1}^{K} r(k)\phi_k(x)$), the gradient is given by $\nabla_r \tilde{J}(x, r) = \phi(x)$, and we obtain the original TD($\lambda$) iteration.

One might hope that our analysis generalizes to the case of nonlinear parameterizations, perhaps under some regularity conditions. Unfortunately, this does not seem to be the case. To illustrate potential difficulties, we present an example for which TD(0) diverges. (By divergence here, we mean divergence of both the approximate value function and the parameters.) For the sake of brevity, we limit our study to a characterization of steady–state dynamics, rather than presenting a rigorous proof, which would require arguments formally relating the steady–state dynamics to the actual stochastic algorithm.

We consider a Markov chain with three states ($S = \{1, 2, 3\}$), all rewards equal to zero, and a discount factor $\alpha \in (0, 1)$. The reward function $J^* \in \Re^3$ is therefore given by $J^* = (0, 0, 0)'$. Let the approximation

$$\tilde{J}(r) = \left(\tilde{J}(1, r), \tilde{J}(2, r), \tilde{J}(3, r)\right)'$$

be parameterized by a single scalar $r$. Let the form of $\tilde{J}$ be defined by letting $\tilde{J}(0)$ be some nonzero vector satisfying $e'\tilde{J}(0) = 0$, where $e = (1, 1, 1)'$, and requiring that $\tilde{J}(r)$ be the unique solution to the linear differential equation

$$\frac{d\tilde{J}}{dr}(r) = (Q + \epsilon I)\tilde{J}(r), \tag{4.1}$$

where $I$ is the $3 \times 3$ identity matrix, $\epsilon$ is a small positive constant, and $Q$ is given by

$$Q = \begin{bmatrix} 1 & 1/2 & 3/2 \\ 3/2 & 1 & 1/2 \\ 1/2 & 3/2 & 1 \end{bmatrix}.$$

We let the transition probability matrix of the Markov chain be

$$P = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 \end{bmatrix}.$$

Since all rewards are 0, the TD(0) operator is given by $T^{(0)}J = \alpha PJ$, for all $J \in \Re^3$.

The TD(0) algorithm applies the update equation

$$r_{t+1} = r_t + \gamma_t \frac{d\tilde{J}}{dr}(r) \left(\alpha \tilde{J}(x_{t+1}, r_t) - \tilde{J}(x_t, r)\right),$$

where $x_t$ is the state visited by the trajectory at time $t$. Since the steady-state distribution resulting from $P$ is uniform, the steady-state expectation of the update direction, within a factor of 3, is given by

$$\sum_{x=1}^{3} \frac{d\tilde{J}}{dr}(x, r)\Big(\alpha(P\tilde{J}(r))(x) - \tilde{J}(x, r)\Big).$$

This is the inner product of the vector $d\tilde{J}/dr$, which is $(Q + \epsilon I)\tilde{J}(r)$, with the vector $\alpha P\tilde{J}(r) - \tilde{J}(r)$.

Given the average direction of motion of the parameter $r$, the stochastic algorithm approximates an ordinary differential equation of the form

$$\begin{aligned} \frac{dr}{dt} &= \Big((Q + \epsilon I)\tilde{J}(r)\Big)' (\alpha P - I)\tilde{J}(r) \\ &= \tilde{J}'(r)(Q' + \epsilon I)(\alpha P - I)\tilde{J}(r). \end{aligned}$$

For $\epsilon = 0$, we have

$$\begin{aligned} \frac{dr}{dt} &= \tilde{J}'(r)Q'(\alpha P - I)\tilde{J}(r) \\ &= \alpha \tilde{J}'(r)Q'P\tilde{J}(r) \\ &= \frac{\alpha}{2}\tilde{J}'(r)(Q'P + P'Q)\tilde{J}(r), \end{aligned}$$

where the first equality follows from the fact that $\tilde{J}'(r)Q'\tilde{J}(r) = 0$, for any $r$. Note that

$$(Q'P + P'Q) = \begin{bmatrix} 2.5 & 1.75 & 1.75 \\ 1.75 & 2.5 & 1.75 \\ 1.75 & 1.75 & 2.5 \end{bmatrix},$$

which is easily verified to be positive definite. Hence, there exists a positive constant $c$ such that

$$\frac{dr}{dt} \geq c\|\tilde{J}(r)\|_2^2. \tag{4.2}$$

By a continuity argument, this inequality remains true (possibly with a smaller positive constant $c$) if $\epsilon$ is positive but sufficiently small. The combination of this inequality and the fact that

$$\frac{d}{dr}\|\tilde{J}(r)\|_2^2 = \tilde{J}'(r)(Q+Q')\tilde{J}(r) + 2\epsilon\|\tilde{J}(r)\|_2^2 \geq 2\epsilon\|\tilde{J}(r)\|_2^2,$$

implies that both $r$ and $\|\tilde{J}(r)\|_2$ diverge to infinity.

## 4.9 Closing Remarks

In order to place our results in perspective, let us discuss their relation to other available work. Several papers have presented positive results about TD($\lambda$). These include [72, 92, 82, 39, 25, 33], all of which only deal with cases where the number of weights is the same as the cardinality of the state space. Such cases are not practical when state spaces are large or infinite. The more general case, involving the use of linear parameterizations, is addressed by results in [24, 64, 83, 31, 69]. The latter three establish almost sure convergence. However, their results only apply to a very limited class of linear parameterizations (slightly broader than the case where basis functions are indicator functions) and only involve variants of TD(0). Dayan [24] establishes convergence in the mean for the general class of linear parameterizations. However, this form of convergence is rather weak, and the analysis used in the paper does not directly lead to approximation error bounds or interpretable characterizations of the limit of convergence. Schapire and Warmuth [64] carry out a (nonprobabilistic) worst-case analysis of an algorithm similar to temporal–difference learning. Fewer assumptions are required by their analysis, but the end results do not imply convergence and establish error bounds that are weak relative to those that can be deduced in the standard probabilistic framework.

In addition to the positive results, counterexamples to the convergence of several variants of the algorithm have been provided in several papers. These include [3, 16, 31, 83]. As suggested by Sutton [73], the key feature that distinguishes these negative results from their positive counterparts is that the variants of temporal-difference learning used do not employ on-line state sampling. In particular, as in the variant discussed in Section 4.7, sampling is done by a mechanism that samples states with frequencies independent from the dynamics of the underlying system. Our results shed light on these counterexamples by showing that, for linear parameterizations, convergence is guaranteed if states are sampled according to the steady-state probabilities, while divergence is possible when states are sampled from distributions independent of the dynamics of the Markov process of interest.

The results of this chapter have appeared previously in a journal article [84]. Around the same time as the original submission of that article, Gurvits [32] inde-

pendently established almost sure convergence in the context of absorbing Markov chains. Also, Pineda [57] derived a stable differential equation for the "mean field" of temporal–difference learning, in the case of finite state absorbing Markov chains, and suggested a convergence proof based on a weighted maximum norm contraction property, which, however, is not satisfied in the presence of function approximation. (The proof was corrected after the paper [84] became available.)

The sensitivity of the error bound to $\lambda$ raises the question of whether or not it ever makes sense to set $\lambda$ to values less than 1. Experimental results [72, 70, 74] suggest that setting $\lambda$ to values less than one can often lead to significant gains in the rate of convergence. A full understanding of how $\lambda$ influences the rate of convergence is yet to be found. Furthermore, it might be desirable to tune $\lambda$ as the algorithm progresses, possibly initially starting with $\lambda = 0$ and approaching $\lambda = 1$ (although the opposite has also been advocated). These are interesting directions for future research.

In applications of controlled TD (as presented in Chapter 2), one deals with a controlled Markov process and at each stage a decision is "greedily" chosen, by minimizing the right-hand side of Bellman's equation, and using the available approximation $\tilde{J}$ in place of $J^*$. Our analysis does not apply to such cases involving changing policies. Of course, if the policy eventually settles into a limiting policy, we are back to the autonomous case and convergence is obtained. However, there exist examples for which the policy does not converge [13]. It remains an open problem to analyze the limiting behavior of the weights $r_t$ and the resulting approximations $\Phi r_t$ for the case where the policy does not converge.

On the technical side, we mention a few straightforward extensions of our results. First, the linear independence of the basis functions is not essential. In the linearly dependent case, some components of $z_t$ and $r_t$ become linear combinations of the other components and can be simply eliminated, which takes us back to the linearly independent case. A second extension is to allow the reward per stage $g(x_t)$ to be noisy and dependent on $x_{t+1}$, as opposed to being a deterministic function of $x_t$. Our line of analysis easily extends to this case. Finally, the mixing requirement can be weakened. In particular, it is sufficient to have an ergodic Markov process.

Our results in Section 4.7 have elucidated the importance of sampling states according to the steady-state distribution of the process under consideration. In particular, variants of TD($\lambda$) that sample states differently can lead to divergence. It is interesting to note that a related issue arises when one "plays" with the evolution equation for the eligibility vector $z_t$. (For example Singh and Sutton [70] have suggested an alternative evolution equation for $z_t$ known as the "replace trace.") A very general class of such mechanisms can be shown to lead to convergent algorithms for the case of lookup table representations [13]. However, different mechanisms for adjusting the coefficients $z_t$ lead to a change in the steady-state average value of $z_t\phi'(x_t)$, and stability of the related ordinary differential equation can be lost.

The example of Section 4.8 identifies the possibility of divergence when TD($\lambda$) is used in conjunction with nonlinear parameterizations. However, the example is somewhat contrived, and it is unclear whether divergence can occur with special classes of nonlinear parameterizations, such as neural networks. This presents an interesting question for future research.

# Chapter 5

# Approximations for Optimal Stopping

The problem of optimal stopping is that of determining an appropriate time at which to terminate a process in order to maximize expected rewards. Examples arise in sequential analysis, the timing of a purchase or sale of an asset, and the analysis of financial derivatives.

In this chapter, we introduce a few classes of optimal stopping problems. In principle, these problems can be solved by classical dynamic programming algorithms. However, the curse of dimensionality prohibits the viability of such exact methods. To address this limitation, we introduce variants of temporal–difference learning that approximate value functions in order to solve optimal stopping problems.

One section is dedicated to each class of problems that we address. The problem classes involve optimal stopping in a few different contexts:

1. stationary mixing processes;
2. independent increments processes;
3. finite horizons;
4. two–player zero–sum games.

In each case, we establish that for any problem in the class, the associated optimal value function exists and is unique. Shiryaev [66] provides a far more comprehensive treatment of optimal stopping problems where, under each of a sequence of increasingly general assumptions, he characterizes optimal value functions and stopping times. We consider a rather restrictive classes of problems relative to those captured by Shiryaev's analysis, but we employ a new line of analysis that relies on different sorts of assumptions and leads to simple characterizations of optimal value functions and stopping times.

The most important aspect of our line of analysis is that it accommodates the development of approximation methods along the lines of temporal–difference learning. For each class of problems, we introduce an approximation algorithm and study its converges properties. We also supply bounds on both the error in approximating

the optimal value function and the difference between performance of the optimal stopping time and that generated by the approximation.

Our first class of problems deals with the optimal stopping of a process that is Markov, stationary, and mixing. We will provide a detailed analysis of this class of problems and the associated approximation algorithm. In subsequent sections, we introduce additional classes of optimal stopping problems. For the sake of brevity, instead of presenting rigorous analyses, we only overview how the analysis provided for the first class of problems can be adapted to suit each situation.

## 5.1 Stationary Mixing Processes

Our first class of problems involves processes that are Markov, stationary, and mixing, like those studied in Chapter 4, in the context of temporal–difference learning. We begin by defining the problem and characterizing its solution in terms of an optimal value function and stopping time. The approximation algorithm is then presented and analyzed in Section 5.1.2.

### 5.1.1 Problem Formulation and Solution

Consider a stochastic process $\{x_t | t = 0, 1, 2, \ldots\}$ taking on values in a state space $\Re^d$ defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$. We denote the $\sigma$–field generated by random variables $x_0, \ldots, x_t$ by $\mathcal{F}_t$, and we take $\mathcal{F}$ to be the smallest $\sigma$–field containing $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$ as sub–$\sigma$–fields. We make the following assumption concerning the dynamics of this process.

**Assumption 5.1** *The process $\{x_t | t = 0, 1, 2, \ldots\}$ is Markov, stationary, and mixing.*

Because the process is stationary, there is a distribution $\pi$ such that $\pi(A) = \mathcal{P}\{x_t \in A\}$ for any set $A \in \mathcal{B}(\Re^d)$ and any nonnegative integer $t$. This distribution is also the limiting distribution that is guaranteed to exist by the mixing property. In particular, $\lim_{t\to\infty} \mathcal{P}\{x_t \in A | x_0 = x\} = \pi(A)$ for any $x \in \Re^d$ and $A \in \mathcal{B}(\Re^d)$. Like in the context of temporal–difference learning, a central object in our analysis is the Hilbert space $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, which is endowed with an inner product $\langle J_1, J_2 \rangle_\pi = \int J_1(x) J_2(x) \pi(dx)$ and a norm $\|J\|_\pi = \langle J, J \rangle_\pi^{1/2}$.

We define a stopping time to be a random variable $\tau$ that takes on values in $\{0, 1, 2, \ldots, \infty\}$ and satisfies $\{\omega \in \Omega | \tau(\omega) \leq t\} \in \mathcal{F}_t$ for all finite $t$. The set of all such random variables is denoted by $\mathcal{T}$. Since we have defined $\mathcal{F}_t$ to be the $\sigma$–algebra generated by $\{x_0, x_1, \ldots, x_t\}$, the stopping time is determined solely by the already available samples of the stochastic process. In particular, we do not consider stopping times that may be influenced by random events other than the stochastic process itself. This preclusion is not necessary for our analysis, but it is introduced to simplify the exposition.

Let $g \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ and $G \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ be reward functions associated with "continuation" and "termination," and let $\alpha \in [0, 1)$ be a discount factor. The

expected reward associated with a stopping time $\tau$ is defined by

$$\mathrm{E}\left[\sum_{t=0}^{\tau-1}\alpha^t g(x_t)+\alpha^\tau G(x_\tau)\right],$$

where $G(x_\tau)$ is taken to be 0 if $\tau=\infty$. An optimal stopping time $\tau^*$ is one that satisfies

$$\mathrm{E}\left[\sum_{t=0}^{\tau^*-1}\alpha^t g(x_t)+\alpha^{\tau^*} G(x_{\tau^*})\right]=\sup_{\tau\in\mathcal{T}}\mathrm{E}\left[\sum_{t=0}^{\tau-1}\alpha^t g(x_t)+\alpha^\tau G(x_\tau)\right].$$

We will provide a theorem that characterizes value functions and optimal stopping times for the class of problems under consideration. Before doing so, let us introduce some useful notation. By the Markov property, there exists a transition probability function $P$ on $\Re^d\times\mathcal{B}(\Re^d)$ such that, for any $A\in\mathcal{B}(\Re^d)$,

$$\mathcal{P}\{x_{t+1}\in A|\mathcal{F}_t\}=P(x_t,A),$$

and $P(\cdot,A)$ is measurable. We will also use $P$ to denote an operator given by

$$(PJ)(x)=\int J(y)P(x,dy).$$

We define an operator $T$ by

$$TJ=\max\{G,g+\alpha PJ\},$$

where the max denotes pointwise maximization. This is the so-called "dynamic programming operator," specialized to the case of an optimal stopping problem. To each stopping time $\tau$, we associate a value function $J^\tau$ defined by

$$J^\tau(x)=\mathrm{E}\left[\sum_{t=0}^{\tau-1}\alpha^t g(x_t)+\alpha^\tau G(x_\tau)\Big|x_0=x\right].$$

The fact that $g$ and $G$ are in $L_2(\Re^d,\mathcal{B}(\Re^d),\pi)$ implies that $J^\tau$ is also an element of $L_2(\Re^d,\mathcal{B}(\Re^d),\pi)$ for any $\tau\in\mathcal{T}$. Hence, a stopping time $\tau^*$ is optimal if and only if

$$\mathrm{E}[J^{\tau^*}(x_0)]=\sup_{\tau\in\mathcal{T}}\mathrm{E}[J^\tau(x_0)].$$

It is not hard to show that optimality in this sense corresponds to pointwise optimality for all elements $x$ of some set $A$ with $\pi(A)=1$. However, this fact will not be used in our analysis.

The main results of this section are captured by the following theorem:

**Theorem 5.2** *Under Assumptions 5.1,*

1. *There exists a function* $J^* \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ *uniquely satisfying*

$$J^* = TJ^*.$$

2. *The stopping time* $\tau^*$*, defined by*

$$\tau^* = \min\{t | G(x_t) \geq J^*(x_t)\},$$

*is an optimal stopping time. (The minimum of an empty set is taken to be* $\infty$*.)*

3. $J^{\tau^*} = J^*$.

As in previous chapters, the equality of functions is in the sense of $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$.

In the remainder of this section, we provide a proof of the theorem. We begin by proving a few lemmas.

### Preliminary Lemmas

Recall that, under Assumption 5.1, Lemma 4.12 states that $\|PJ\|_\pi \leq \|J\|_\pi$ for all $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. The following lemma uses this fact to establish that $T$ is a contraction.

**Lemma 5.3** *Under Assumptions 4.5,*

$$\|TJ_1 - TJ_2\|_\pi \leq \alpha \|J_1 - J_2\|_\pi,$$

*for all* $J_1, J_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$.

**Proof:** For any scalars $c_1$, $c_2$, and $c_3$,

$$|\max\{c_1, c_3\} - \max\{c_2, c_3\}| \leq |c_1 - c_2|.$$

It follows that for any $x \in \Re^d$ and $J_1, J_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$|(TJ_1)(x) - (TJ_2)(x)| \leq \alpha |(PJ_1)(x) - (PJ_2)(x)|.$$

Given this fact, the result easily follows from Lemma 4.12. **Q.E.D.**

The fact that $T$ is a contraction implies that it has a unique fixed point. Let $J^*$ denote the fixed point of $T$. Let us define a second operator $T^*$ by

$$T^*J = \begin{cases} G(x), & \text{if } G(x) \geq J^*(x), \\ g(x) + (\alpha PJ)(x), & \text{otherwise.} \end{cases}$$

(Note that $T^*$ is the dynamic programming operator corresponding to the case of a fixed policy, namely, the policy corresponding to the stopping time $\tau^*$ defined in the statement of the above theorem.) The following lemma establishes that $T^*$ is also a contraction, and furthermore, the fixed point of this contraction is equal to $J^*$.

**Lemma 5.4** *Under Assumptions 4.5,*

$$\|T^*J_1 - T^*J_2\|_\pi \leq \alpha\|J_1 - J_2\|_\pi,$$

*for all* $J_1, J_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. *Furthermore,* $J^* \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ *is the unique fixed point of* $T^*$.

**Proof:** For any $J_1, J_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$\begin{aligned} \|T^*J_1 - T^*J_2\|_\pi &\leq \|\alpha P J_1 - \alpha P J_2\|_\pi \\ &\leq \alpha\|J_1 - J_2\|_\pi, \end{aligned}$$

where the final inequality follows from Lemma 4.12.

Recall that $J^*$ uniquely satisfies $J^* = TJ^*$, or written differently,

$$J^* = \max\{G, g + \alpha P J^*\}.$$

This equation can also be rewritten as

$$J^*(x) = \begin{cases} G(x), & \text{if } G(x) \geq g(x) + (\alpha P J^*)(x), \\ g(x) + (\alpha P J)(x), & \text{otherwise.} \end{cases}$$

Note that for almost all $x$, $G(x) \geq g(x) + (\alpha P J^*)(x)$ if and only if $G(x) = J^*(x)$. Hence, $J^*$ satisfies

$$J^*(x) = \begin{cases} G(x), & \text{if } G(x) \geq J^*(x), \\ g(x) + (\alpha P J)(x), & \text{otherwise.} \end{cases}$$

or more concisely, $J^* = T^*J^*$. Since $T^*$ is a contraction, it has a unique fixed point, and this fixed point is $J^*$. **Q.E.D.**

### Proof of Theorem 1

Part (1) of the result follows from Lemma 5.3. As for Part (3),

$$\begin{aligned} J^{\tau^*}(x) &= \begin{cases} G(x), & \text{if } G(x) \geq J^*(x), \\ g(x) + (\alpha P J^{\tau^*})(x), & \text{otherwise,} \end{cases} \\ &= (T^*J^{\tau^*})(x), \end{aligned}$$

and since $T^*$ is a contraction with fixed point $J^*$ (Lemma 5.4), it follows that

$$J^{\tau^*} = J^*.$$

We are left with the task of proving Part (2). For any nonnegative integer $n$,

$$\sup_{\tau \in \mathcal{T}} \mathrm{E}[J^\tau(x_0)] \leq \sup_{\tau \in \mathcal{T}} \mathrm{E}[J^{\tau \wedge n}(x_0)] + \mathrm{E}\left[\sum_{t=n}^{\infty} \alpha^t (|g(x_t)| + |G(x_t)|)\right]$$

$$
\begin{aligned}
&= \sup_{\tau \in \mathcal{T}} \mathrm{E}[J^{\tau \wedge n}(x_0)] + \frac{\alpha^n}{1-\alpha} \mathrm{E}\left[|g(x_0)| + |G(x_0)|\right] \\
&\leq \sup_{\tau \in \mathcal{T}} \mathrm{E}[J^{\tau \wedge n}(x_0)] + \alpha^n C,
\end{aligned}
$$

for some scalar $C$ that is independent of $n$, where the equality follows from the Tonelli–Fubini theorem and stationarity. By arguments standard to the theory of finite–horizon dynamic programming,

$$
\sup_{\tau \in \mathcal{T}} J^{\tau \wedge n}(x) = (T^n G)(x), \qquad \forall x \in \Re^d.
$$

(This equality is simply saying that the optimal reward for an $n$–horizon problem is obtained by applying $n$ iterations of the dynamic programming recursion.) It is easy to see that $T^n G$, and therefore also $\sup_{\tau \in \mathcal{T}} J^{\tau \wedge n}(\cdot)$, is measurable. It follows that

$$
\sup_{\tau \in \mathcal{T}} \mathrm{E}[J^{\tau \wedge n}(x_0)] \leq \mathrm{E}\left[\sup_{\tau \in \mathcal{T}} J^{\tau \wedge n}(x_0)\right] = \mathrm{E}[(T^n G)(x_0)].
$$

Combining this with the bound on $\sup_{\tau \in \mathcal{T}} \mathrm{E}[J^{\tau}(x_0)]$, we have

$$
\sup_{\tau \in \mathcal{T}} \mathrm{E}[J^{\tau}(x_0)] \leq \mathrm{E}[(T^n G)(x_0)] + \alpha^n C.
$$

Since $T$ is a contraction on $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ (Lemma 5.3), $T^n G$ converges to $J^*$ in the sense of $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. It follows that

$$
\lim_{n \to \infty} \mathrm{E}[(T^n G)(x_0)] = \mathrm{E}[J^*(x_0)],
$$

and we therefore have

$$
\sup_{\tau \in \mathcal{T}} \mathrm{E}[J^{\tau}(x_0)] \leq \lim_{n \to \infty} \mathrm{E}[(T^n G)(x_0)] = \mathrm{E}[J^*(x_0)] = \mathrm{E}[J^{\tau^*}(x_0)].
$$

Hence, the stopping time $\tau^*$ is optimal. **Q.E.D.**

### 5.1.2 Approximation Algorithm

As shown in the previous section, the value function $J^*$ can be used to generate an optimal stopping time. However, computation and storage of $J^*$ becomes intractable when state spaces are large or infinite. In this section we develop an approximation algorithm that is amenable to such situations. Instead of the value function $J^*$, our algorithm approximates the $Q$–function, as discussed in Chapter 2. Recall that the $Q$–function has as its domain the product of the state and decision spaces. In optimal stopping, there are two possible decisions: "stop" or "continue." The $Q$–function for an optimal stopping problem would map a state $x$ and the stopping decision to the reward contingent on stopping, which is $G(x)$. On the other hand, a state $x$ and the continuation decision would map to the optimal reward starting at $x$

contingent on continuing for one time step. Since $G(x)$ is readily available, we need only concern ourselves with approximating the $Q$–function over the portion of the domain corresponding to continuation decisions. Accordingly, we define a function $Q^* : \Re^d \mapsto \Re$ by

$$Q^* = g + \alpha P J^*. \tag{5.1}$$

An optimal stopping time can then be generated according to

$$\tau^* = \min\{t | G(x_t) \geq Q^*(x_t)\}.$$

We will consider approximations comprised of linear combinations of basis functions $\phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. As in previous chapters, let $\Phi : \Re^K \mapsto L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ be defined by $\Phi r = \sum_{k=1}^K r(k)\phi_k$, and let $\phi : \Re^d \mapsto \Re^K$ be defined by $\phi(x) = (\phi_1(x), \ldots, \phi_K(x))'$. Also, let $\Pi$ be the operator that projects in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ onto the span of the basis functions.

The algorithm is initialized with a weight vector $r_0 \in \Re^K$. During the simulation of a trajectory $\{x_t | t = 0, 1, 2, \ldots\}$ of the Markov chain, the algorithm generates a sequence $\{r_t | t = 1, 2, \ldots\}$ according to

$$r_{t+1} = r_t + \gamma_t \phi(x_t)\Big(g(x_t) + \alpha \max\{(\Phi r_t)(x_{t+1}), G(x_{t+1})\} - (\Phi r_t)(x_t)\Big), \tag{5.2}$$

where each $\gamma_t$ is a positive scalar step size. We will prove that, under certain conditions, the sequence $r_t$ converges to a vector $r^*$, and $\Phi r^*$ approximates $Q^*$. Furthermore, the stopping time $\tilde{\tau}$, given by

$$\tilde{\tau} = \min\{t | G(x_t) \geq (\Phi r^*)(x_t)\},$$

approximates the performance of $\tau^*$.

Let us now introduce our assumptions so that we can formally state results concerning the approximation algorithm. Our first assumption pertains to the basis functions.

**Assumption 5.5** *The basis functions $\phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ are linearly independent.*

Our next assumption requires that the Markov chain exhibits a certain "degree of stability" and that certain functions do not grow to quickly. (We use $\|\cdot\|_2$ to denote the Euclidean norm on finite–dimensional spaces.)

**Assumption 5.6** *The following conditions hold:*

1. *For any positive scalar $q$, there exists a scalar $\mu_q$ such that for all $x$ and $t$,*

$$\mathrm{E}[1 + \|x_t\|_2^q | x_0 = x] \leq \mu_q(1 + \|x\|_2^q).$$

2. *There exist scalars $C_1, q_1$ such that, for any function $J$ satisfying $|J(x)| \leq$*

$C_2(1 + \|x\|_2^{q_2})$, *for some scalars* $C_2$ *and* $q_2$,

$$\sum_{t=0}^{\infty} \left| \mathrm{E}[J(x_t)|x_0 = x] - \mathrm{E}[J(x_0)] \right| \leq C_1 C_2 (1 + \|x\|_2^{q_1 q_2}), \qquad \forall x \in \Re^d.$$

3. *There exist scalars* $C$ *and* $q$ *such that for all* $x \in \Re^d$, $|g(x)| \leq C(1 + \|x\|_2^q)$, $|G(x)| \leq C(1 + \|x\|_2^q)$, *and* $\|\phi(x)\|_2 \leq C(1 + \|x\|_2^q)$.

Our final assumption places constraints on the sequence of step sizes.

**Assumption 5.7** *The sequence* $\{\gamma_t | t = 0, 1, 2, \ldots\}$ *is prespecified (deterministic), nonincreasing, and satisfies*

$$\sum_{t=0}^{\infty} \gamma_t = \infty \qquad \text{and} \qquad \sum_{t=0}^{\infty} \gamma_t^2 < \infty.$$

We define an additional operator $F$ by

$$FQ = g + \alpha P \max\{G, Q\}, \tag{5.3}$$

for all $Q \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$.

The main result of this section follows:

**Theorem 5.8** *Under Assumptions 5.1–5.7,*

1. *The sequence* $\{r_t | t = 0, 1, 2, \ldots\}$ *almost surely converges.*

2. *The limit of convergence* $r^*$ *is the unique solution of the equation*

$$\Pi F(\Phi r^*) = \Phi r^*.$$

3. *The limit of convergence* $r^*$ *satisfies*

$$\|\Phi r^* - Q^*\|_\pi \leq \frac{1}{\sqrt{1 - \kappa^2}} \|\Pi Q^* - Q^*\|_\pi,$$

*where* $\kappa$ *is the contraction factor of* $\Pi F$ *and satisfies* $\kappa \leq \alpha$.

4. *Let* $\tilde{\tau}$ *be defined by*

$$\tilde{\tau} = \min\{t | G(x_t) \geq (\Phi r^*)(x_t)\}.$$

*Then,*

$$\mathrm{E}[J^*(x_0)] - \mathrm{E}[J^{\tilde{\tau}}(x_0)] \leq \frac{2}{(1 - \alpha)\sqrt{1 - \kappa^2}} \|\Pi Q^* - Q^*\|_\pi.$$

Note that the bounds provided by statements (3) and (4) involve a term $\|\Pi Q^* - Q^*\|_\pi$. This term represents the smallest approximation error (in terms of $\|\cdot\|_\pi$) that can be achieved given the choice of basis functions. Hence, as the subspace spanned by the basis functions comes closer to $Q^*$, the error generated by the algorithm diminishes to zero and the performance of the resulting stopping time approaches optimality.

The remainder of this section focuses on proving Theorem 5.8. We begin by proving some preliminary lemmas, which are later integrated with the machinery from Chapter 3 to prove the theorem.

### Preliminary Lemmas

We begin with a lemma establishing that $F$ is a contraction in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ and that $Q^*$ is its fixed point.

**Lemma 5.9** *Under Assumptions 5.1, the operator $F$ satisfies*

$$\|FQ_1 - FQ_2\|_\pi \leq \alpha \|Q_1 - Q_2\|_\pi, \qquad \forall Q_1, Q_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi).$$

*Furthermore, $Q^*$ is the unique fixed point of $F$ in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$.*

**Proof:** For any $Q_1, Q_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, we have

$$\begin{aligned} \|FQ_1 - FQ_2\|_\pi &= \alpha \|P \max\{G, Q_1\} - P \max\{G, Q_2\}\|_\pi \\ &\leq \alpha \|\max\{G, Q_1\} - \max\{G, Q_2\}\|_\pi \\ &\leq \alpha \|Q_1 - Q_2\|_\pi, \end{aligned}$$

where the first inequality follows from Lemma 4.12 and the second makes use of the fact that

$$|\max\{c_1, c_3\} - \max\{c_2, c_3\}| \leq |c_1 - c_2|,$$

for any scalars $c_1$, $c_2$, and $c_3$. Hence, $F$ is a contraction on $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. It follows that $F$ has a unique fixed point. By Theorem 5.2, we have

$$\begin{aligned} J^* &= TJ^*, \\ g + \alpha P J^* &= g + \alpha P \max\{G, g + \alpha P J^*\}, \\ Q^* &= g + \alpha P \max\{G, Q^*\}, \\ Q^* &= FQ^*, \end{aligned}$$

and therefore, $Q^*$ is the fixed point. **Q.E.D.**

Since $F$ is a contraction, by Theorem 3.9, the composition $\Pi F$ is a contraction with contraction factor $\kappa \leq \alpha$, and it has a unique fixed point. Since the basis functions are linearly independent, there is a unique vector $r^* \in \Re^K$ such that the fixed point is given by $\Phi r^*$. Furthermore, by Theorem 3.9, $r^*$ satisfies

$$\|\Phi r^* - Q^*\|_\pi \leq \frac{1}{\sqrt{1 - \kappa^2}} \|\Pi Q^* - Q^*\|_\pi.$$

Let the stopping time $\tilde{\tau} \in \mathcal{T}$ be defined by $\tilde{\tau} = \min\{t | G(x_t) \geq (\Phi r^*)(x_t)\}$. Let us define operators $H$ and $\tilde{F}$ by

$$(HQ)(x) = \begin{cases} G(x), & \text{if } G(x) \geq (\Phi r^*)(x), \\ Q(x), & \text{otherwise,} \end{cases}$$

and

$$\tilde{F}Q = g + \alpha PHQ. \tag{5.4}$$

The next lemma establishes that $\tilde{F}$ is a contraction on $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ with a fixed point $\tilde{Q} = g + \alpha PJ^{\tilde{\tau}}$.

**Lemma 5.10** *Under Assumptions 5.1, and 4.6, for any $Q_1, Q_2 \in L_2(\Re^d, \mathcal{B}(\Re^d); \pi)$,*

$$\|\tilde{F}Q_1 - \tilde{F}Q_2\|_\pi \leq \alpha \|Q_1 - Q_2\|_\pi.$$

*Furthermore, $\tilde{Q} = g + \alpha PJ^{\tilde{\tau}}$ is the unique fixed point of $\tilde{F}$.*

**Proof:** For any $Q_1, Q_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, we have

$$\begin{aligned}
\|\tilde{F}Q_1 - \tilde{F}Q_2\|_\pi &= \|(g + \alpha PHQ_1) - (g + \alpha PHQ_2)\|_\pi \\
&\leq \alpha \|HQ_1 - HQ_2\|_\pi \\
&\leq \alpha \|\max\{G, Q_1 - Q_2\}\|_\pi \\
&\leq \alpha \|Q_1 - Q_2\|_\pi,
\end{aligned}$$

where the first inequality follows from Lemma 4.12.

To prove that $\tilde{Q} = g + \alpha PJ^{\tilde{\tau}}$ is the fixed point, observe that

$$\begin{aligned}
(H\tilde{Q})(x) &= (H(g + \alpha PJ^{\tilde{\tau}}))(x) \\
&= \begin{cases} G(x), & \text{if } G(x) \geq (\Phi r^*)(x), \\ \tilde{Q}(x), & \text{otherwise,} \end{cases} \\
&= \begin{cases} G(x), & \text{if } G(x) \geq (\Phi r^*)(x), \\ g(x) + (\alpha PJ^{\tilde{\tau}})(x), & \text{otherwise,} \end{cases} \\
&= J^{\tilde{\tau}}(x).
\end{aligned}$$

Therefore,

$$\tilde{F}\tilde{Q} = g + \alpha PH\tilde{Q} = g + \alpha PJ^{\tilde{\tau}} = \tilde{Q},$$

as desired. **Q.E.D.**

The next lemma places a bound on the loss in performance incurred when using the stopping time $\tilde{\tau}$ instead of an optimal stopping time.

**Lemma 5.11** *Under Assumptions 5.1 and 5.5, the stopping time $\tilde{\tau}$ satisfies*

$$\mathrm{E}[J^*(x_0)] - \mathrm{E}[J^{\tilde{\tau}}(x_0)] \leq \frac{2}{(1-\alpha)\sqrt{1-\kappa^2}} \|\Pi Q^* - Q^*\|_\pi,$$

*where $\kappa$ is the contraction factor of $\Pi F$ and satisfies $\kappa \leq \alpha$.*

**Proof:** By stationarity and Jensen's inequality, we have

$$\begin{aligned}
\mathrm{E}[J^*(x_0)] - \mathrm{E}[J^{\tilde{\tau}}(x_0)] &= \mathrm{E}[(PJ^*)(x_0)] - \mathrm{E}[(PJ^{\tilde{\tau}})(x_0)] \\
&\leq \left|\mathrm{E}[(PJ^*)(x_0)] - \mathrm{E}[(PJ^{\tilde{\tau}})(x_0)]\right| \\
&\leq \|PJ^* - PJ^{\tilde{\tau}}\|_\pi.
\end{aligned}$$

Recall that $Q^* = g + \alpha P J^*$ and $\tilde{Q} = g + \alpha P J^{\tilde{r}}$. We therefore have

$$\begin{aligned} \mathrm{E}[J^*(x_0)] - \mathrm{E}[J^{\tilde{r}}(x_0)] &\leq \frac{1}{\alpha}\|(g + \alpha P J^*) - (g + \alpha P J^{\tilde{r}})\|_\pi \\ &= \frac{1}{\alpha}\|Q^* - \tilde{Q}\|_\pi. \end{aligned}$$

Hence, it is sufficient to place a bound on $\|Q^* - \tilde{Q}\|_\pi$.

It is easy to show that $F(\Phi r^*) = \tilde{F}(\Phi r^*)$ (compare definitions (5.3) and (5.4)). Using this fact, the triangle inequality, the equality $FQ^* = Q^*$ (Lemma 3.9), and the equality $\tilde{F}\tilde{Q} = \tilde{Q}$ (Lemma 5.10), we have

$$\begin{aligned} \|Q^* - \tilde{Q}\|_\pi &\leq \|Q^* - F(\Phi r^*)\|_\pi + \|\tilde{Q} - \tilde{F}(\Phi r^*)\|_\pi \\ &\leq \alpha\|Q^* - \Phi r^*\|_\pi + \alpha\|\tilde{Q} - \Phi r^*\|_\pi \\ &\leq 2\alpha\|Q^* - \Phi r^*\|_\pi + \alpha\|Q^* - \tilde{Q}\|_\pi, \end{aligned}$$

and it follows that

$$\begin{aligned} \|Q^* - \tilde{Q}\|_\pi &\leq \frac{2\alpha}{1-\alpha}\|Q^* - \Phi r^*\|_\pi \\ &\leq \frac{2\alpha}{(1-\alpha)\sqrt{1-\kappa^2}}\|Q^* - \Pi Q^*\|_\pi, \end{aligned}$$

where the final inequality follows from Theorem 3.9. Finally, we obtain

$$\mathrm{E}[J^*(x_0)] - \mathrm{E}[J^{\tilde{r}}(x_0)] \leq \frac{2}{(1-\alpha)\sqrt{1-\kappa^2}}\|\Pi Q^* - Q^*\|_\pi.$$

**Q.E.D.**

### Proof of the Theorem

We now continue by casting our algorithm as one amenable to Theorem 3.10. Let us define a stochastic process $\{y_t | t = 0, 1, 2, \ldots\}$ taking on values in $\Re^{2d}$ where $y_t = (x_t, x_{t+1})$. It is easy to see that this process is Markov. Furthermore, the iteration given by Equation (5.2) can be rewritten as

$$r_{t+1} = r_t + \gamma_t s(y_t, r_t),$$

for a function

$$s(y, r) = \phi(x)\Big(g(x) + \alpha \max\{(\Phi r)(\overline{x}), G(\overline{x})\} - (\Phi r)(x)\Big),$$

for any $r \in \Re^K$ and $y = (x, \overline{x})$. We define a function $\overline{s} : \Re^K \mapsto \Re^K$ by

$$\overline{s}(r) = \mathrm{E}[s(y_0, r)].$$

(Note that this is an expectation over $y_0$ for a fixed $r$. It is easy to show that the random variable $s(y_0, r)$ is absolutely integrable and $\overline{s}(r)$ is well–defined as a consequence of Assumption 5.6.) Note that each component $\overline{s}_k(r)$ can be represented in terms of an inner product according to

$$\begin{aligned}
\overline{s}_k(r) &= \mathrm{E}\Big[\phi_k(x_0)\Big(g(x_0) + \alpha \max\{(\Phi r)(x_1), G(x_1)\} - (\Phi r)(x_0)\Big)\Big] \\
&= \mathrm{E}\Big[\phi_k(x_0)\Big(g(x_0) + \alpha \mathrm{E}[\max\{(\Phi r)(x_1), G(x_1)\}|x_0] - (\Phi r)(x_0)\Big)\Big] \\
&= \mathrm{E}\Big[\phi_k(x_0)\Big(g(x_0) + \alpha P \max\{\Phi r, G\}(x_0) - (\Phi r)(x_0)\Big)\Big] \\
&= \Big\langle \phi_k, F\Phi r - \Phi r \Big\rangle_\pi,
\end{aligned}$$

where the definition of the operator $P$ is used.

We will prove convergence by establishing that each conditions of Theorem 3.10 is valid.

1. Since $F$ is a contraction (Lemma 5.9) and the basis functions are linearly independent (Assumption 5.5), this condition follows from Theorem 3.11 for the function $\overline{s}$ defined above.

2. This condition follows immediately from Assumption 5.6.

3. To establish validity of this condition, for any $r$ and $y = (x, \overline{x})$, we have

$$\begin{aligned}
\|s(y,r)\|_2 &= \Big\|\phi(x)\Big(g(x) + \alpha \max\{(\Phi r)(\overline{x}), G(\overline{x})\} - (\Phi r)(x)\Big)\Big\|_2 \\
&\leq \|\phi(x)\|_2\Big(|g(x)| + \alpha(\|\phi(\overline{x})\|_2\|r\|_2 + |G(\overline{x})|) + \|\phi(x)\|_2\|r\|_2\Big) \\
&\leq \|\phi(x)\|_2(|g(x)| + \alpha|G(\overline{x})|) + \|\phi(x)\|_2(\alpha\|\phi(\overline{x})\|_2 + \|\phi(x)\|_2)\|r\|_2.
\end{aligned}$$

   The condition then easily follows from the polynomial bounds of Assumption 5.6(3).

4. Given that the previous condition is valid, this condition follows from Assumptions 5.6(1) and 5.6(2) in a straightforward manner. (Using these assumptions, it is easy to show that a condition analogous to Assumption 5.6(2) holds for functions of $y_t = (x_t, x_{t+1})$ that are bounded by polynomials in $x_t$ and $x_{t+1}$.)

5. We first note that for any $r$, $\overline{r}$, and $y$, we have

$$\begin{aligned}
\|s(y,r) - s(y,\overline{r})\|_2 &= \Big\|\phi(x)\Big(\alpha \max\{(\Phi r)(y), G(y)\} - \alpha \max\{(\Phi \overline{r})(y), G(y)\} \\
&\quad -(\Phi r)(x) + (\Phi \overline{r})(x)\Big)\Big\|_2 \\
&\leq \alpha\|\phi(x)\|_2\Big|\max\{\phi'(y)r, G(y)\} - \max\{\phi'(y)\overline{r}, G(y)\}\Big| \\
&\quad +\|\phi(x)\|_2|\phi'(x)r - \phi'(x)\overline{r}| \\
&\leq \alpha\|\phi(x)\|_2|\phi'(y)r - \phi'(y)\overline{r}| + \|\phi(x)\|_2^2\|r - \overline{r}\|_2 \\
&\leq \alpha\|\phi(x)\|_2\|\phi(y)\|_2\|r - \overline{r}\|_2 + \|\phi(x)\|_2^2\|r - \overline{r}\|_2.
\end{aligned}$$

It then follows from the polynomial bounds of Assumption 5.6(3) that there exist scalars $C_2$ and $q_2$ such that for any $r$, $\overline{r}$, and $y$,

$$\|s(y,r) - s(y,\overline{r})\|_2 \leq C_2\|r - \overline{r}\|_2(1 + \|y\|_2^{q_2}).$$

Finally, it follows from Assumptions 5.6(1) and 5.6(2) that there exist scalars $C_1$ and $q_1$ such that

$$\sum_{t=0}^{\infty} \left\| \mathrm{E}[s(y_t,r) - s(y_t,\overline{r})|y_0 = y] - \mathrm{E}[s(y_0,r) - s(y_0,\overline{r})] \right\|_2 \leq C_1 C_2 \|r - \overline{r}\|_2 (1 + \|y\|_2^{q_1 q_2}).$$

Validity of this condition follows.

6. This condition is the same as Assumption 5.7.

The validity of these conditions ensure convergence (statement (1) of the Theorem 5.8). To wrap up the proof, statements (2) and (3) of the theorem follow from Theorem 3.9, while statement (4) is established by Lemma 5.11. **Q.E.D.**

## 5.2 Independent Increments Processes

In the previous section, we assumed that the Markov process of interest is mixing. This assumption ensures a certain sense of "stability" in the underlying system. In this section, we examine a class of "unstable" Markov processes in which the distribution over states becomes increasingly diffuse over time. Specifically, we will study stopping problems involving Markov processes with independent increments.

### 5.2.1 Problem Formulation and Solution

We will assume that the underlying process is Markov and that the transition probability kernel satisfies

$$P(x,A) = P(0,\{y - x|y \in A\}) \qquad \forall x \in \Re^d, A \in \mathcal{B}(\Re^d).$$

Hence, each increment $x_{t+1} - x_t$ is independent of $x_t$.

Let the reward functions $g$ and $G$ be elements of $L_2(\Re^d, \mathcal{B}(\Re^d), l)$ – the Hilbert space with inner product $\langle J, \overline{J} \rangle = \int J(x)\overline{J}(x)dx$ and norm $\|J\| = (\int J^2(x)dx)^{1/2}$. Let $\alpha \in [0,1)$ be a discount factor. Given a stopping time $\tau \in \mathcal{T}$, we define the value function $J^\tau$ by

$$J^\tau(x) = \mathrm{E}\left[\sum_{t=0}^{\tau-1} \alpha^t g(x_t) + \alpha^\tau G(x_\tau) \Big| x_0 = x\right].$$

It turns out that,for any $\tau \in \mathcal{T}$, $J^\tau$ is also an element of $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. We will consider a stopping time $\tau^* \in \mathcal{T}$ optimal if

$$\int_A J^{\tau^*}(x)dx = \sup_{\tau \in \mathcal{T}} \int_A J^\tau(x)dx,$$

for every bounded set $A$.

The keystone of our analysis in Section 5.1.1 was Lemma 4.12, which stated that $P$ is a nonexpansion in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ (i.e., $\|PJ\|_\pi \leq \|J\|_\pi$). Since processes with independent increments do not possess steady–state distributions, a new notion is required. It turns out that the appropriate object is a new lemma, stating that for such processes, $P$ is a nonexpansion in $L_2(\Re^d, \mathcal{B}(\Re^d), l)$. This fact can be established by an argument analogous to that used in proving Lemma 4.12. In particular, by Jensen's inequality, for any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), l)$, we have

$$\begin{aligned} \|PJ\|^2 &= \int (PJ)^2(x)dx \\ &= \int \left(\mathrm{E}[J(x_1)|x_0 = x]\right)^2 dx \\ &\leq \int \mathrm{E}[J^2(x_1)|x_0 = x]dx, \end{aligned}$$

and noting that the increment $\Delta = x_1 - x_0$ is independent of $x_0$, it follows that

$$\begin{aligned} \|PJ\|^2 &\leq \mathrm{E}\left[\int J^2(x+\Delta)dx\right] \\ &= \mathrm{E}\left[\|J\|^2\right] \\ &= \|J\|^2. \end{aligned}$$

Using this fact and arguments analogous to those from Section 5.1.1, it is possible to prove an analog of Theorem 5.2 for processes with independent increments. The results would be the same as those of Theorem 5.2, except that equalities are now in the sense of $L_2(\Re^d, \mathcal{B}(\Re^d), l)$.

### 5.2.2 Approximation Algorithm

We will now discuss an approximation algorithm that is suitable for our new class of optimal stopping problems. Similar to the algorithm of Section 5.1.2, we start by selecting a set of linearly independent basis functions $\phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), l)$. However, we impose an additional requirement that the basis functions have compact support. In particular, there exists a bounded set $A \in \mathcal{B}(\Re^d)$ such that $\phi_k(x) = 0$ for all $k = 1, \ldots, K$ and all $x \notin A$.

Since the Markov process is "unstable," it is no longer viable to generate weight updates based solely on a single simulated trajectory. Instead, given the basis functions and an initial weight vector $r_0$, we generate a sequence according to

$$r_{m+1} = r_m + \gamma_t \phi(x_m)\Big(g(x_m) + \alpha \max\left\{(\Phi r_m)(x_m + \Delta_m), G(x_m + \Delta_m)\right\} - (\Phi r_m)(x_m)\Big),$$

where the $x_m$'s are independent identically distributed random variables drawn from a uniform distribution over the bounded set $A$ that supports the basis functions, and each $\Delta_m$ is drawn independently from all other random variables according to $\mathrm{Prob}\{\Delta_m \in B\} = P(0, B)$ for all $B \in \mathcal{B}(\Re^d)$.

Once more, we define the operator $F$ by $FQ = g + \alpha P \max\{G, Q\}$. This operator can be shown to be a contraction on $L_2(\Re^d, \mathcal{B}(\Re^d), l)$ via arguments analogous to those in the proof of Lemma 5.9. Defining $\Pi$ to be the operator that projects in $L_2(\Re^d, \mathcal{B}(\Re^d), l)$ onto the span of the basis functions, Theorem 3.9 states that the composition $\Pi F$ is a contraction and that its unique fixed point $\Phi r^*$ satisfies an error bound analogous to that of Theorem 5.8(3). A bound on the quality of the resulting stopping time can also be generated using arguments from the proof of Lemma 5.11.

Following the line of reasoning from Section 5.1.2, we rewrite the above update equation as

$$r_{m+1} = r_m + \gamma_m s(y_m, r_m),$$

this time with $s$ defined by

$$s(y, r) = \phi(x)\Big(g(x) + \alpha \max\Big\{(\Phi r)(x + \Delta), G(x + \Delta)\Big\} - (\Phi r)(x)\Big),$$

for any $r$ and $y = (x, \Delta)$. Furthermore, defining $\overline{s}(r) = E[s(y_0, r)]$, we now have

$$\overline{s}_k(r) = \Big\langle \phi_k, F\Phi r - \Phi r \Big\rangle.$$

Hence, by Theorem 3.11, condition (1) of Theorem 3.10, which addresses convergence of stochastic approximation algorithms, is valid. The fact that each $x_m$ is drawn independently alleviates the need for a counterpart to Assumption 5.6 and makes it particularly easy to show that the technical "stability conditions" of Theorem 3.10 hold. This leads to the conclusion that $\{r_m | m = 0, 1, 2, \ldots\}$ converges to $r^*$ (almost surely).

To summarize, under our new assumptions concerning the Markov chain and basis functions together with linear independence of the basis functions (Assumption 5.5 and a technical step size condition (Assumption 5.7), we can establish results analogous to those of Theorem 5.8 for our new algorithm. The only differences are that the norm $\|\cdot\|_\pi$ is replaced by $\|\cdot\|$ and equalities are in the sense of $L_2(\Re^d, \mathcal{B}(\Re^d), l)$.

## 5.3 Finite Horizon Problems

In certain practical situations, one may be interested in optimizing rewards over only a finite time horizon. Such problems are generally simpler to analyze than their infinite horizon counterparts, but at the same time, involve an additional complication because expected rewards will generally depend on the remaining time. In this section, we develop an approximation algorithm that is suitable for such problems.

### 5.3.1 Problem Formulation and Solution

We assume that the underlying process is initialized with $x_0 = \overline{x}$ and that its evolution is Markov. For each nonnegative integer $t$, we define a distribution

$$\pi_t(A) = \text{Prob}\{x_t \in A\}.$$

We assume that the discount factor $\alpha$ is in $[0, 1)$ and that the reward functions $g$ and $G$ satisfy

$$\int g^2(x)\pi_t(dx) < \infty \quad \text{and} \quad \int G^2(x)\pi_t(dx) < \infty,$$

for all $t$.

The horizon of the problem is defined by a nonnegative integer $h$, which we take to be fixed. An optimal stopping time $\tau^* \in \mathcal{T}$ is one that satisfies

$$\mathrm{E}[J^{\tau^* \wedge h}(x_0)] = \sup_{\tau \in \mathcal{T}} \mathrm{E}[J^{\tau \wedge h}(x_0)].$$

Standard results from the finite–horizon dynamic programming literature apply. In particular,

$$\sup_{\tau \in \mathcal{T}} J^{\tau \wedge h}(x) = (T^h G)(x) = J^{\tau_h}(x), \qquad \forall x \in \Re^d,$$

where

$$TJ = \max\{G, \alpha P J\},,$$

and $\tau_h \in \mathcal{T}$ is defined by

$$\tau_h = \min\left\{t \leq h | G(x_t) \geq (T^{h-t}G)(x_t)\right\}.$$

(We let $\tau_h = \infty$ if the set is empty.) Hence, $\tau^* = \tau_h$ is an optimal stopping time.

### 5.3.2 Approximation Algorithm

As in Section 5.1.2, let the operator $F$ be defined by $FQ = g + \alpha P \max\{G, Q\}$. It is easy to verify that the optimal stopping time $\tau^*$ can alternatively be generated according to

$$\tau^* = \min\left\{t \leq h | G(x_t) \geq (F^{h-t}G)(x_t)\right\}.$$

Note that this construction relies on knowledge of $FG, F^2G, \ldots, F^hG$. A suitable approximation algorithm should be designed to accommodate such needs.

Let the measure $\mu$ over the product space $(\mathcal{B}(\Re^d))^h$ be defined by

$$\mu(A) = \pi_0(A_0) + \cdots + \pi_{h-1}(A_{h-1}), \qquad \forall A = A_0 \times A_1 \times \cdots \times A_{h-1},$$

and let $L_2^h(\Re^d, \mathcal{B}(\Re^d), \mu)$ be the Hilbert space defined with respect to this measure. Note that the domain of functions in this Hilbert space is $\Re^d \times \{1, \ldots, h\}$, and the norm on the Hilbert space is given by

$$\|(Q_1, \ldots, Q_{h-1})\|_\mu^{1/2} = \int Q_0^2(x)\pi_0(dx) + \cdots + \int Q_{h-1}^2(x)\pi_{h-1}(dx).$$

Our approximation algorithm here employs a set of basis functions $\phi_1, \ldots, \phi_K \in L_2^h(\Re^d, \mathcal{B}(\Re^d), \mu)$. The algorithm we propose for finite horizon problems simulates a sequence of finite length trajectories rather than a single infinite trajectory. In particular, for each $i = 1, 2, 3, \ldots$, let $x_0^i, \ldots, x_h^i$ be a simulated trajectory (with

$x_0^i = \overline{x}$). The weight vector is initialized to some arbitrary $r_0 \in \Re^K$, and is updated according to

$$r_{i+1} = r_i + \gamma_i \sum_{t=0}^{h-1} \phi(x_t^i, t) d_t^i,$$

where

$$d_t^i = \begin{cases} g(x_t^i) + \alpha \max\left\{(\Phi r_i)(x_{t+1}^i, t+1), G(x_{t+1}^i)\right\} - (\Phi r_i)(x_t^i, t), & \text{if } t+1 < h, \\ \alpha G(x_{t+1}^i) - (\Phi r_i)(x_t^i, t), & \text{otherwise.} \end{cases}$$

As usual, we assume that the step sizes satisfy $\sum_{i=0}^{\infty} \gamma_i = \infty$ and $\sum_{i=0}^{\infty} \gamma_i^2 < \infty$.

We now discuss how ideas analogous to those of Section 5.1.2 can be used to establish convergence and an error bound.

Define an operator $H$ by

$$H(Q_0, \ldots, Q_{h-1}) = (FQ_1, FQ_2, \ldots, FQ_{h-1}, FG), \qquad \forall (Q_0, Q_1, \ldots, Q_{h-1}) \in L_2(\mu).$$

Using an argument similar to that of Lemma 5.9, it is easy to show that $H$ is a contraction on $L_2^h(\Re^d, \mathcal{B}(\Re^d), \mu)$. Furthermore, the unique fixed point is given by

$$(Q_0^*, Q_1^*, \ldots, Q_{h-1}^*) = (F^h G, F^{h-1} G, \ldots, F^2 G, FG),$$

which can be used to generate an optimal stopping time.

We assume that the basis functions are linearly independent. Let $\Pi$ be the operator that projects in $L_2^h(\Re^d, \mathcal{B}(\Re^d), \mu)$ onto the subspace spanned by the basis functions. By Theorem 3.9, the composition $\Pi H$ is a contraction on $L_2^h(\Re^d, \mathcal{B}(\Re^d), \mu)$. Hence, it has a unique fixed point of the form $\Phi r^*$ for some $r^* \in \Re^K$. Furthermore, $r^*$ satisfies

$$\|\Phi r^* - (Q_0^*, \ldots, Q_{h-1}^*)\|_\mu \leq \frac{1}{\sqrt{1-\kappa^2}} \|\Pi(Q_0^*, \ldots, Q_{h-1}^*) - (Q_0^*, \ldots, Q_{h-1}^*)\|_\mu,$$

where $\kappa \leq \alpha$ is the contraction factor of $\Pi H$. A bound on the performance of a stopping time $\tilde{\tau} = \min\left\{t \leq h | G(x_t) \geq (\Phi r^*)(x_t, t)\right\}$ can also be established:

$$J_{0,h}^*(\overline{x}) - J_{0,h}^{\tilde{\tau}}(\overline{x}) \leq \frac{2}{(1-\alpha)\sqrt{1-\kappa^2}} \|\Pi(Q_0^*, \ldots, Q_{h-1}^*) - (Q_0^*, \ldots, Q_{h-1}^*)\|_\mu.$$

(Both bounds can be strengthened, if we allow coefficients on the right–hand–sides to depend on $h$, but we will not pursue this issue further here.)

Once again, Theorem 3.10 can be used to prove convergence. As usual, we rewrite the update equation in the form

$$r_{i+1} = r_i + \gamma_i s(y_i, r_i),$$

except that we now define $y_i = (x_0^i, \ldots, x_h^i)$. Some algebra gives us

$$\overline{s}_k(r) = \mathrm{E}[s_k(y_0, r)] = \langle \phi_k, H\Phi r - \Phi r \rangle_\mu.$$

By Theorem 3.11, this satisfies condition (1) of Theorem 3.10. Since the $y_i$'s are independent and identically distributed, it is easy to show that the various technical conditions of Theorem 3.10 are also satisfied, which would imply that $\{r_i | i = 0, 1, 2, \ldots\}$ converges to $r^*$.

## 5.4 A Two–Player Zero–Sum Game

Many interesting phenomena arise when multiple participants make decisions within a single system. Variants of temporal–difference learning amenable to two–player zero–sum games were used to produce the Backgammon player of Tesauro [79, 80, 81] and have been studied by Littman [46] (for the case involving an exhaustive representation of the value function). In this section, we consider a simple two–player zero–sum game in which a reward–maximizing player ("player 1") is allowed to stop a process at any even time step and a reward–minimizing player ("player 2") can opt to terminate during odd time steps.

### 5.4.1 Problem Formulation and Solution

We consider a stationary mixing Markov process with a steady–state distribution $\pi$ together with reward functions $g, G_1, G_2 \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ and a discount factor $\alpha \in [0, 1)$. Prior to termination, a reward of $g(x_t)$ is obtained during each time step, and upon termination, a reward of either $G_1(x_t)$ or $G_2(x_t)$ is generated depending on which player opted to terminate. We define sets $\mathcal{T}_1 = \{\tau \in \mathcal{T} | \tau \text{ even}\}$ and $\mathcal{T}_2 = \{\tau \in \mathcal{T} | \tau \text{ odd}\}$ corresponding to admissible strategies for players 1 and 2, respectively. For each pair of stopping times $\tau_1 \in \mathcal{T}_1$ and $\tau_2 \in \mathcal{T}_2$, we define a value function

$$J^{\tau_1,\tau_2}(x) = \mathrm{E}\left[\sum_{t=0}^{\tau_1 \wedge \tau_2 - 1} \alpha^t g(x_t) + \psi_1 \alpha^{\tau_1} G_1(x_{\tau_1}) + \psi_2 \alpha^{\tau_2} G_2(x_{\tau_2}) \Big| x_0 = x\right],$$

where $\psi_1$ and $\psi_2$ are indicators of the events $\{\tau_1 < \tau_2, \tau_1 < \infty\}$ and $\{\tau_2 < \tau_1, \tau_2 < \infty\}$, respectively. Hence, if players 1 and 2 take $\tau_1 \in \mathcal{T}_1$ and $\tau_2 \in \mathcal{T}_2$ as their strategies, the expected reward for the game is $\mathrm{E}[J^{\tau_1,\tau_2}(x_0)]$. We consider sup-inf and inf-sup expected rewards

$$\sup_{\tau_1 \in \mathcal{T}_1} \inf_{\tau_2 \in \mathcal{T}_2} \mathrm{E}\left[J^{\tau_1,\tau_2}(x_0)\right] \quad \text{and} \quad \inf_{\tau_2 \in \mathcal{T}_2} \sup_{\tau_1 \in \mathcal{T}_1} \mathrm{E}\left[J^{\tau_1,\tau_2}(x_0)\right].$$

which correspond to different orders in which the players select their strategies. When both of these expressions take on the same value, this is considered to be the *equilibrium value* of the game. A pair of stopping times $\tau_1^* \in \mathcal{T}_1$ and $\tau_2^* \in \mathcal{T}_2$ are optimal if

$$\mathrm{E}\left[J^{\tau_1^*,\tau_2^*}(x_0)\right] = \sup_{\tau_1 \in \mathcal{T}_1} \inf_{\tau_2 \in \mathcal{T}_2} \mathrm{E}\left[J^{\tau_1,\tau_2}(x_0)\right] = \inf_{\tau_2 \in \mathcal{T}_2} \sup_{\tau_1 \in \mathcal{T}_1} \mathrm{E}\left[J^{\tau_1,\tau_2}(x_0)\right].$$

The problem of interest is that of finding such stopping times.

We define operators $T_1 J = \max\{G_1, g + \alpha P J\}$ and $T_2 J = \min\{G_2, g + \alpha P J\}$. By the same argument as that used to prove Lemma 5.3, both these operators are contractions on $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. It follows that the compositions $T_1 T_2$ and $T_2 T_1$ are also contractions. We will denote the fixed points of $T_1 T_2$ and $T_2 T_1$ by $J_1^*$ and $J_2^*$, respectively.

We define stopping times $\tau_1^* = \min\{\text{even } t | G(x_t) \geq J_1^*(x_t)\}$ and $\tau_2^* = \min\{\text{odd } t | G(x_t) \leq J_2^*(x_t)\}$. Using the fact that $T_1 T_2$ and $T_2 T_1$ are contractions, the arguments of Section 5.1.1 can be generalized to prove that

$$J_1^* = J^{\tau_1^*, \tau_2^*},$$

and

$$\sup_{\tau_1 \in \mathcal{T}_1} \inf_{\tau_2 \in \mathcal{T}_2} \mathrm{E}[J^{\tau_1,\tau_2}(x_0)] = \inf_{\tau_2 \in \mathcal{T}_2} \sup_{\tau_1 \in \mathcal{T}_1} \mathrm{E}[J^{\tau_1,\tau_2}(x_0)] = \mathrm{E}[J^{\tau_1^*,\tau_2^*}(x_0)].$$

In other words, the pair of stopping times $\tau_1^*$ and $\tau_2^*$ is optimal.

### 5.4.2 Approximation Algorithm

We now present an algorithm for approximating a pair of optimal stopping times and the equilibrium value of the game. Given a set of linearly independent basis functions $\phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, we begin with initial weight vectors $r_{1,0}, r_{2,0} \in \Re^K$ and generate two sequences according to

$$r_{1,t+1} = r_{1,t} + \gamma_t \phi(x_t) \Big( g(x_t) + \alpha \min\{(\Phi r_{2,t})(x_{t+1}), G_2(x_{t+1})\} - (\Phi r_{1,t})(x_t) \Big),$$

and

$$r_{2,t+1} = r_{2,t} + \gamma_t \phi(x_t) \Big( g(x_t) + \alpha \max\{(\Phi r_{1,t})(x_{t+1}), G_1(x_{t+1})\} - (\Phi r_{2,t})(x_t) \Big),$$

where the step sizes satisfy Assumption 5.7.

To generalize the analysis of Section 5.1.2, we define operators $F_1 Q = g + \alpha P \min\{G_2, Q\}$ and $F_2 Q = g + \alpha P \max\{G_1, Q\}$. It is easy to show that these operators are contractions on $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, and so are their compositions $F_1 F_2$ and $F_2 F_1$. It is also easy to verify that the fixed points of $F_1 F_2$ and $F_2 F_1$ are given by $Q_1^* = g + \alpha P J_2^*$ and $Q_2^* = g + \alpha P J_1^*$, respectively. Furthermore, $Q_1^* = F_1 Q_2^*$, $Q_2^* = F_2 Q_1^*$, and the stopping times $\tau_1^*$ and $\tau_2^*$ can alternatively be generated according to $\tau_1^* = \min\{\text{even } t | G(x_t) \geq Q_1^*(x_t)\}$ and $\tau_2^* = \min\{\text{odd } t | G(x_t) \leq Q_2^*(x_t)\}$.

Let us define a measure $\mu$ over the product space $(\mathcal{B}(\Re^d))^2$ by $\mu(A_1, A_2) = \pi(A_1) + \pi(A_2)$ and an operator $H : L_2^2(\Re^d, \mathcal{B}(\Re^d), \mu) \mapsto L_2^2(\Re^d, \mathcal{B}(\Re^d), \mu)$, given by

$$H(Q_1, Q_2) = (F_1 Q_2, F_2 Q_1).$$

It is easy to show that $H$ is a contraction on $L_2^2(\Re^d, \mathcal{B}(\Re^d), \mu)$ with fixed point $(Q_1^*, Q_2^*)$.

Let $\Pi$ be the operator that projects in $L_2^2(\Re^d, \mathcal{B}(\Re^d), \mu)$ onto the subspace $\{(\Phi r, \Phi \overline{r}) | r, \overline{r} \in \Re^K\}$. By Theorem 3.9, the composition $\Pi H$ is a contraction in $L_2(\Re^d, \mathcal{B}(\Re^d), \mu)$ with

a unique fixed point $(\Phi r_1^*, \Phi r_2^*)$. Furthermore, this fixed point satisfies

$$\|(\Phi r_1^*, \Phi r_2^*) - (Q_1^*, Q_2^*)\|_\mu \leq \frac{1}{\sqrt{1-\kappa^2}} \|\Pi(Q_1^*, Q_2^*) - (Q_1^*, Q_2^*)\|_\mu,$$

where $\kappa \leq \alpha$ is the contraction factor of $\Pi H$. Finally, the value of the game under stopping times $\tilde{\tau}_1 = \min\{\text{even } t | G(x_t) \geq (\Phi r_1^*)(x_t)\}$ and $\tilde{\tau}_2 = \min\{\text{odd } t | G(x_t) \leq (\Phi r_2^*)(x_t)\}$ deviates by a bounded amount from the equilibrium value. In particular,

$$|\mathrm{E}[J_1^*(x_0)] - \mathrm{E}[J^{\tilde{\tau}_1, \tilde{\tau}_2}(x_0)]| \leq \frac{2}{(1-\alpha)\sqrt{1-\kappa^2}} \|\Pi(Q_1^*, Q_2^*) - (Q_1^*, Q_2^*)\|_\mu.$$

To establish convergence, we rewrite the update equation in the form

$$(r_{1,t+1}, r_{2,t+1}) = (r_{1,t}, r_{2,t}) + \gamma_t s(y_t, (r_{1,t}, r_{2,t})),$$

where $y_t = (x_t, x_{t+1})$, and note that $\bar{s}(r_1, r_2) = \mathrm{E}[s(y_0, (r_1, r_2))]$ is given by

$$\bar{s}_k(r_1, r_2) = \begin{cases} \left\langle \phi_k, F_1 \Phi r_2 - \Phi r_1 \right\rangle_\pi, & \text{if } k \leq K, \\ \left\langle \phi_{k-K}, F_2 \Phi r_1 - \Phi r_2 \right\rangle_\pi, & \text{otherwise,} \end{cases}$$

for any $r_1, r_2 \in \Re^K$. Validity of condition (1) of Theorem 3.10 is then valid by Theorem 3.11. Combining this fact with an analog of Assumption 5.6, the technical requirements of Theorem 3.10 can be verified. This would imply that $r_{1,t}$ and $r_{2,t}$ almost surely converge to $r_1^*$ and $r_2^*$, respectively.

## 5.5 Closing Remarks

The analysis we have presented is the first that proves convergence of a variant of temporal–difference learning in a context involving a sequential decision problem (not just a fixed policy) and general linear parameterizations. An open question is whether this line of reasoning can somehow be extended to broader classes of problems of practical interest.

Aside from its relation to temporal–difference learning, our work constitutes a new development in the theory of optimal stopping. In particular, though the characterization of optimal value functions and optimal stopping times are of a standard flavor, the assumptions and line of reasoning that arrive at these results are not. Most important, however, are the results pertaining to approximations. Our approximation algorithms may offer a sound approach to solving optimal stopping problems with high–dimensional state spaces that would otherwise be dismissed as intractable to systematic methodologies. In the next chapter, we explore the utility of one algorithm through a case study relevant to the derivatives industry.

# Chapter 6

# Pricing High–Dimensional Derivatives

In this chapter, we present a case study involving the application of an algorithm from the previous chapter to approximate the solution to a high–dimensional optimal stopping problem. The problem is representative of high–dimensional derivatives pricing problems arising in the rapidly growing structured products (a.k.a. "exotics") industry [56]. Our approach involving the approximation of a value function is similar in spirit to the earlier experimental work of Barraquand and Martineau [4]. However, the algorithm employed in that study is different from ours, and the approximations were comprised of piecewise constant functions.

Another notable approach to approximating solutions of high–dimensional optimal stopping problems that arise in derivatives pricing is the "stochastic mesh" methods of Broadie and Glasserman [19, 18]. These methods can be thought of as variants of Rust's algorithm [63], specialized to the context of optimal stopping. Values are approximated at points in a finite mesh over the state space in a spirit similar to grid techniques. The difference is that the mesh includes a tractable collection of randomly sampled states, rather than the intractable grid that would arise in standard state space discretization. When the state space is high–dimensional, except for cases that satisfy unrealistically restrictive assumptions as those presented in [63], the randomly sampled states may not generally be sufficiently representative for effective value function approximation.

We will begin by providing some background on financial derivative securities. Section 6.2 then introduces the particular security we consider and a related optimal stopping problem. Section 6.3 presents the performance of some simple stopping strategies. Finally, the selection of basis functions and computational results generated by our approximation algorithm are discussed in Section 6.4.

## 6.1 Background

Financial derivative securities (or derivatives, for short) are contracts that promise payoffs contingent on the future prices of basic assets such as stocks, bonds, and

commodities. Certain types of derivatives, such as put and call options, are in popular demand and traded alongside stocks in large exchanges. Other more exotic derivatives are tailored by banks and other financial intermediaries in order to suit specialized needs of various institutions and are sold in "over-the-counter" markets.

Exotic derivatives tend to be illiquid relative to securities that are traded in mainstream markets. Consequently, it may be difficult for an institution to "cash in" on the worth of the contract when the need arises unless such a situation is explicitly accommodated by the terms of the contract. Because institutions desire flexibility, derivatives typically allow the possibility of "early exercise." In particular, an institution may "exercise" the security at various points during the lifetime of the contract, thereby settling with the issuer according to certain prespecified terms.

Several important considerations come into play when a bank designs a derivative security. First, the product should well suit the needs of clients, incurring low costs for large gains in customer satisfaction. Second, it is necessary to devise a hedging strategy, which is a plan whereby the bank can be sure to fulfill the terms of the contract without assuming significant risks. Finally, the costs of implementing the hedging strategy must be computed in order to determine an appropriate price to charge clients.

Under certain technical assumptions, it is possible to devise a hedging strategy that perfectly replicates the payoffs of a derivative security. Hence, the initial investment required to operate this hedging strategy must be equal to the value of the security. This approach to replication and valuation, introduced by Black and Scholes [14] and Merton [50] and presented in its definitive form by Harrison and Kreps [34] and Harrison and Pliska [35], has met wide application and is the subject of much subsequent research.

When there is a possibility of early exercise, the value of the derivative security depends on how the client chooses a time to exercise. Given that the bank can not control the client's behavior, it must prepare for the worst by assuming that the client will employ an exercising strategy that maximizes the value of the security. Pricing the derivative security in this context generally requires solving an optimal stopping problem.

In the next few sections, we present one fictitious derivative security that leads to a high–dimensional optimal stopping problem, and we employ the algorithm we have developed in order to approximate its price. Our focus here is to demonstrate the use of the algorithm, rather than to solve a real–world problem. Hence, we employ very simple models and ignore details that may be required in order to make the problem realistic.

## 6.2 Problem Formulation

The financial derivative instrument we will consider generates payoffs that are contingent on prices of a single stock. At the end of any given day, the holder may opt to exercise. At the time of exercise, the contract is terminated, and a payoff is received in an amount equal to the current price of the stock divided by the price prevailing

one hundred days beforehand.

One interesting interpretation of this derivative security is as an oracle that offers a degree of foresight. The payoff is equal to the amount that would accrue from a one Dollar investment in the stock made one hundred days prior to exercise. However, the holder of the security can base her choice of the time at which this Dollar is invested on knowledge of the returns over the one hundred days. The price of this security should in some sense represent the value of this foresight.

We will employ a standard continuous–time economic model involving a stochastic stock price process and deterministic returns generated by short–term bonds. Given this model, under certain technical conditions, it is possible to replicate derivative securities that are contingent on the stock price process by rebalancing a portfolio of stocks and bonds. This portfolio needs only an initial investment, and is self–financing thereafter. Hence, to preclude arbitrage, the price of the derivative security must be equal to the initial investment required by such a portfolio. Karatzas [42] provides a comprehensive treatment of this pricing methodology in the case where early exercising is allowed. In particular, the value of the security is equal to the optimal reward for a particular optimal stopping problem. The framework of [42] does not explicitly capture our problem at hand (the framework allows early exercise at any positive time, while our security can only be exercised at the end of each day), but the extension is immediate. Since our motivation is to demonstrate the use of our algorithm, rather than dwelling on the steps required to formally reduce pricing to an optimal stopping problem, we will simply present the underlying economic model and the optimal stopping problem it leads to, omitting the technicalities needed to formally connect the two.

We model time as a continuous variable $t \in [-100, \infty)$ and assume that the derivative security is issued at time $t = 0$. Each unit of time is taken to be a day, and the security can be exercised at times $t \in \{0, 1, 2, \ldots\}$. We model the stock price process $\{p_t | t = -100, -99, -98, \ldots\}$ as a geometric Brownian motion

$$p_t = p_{-100} + \int_{s=-100}^{t} \mu p_s ds + \int_{s=-100}^{t} \sigma p_s dw_s,$$

for some positive scalars $p_0$, $\mu$, and $\sigma$, and a standard Brownian motion $w_t$. The payoff received by the security holder is equal to $p_\tau / p_{\tau-100}$ where $\tau \geq 0$ is the time of exercise. Note that we consider negative times because the stock prices up to a hundred days prior to the date of issue may influence the payoff of the security. We assume that there is a constant continuously compounded short–term interest rate $\rho$. In other words, $D_0$ Dollars invested in the money market at time 0 grows to a value

$$D_t = D_0 e^{\rho t},$$

at time $t$.

We will now characterize the price of the derivative security in a way that gives rise to a related optimal stopping problem. Let $\{\tilde{p}_t | t = -100, -99, -98, \ldots\}$ be a

stochastic process that evolves according to

$$d\tilde{p}_t = \rho\tilde{p}_t dt + \sigma\tilde{p}_t dw_t.$$

Define a discrete–time process $\{x_t | t = 0, 1, 2, \ldots\}$ taking values in $\Re^{100}$, with

$$x_t = \left(\frac{\tilde{p}_{t-99}}{\tilde{p}_{t-100}}, \frac{\tilde{p}_{t-98}}{\tilde{p}_{t-100}}, \ldots, \frac{\tilde{p}_t}{\tilde{p}_{t-100}}\right)'.$$

Intuitively, the $i$th component $x_t(i)$ of $x_t$ represents the amount a one Dollar investment made in the stock at time $t - 100$ would grow to at time $t - 100 + i$ if the stock price followed $\{\tilde{p}_t\}$. It is easy to see that this process $\{x_t | t = 0, 1, 2, \ldots\}$ is Markov. Furthermore, it is stationary and mixing since, for any $t \in \{0, 1, 2, \ldots\}$, the random variables $x_t$ and $x_{t+100}$ are independent and identically distributed. Consequently, the algorithm of Section 5.1.2 is applicable. Letting $\alpha = e^{-\rho}$, $G(x) = x(100)$, and

$$x = \left(\frac{p_{-99}}{p_{-100}}, \frac{p_{-98}}{p_{-100}}, \ldots, \frac{p_t}{p_{-100}}\right)',$$

the value of the derivative security is given by

$$\sup_{\tau \in \mathcal{T}} E[\alpha^\tau G(x_\tau) | x_0 = x].$$

If $\tau^*$ is an optimal stopping time, we have

$$E[\alpha^{\tau^*} G(x_{\tau^*}) | x_0 = x] = \sup_{\tau \in \mathcal{T}} E[\alpha^\tau G(x_\tau) | x_0 = x],$$

for almost every $x_0$. Hence, given an optimal stopping time, we can price the security by evaluating an expectation, possibly through use of Monte–Carlo simulation. However, because the state space is so large, it is unlikely that we will be able to compute an optimal stopping time. Instead, we must resort to generating a suboptimal stopping time $\tilde{\tau}$ and computing

$$E[\alpha^{\tilde{\tau}} G(x_{\tilde{\tau}}) | x_0 = x],$$

as an approximation to the security price. Note that this approximation is a lower bound for the true price. The approximation generally improves with the performance of the optimal stopping strategy. In the next two sections, we present computational results involving the selection of stopping times for this problem and the assessment of their performance. In the particular example we will consider, we use the settings $\sigma = 0.02$ and $\rho = 0.0004$ (the value of the drift $\mu$ is inconsequential). Intuitively, these choices correspond to a stock with a daily volatility of 2% and an annual interest rate of about 10%. (Interest accrues and stock prices fluctuate only on days during which the market is open.)

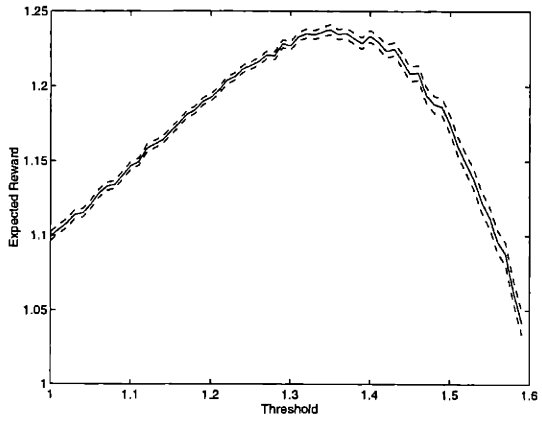


Figure 6-1: Expected reward as a function of threshold. The values plotted are estimates generated by averaging rewards obtained over ten thousand simulated trajectories, each initialized according to the steady–state distribution and terminated according to the stopping time dictated by the thresholding strategy. The dashed lines represent confidence bounds generated by estimating the standard deviation of each sample mean, and adding/subtracting twice this estimate to/from the sample mean.

## 6.3 A Thresholding Strategy

In order to provide a baseline against which we can compare the performance of our approximation algorithm, let us first discuss the performance of a simple heuristic stopping strategy. In particular, consider the stopping time $\tau_B = \min\{t | G(x_t) \geq B\}$ for a scalar threshold $B \in \Re$. We define the performance of such a stopping time in terms of the expected reward $E[J^{\tau_B}(x_0)]$. In the context of our pricing problem, this quantity represents the average price of the derivative security (averaged over possible initial states). Expected rewards generated by various threshold values are presented in Figure 6-1. The optimal expected reward over the thresholds tried was 1.238.

It is clear that a thresholding strategy is not optimal. For instance, if we know that there was a large slump and recovery in the process $\{\tilde{p}_t\}$ within the past hundred days, we should probably wait until we are about a hundred days past the low point in order to reap potential benefits. However, the thresholding strategy, which relies exclusively on the ratio between $\tilde{p}_t$ and $\tilde{p}_{t-100}$, cannot exploit such information.

What is not clear is the *degree* to which the thresholding strategy can be improved. In particular, it may seem that events in which such a strategy makes significantly inadequate decisions are rare, and it therefore might be sufficient, for practical purposes, to limit attention to thresholding strategies. In the next section, we rebut this hypothesis by generating a substantially superior stopping time using our approximation methodology.

## 6.4 Using the Approximation Algorithm

Perhaps the most important step prior to applying the approximation algorithm from Section 5.1.2 is selecting an appropriate set of basis functions. Though analysis can sometimes help, this task is largely an art form, and the process of basis function selection typically entails repetitive trial and error.

We were fortunate in that our first choice of basis functions for the problem at hand delivered promising results relative to thresholding strategies. To generate some perspective, along with describing the basis functions, we will provide brief discussions concerning our (heuristic) rationale for selecting them. The first two basis functions were simply a constant function $\phi_1(x) = 1$ and the reward function $\phi_2(x) = G(x)$. Next, thinking that it might be important to know the maximal and minimal returns over the past hundred days, and how long ago they occurred, we constructed the following four basis functions

$$\begin{aligned}
\phi_3(x) &= \min_{i=1,\ldots,100} x(i) - 1, \\
\phi_4(x) &= \max_{i=1,\ldots,100} x(i) - 1, \\
\phi_5(x) &= \frac{1}{50} \operatorname*{argmin}_{i=1,\ldots,100} x(i) - 1, \\
\phi_6(x) &= \frac{1}{50} \operatorname*{argmax}_{i=1,\ldots,100} x(i) - 1.
\end{aligned}$$

Note that that the basis functions involve constant scaling factors and/or offsets. The purpose of these transformation is to maintain the ranges of basis function values within the same regime. Though this is not required for convergence of our algorithm, it can speed up the process significantly.

As mentioned previously, if we invested one dollar in the stock at time $t - 100$ and the stock price followed the process $\{\tilde{p}_t\}$, then the sequence $x_t(1), \ldots, x_t(100)$ represents the daily values of the investment over the following hundred day period. Conjecturing that the general shape of this hundred–day sample path is of importance, we generated four basis functions aimed at summarizing its characteristics. These basis functions represent inner products of the sample path with Legendre polynomials of degrees one through four. In particular, letting $j = i/50 - 1$, we defined

$$\begin{aligned}
\phi_7(x) &= \frac{1}{100} \sum_{i=1}^{100} \frac{x(i) - 1}{\sqrt{2}}, \\
\phi_8(x) &= \frac{1}{100} \sum_{i=1}^{100} x(i) \sqrt{\frac{3}{2}} j, \\
\phi_9(x) &= \frac{1}{100} \sum_{i=1}^{100} x(i) \sqrt{\frac{5}{2}} \left( \frac{3j^2}{2} - \frac{1}{2} \right), \\
\phi_{10}(x) &= \frac{1}{100} \sum_{i=1}^{100} x(i) \sqrt{\frac{7}{2}} \left( \frac{5j^3}{2} - \frac{3j}{2} \right).
\end{aligned}$$

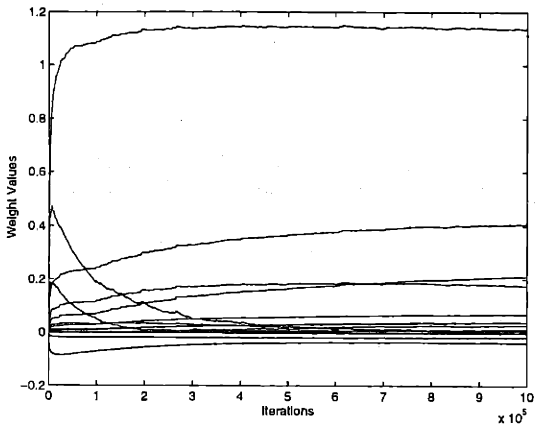


Figure 6-2: The evolution of weights during execution of the algorithm. The value of the security under the resulting strategy was 1.282.

So far, we have constructed basis functions in accordance to "features" of the state that might be pertinent to effective decision–making. Since our approximation of the value function will be composed of a weighted sum of the basis functions, the nature of the relationship between these features and approximated values is restricted to linear. To capture more complex trade-offs between features, it is useful to consider nonlinear combinations of certain basis functions. For our problem, we constructed six additional basis functions using products of the original features. These basis functions are given by

$$\begin{aligned}
\phi_{11}(x) &= \phi_2(x)\phi_3(x), \\
\phi_{12}(x) &= \phi_2(x)\phi_4(x), \\
\phi_{13}(x) &= \phi_2(x)\phi_7(x), \\
\phi_{14}(x) &= \phi_2(x)\phi_8(x), \\
\phi_{15}(x) &= \phi_2(x)\phi_9(x), \\
\phi_{16}(x) &= \phi_2(x)\phi_{10}(x).
\end{aligned}$$

Using our sixteen basis functions, we generated a sequence of parameters $r_0, r_1, \ldots, r_{10^6}$ by initializing each component of $r_0$ to 0 and iterating the update equation one million times with a step size of $\gamma_t = 0.001$. The evolution of the iterates is illustrated in Figure 6-2.

The weight vector $r_{10^6}$ resulting from our numerical procedure was used to generate a stopping time $\tilde{\tau} = \min\{t | G(x_t) \geq (\Phi r_{10^6})(x_t)\}$. The corresponding expected reward $E[J^{\tilde{\tau}}(x_0)]$, estimated by averaging the results of ten thousand trajectories each initialized according to the steady–state distribution and terminated according to the stopping time $\tilde{\tau}$, was 1.282 (the estimated standard deviation for this sample mean was 0.0022). This value is significantly greater than the expected reward generated

by the optimized threshold strategy of the previous section. In particular, we have

$$E[J^{\tilde{\tau}}(x_0) - J^{\tau_B}(x_0)] \approx 0.044.$$

As a parting note, we mention that each stopping time $\tau$ corresponds to an exercising strategy that the holder of the security may follow, and $J^{\tau}(x_0)$ represents the value of the security under this exercising strategy. Hence, the difference between $E[J^{\tilde{\tau}}(x_0)]$ and $E[J^{\tau_B}(x_0)]$ implies that, on average (with respect to the steady–state distribution of $x_t$), the fair price of the security is about four percent higher when exercised according to $\tilde{\tau}$ instead of $\tau_B$. In the event that a bank assumes that $\tau_B$ is optimal and charges a price of $J^{\tau_B}(x_0)$, an arbitrage opportunity may become available.

## 6.5 Closing Remarks

In addition to pricing a derivative security contract, issuers must hedge their positions. In the Black–Scholes–Merton model, this hedging can be achieved via managing a portfolio of underlying assets and bonds. The idea is to maintain a portfolio such that changes in the value of the contract are offset by changes in value of the portfolio. The portfolio weights vary continuously with time, and they can be generated by taking gradients of the value function with respect to underlying asset prices and time. When the exact value function is unavailable, as is the case in the example of this chapter, we could consider using an approximate value function in its place. In particular, hedging decisions can be made based on gradients of $\Phi r^*$. Let us also mention an alternative, which may sometimes deliver superior results. This approach employs the gradient of the value function $J^{\tilde{\tau}}$ corresponding to the policy $\tilde{\tau}$ and uses this quantity to make hedging decisions. Note that, since this policy is available to us, the associated value function can be estimated via Monte Carlo simulation, in the same spirit as "rollout methods" (see, e.g., [13]).

The problem of this chapter was chosen as a simple illustration. In reality, almost all derivative securities have a prespecified expiration date, upon which the contract terminates if it has not been exercised. Such contracts lead to finite horizon problems, rather than one of the infinite horizon variety considered in this chapter. Furthermore, the payoff function employed in our case study does not correspond to that of any particularly popular contract. The following list provides a few more popular examples that lead to similar high–dimensional optimal stopping problems.

- **Min/Max options:** Payoff is a function of the minimum (or maximum) among prices of a set of securities (see, e.g., [40]).
- **Asian options:** Payoff is contingent on an arithmetic moving average of a security price.
- **Lookback options:** Payoff is contingent on the current security price as well as certain past prices.

- **Fixed income derivatives:** Payoff is contingent on the term structure of interest rates at the time of exercise. It is common to view the term structure as an infinite–dimensional process driven by a finite number of factors (see, e.g., [37]). The pricing of fixed income derivatives, in this context, often entails solving optimal stopping problems with state spaces of dimension equal to the number of factors. An interesting issue here concerns the trade–offs between approximate solution via methods of the type we have proposed for models involving many factors and exact solution by dynamic programming for approximate models involving fewer factors.

As a parting note, let us speculate that the ideas of Hilbert space approximation may be useful in designing efficient numerical techniques for pricing "vanilla options," which involve low–dimensional stopping problems. Though these problems are computationally tractable, because of the enormous number of such problems that must be solved daily on a trading floor, there is significant interest in designing methods that are as fast as possible. Current state–of–the–art approaches exploit the structure of "stopping regions" for very specialized classes of problems (see, e.g., [41]). An alternative might be to select a set of basis functions that can closely approximate value functions corresponding to an important class of problems, and when presented with a problem instance, to quickly generate weights via solving a fixed point equation $J^* = \Pi T J^*$. Though stochastic methods would probably be inappropriate for this context, one might design deterministic algorithms that make use of ideas from Chapter 3.

# Chapter 7

# Temporal–Difference Learning with Averaged Rewards

Until now, we have considered expected discounted rewards and the generation of policies that optimize such a criterion. In that context, the value function provided a natural guide for decision making. An alternative criterion that is more appropriate in certain practical situations involves averaged rewards. In this case, the object of interest is the *differential* value function. Given that the average reward of a process is $\mu^*$, the differential value function is defined by

$$J^*(x) = E\left[\sum_{t=0}^{\infty}(g(x_t) - \mu^*)\Big|x_0 = x\right].$$

Such functions are used in classical dynamic programming algorithms that optimize averaged rewards (see, e.g., [8]).

In this chapter, we propose and analyze a variant of temporal–difference learning that approximates differential value functions in autonomous processes. For this algorithm, which we will refer to as *average reward temporal–difference learning*, we establish results that parallel those of Chapter 4. We begin by defining the algorithm and presenting the associated convergence results and analysis. Relationships between the algorithm of this chapter and that studied in Chapter 4 (for discounted rewards) are discussed in Section 7.3. It turns out that, when the discount factor $\alpha$ is close to 1, discounted reward temporal–difference learning and average reward temporal–difference learning converge to approximations that are, in some sense, very close. However, the transient behavior of the two algorithms can be different, and there can be computational benefits to the average reward version. In a closing section, we comment on extensions and other available work with regards to average reward temporal–difference learning.

## 7.1 Definition and Convergence Theorem

Consider a stochastic process $\{x_t|t = 0, 1, 2, \ldots\}$ taking on values in a state space $\Re^d$ defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$. We denote the $\sigma$–field generated by

random variables $x_0, \ldots, x_t$ by $\mathcal{F}_t$, and we take $\mathcal{F}$ to be the smallest $\sigma$–field containing $\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2, \ldots$ as sub–$\sigma$–fields. We make the following assumption concerning the dynamics of this process.

**Assumption 7.1** *The process* $\{x_t | t = 0, 1, 2, \ldots\}$ *is Markov, stationary, and mixing.*

Because the process is stationary, there is a distribution $\pi$ such that $\pi(A) = \mathcal{P}\{x_t \in A\}$ for any set $A \in \mathcal{B}(\Re^d)$ and any nonnegative integer $t$. This distribution is also the limiting distribution that is guaranteed to exist by the mixing property. In particular, $\lim_{t\to\infty} \mathcal{P}\{x_t \in A | x_0 = x\} = \pi(A)$ for any $x \in \Re^d$ and $A \in \mathcal{B}(\Re^d)$. As in the context of temporal–difference learning with discounted rewards, we define a Hilbert space $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, which is endowed with an inner product $\langle J_1, J_2 \rangle_\pi = \int J_1(x) J_2(x) \pi(dx)$ and a norm $\|J\|_\pi = \langle J, J \rangle_\pi^{1/2}$.

By the Markov property, there exists a transition probability function $P$ on $\Re^d \times \mathcal{B}(\Re^d)$ such that, for any $A \in \mathcal{B}(\Re^d)$,

$$\mathcal{P}\{x_{t+1} \in A | \mathcal{F}_t\} = P(x_t, A),$$

and $P(\cdot, A)$ is measurable. We will also use $P$ to denote an operator given by

$$(PJ)(x) = \int J(y) P(x, dy).$$

We make an additional assumption that was not present in our analysis of temporal–difference learning with discounted rewards.

**Assumption 7.2** *There exists a scalar* $\beta \in [0, 1)$ *such that* $\|PJ\|_\pi \leq \beta \|J\|_\pi$ *for all* $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ *that are orthogonal to* $e$ *(i.e.,* $\langle J, e \rangle_\pi = 0$*).*

Recall that Lemma 4.12 states that, under Assumption 7.1, $\|PJ\|_\pi \leq \|J\|_\pi$ for all $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. Assumption 7.2 presents a slightly more restrictive condition. Intuitively, $\beta$ is related to the "mixing rate" of the Markov process. Observe, for instance, that this assumption is equivalent to having $\|P^t J - \langle J, e \rangle_\pi e\|_\pi$ decay at a uniform geometric rate for all $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. The assumption is satisfied in many situations of interest, and we provide one example as an illustration.

> **Example 7.1** Consider an irreducible aperiodic finite state Markov chain with a transition matrix $P$. It is well known that the largest eigenvalue of this matrix is equal to one, and every other eigenvalue is strictly less than one [28]. Furthermore, the right eigenvector corresponding to the largest eigenvalue is $e = (1, 1, \ldots, 1, 1)'$. It is straightforward to show that the validity of Assumption 7.2 follows as a consequence.

Let $g \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ be a reward function, and let the average reward be denoted by $\mu^* = \mathrm{E}[g(x_t)]$. The differential value function $J^* : \Re^d \mapsto \Re$ is then defined by

$$J^*(x) = \mathrm{E}\left[\sum_{t=0}^{\infty} (g(x_t) - \mu^*) \Big| x_0 = x\right],$$

(we will establish later that $J^*$ is in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$).

Let $\phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ be a set of basis functions, and let $\Pi$ denote the projection operator that projects onto their span. The following assumption applies to the basis functions.

**Assumption 7.3** *The functions $e, \phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ are linearly independent.*

Note that this assumption is different from Assumption 4.6, pertaining to basis functions used in the context of discounted rewards. In particular, we now require that the function $e$ is not contained within the span of the basis functions.

As in previous chapters, let a vector–valued function $\phi : \Re^d \mapsto \Re^K$, be defined by $\phi(x) = (\phi_1(x), \ldots, \phi_K(x))'$. Also, let an operator $\Phi : L_2(\Re^d, \mathcal{B}(\Re^d), \pi) \mapsto L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ be defined by

$$\Phi r = \sum_{k=1}^{K} r(k)\phi_k,$$

for any $r = (r(1), \ldots, r(K))'$.

The algorithm we propose is initialized with a weight vector $r_0 \in \Re^K$ and a scalar $\mu_0$ and recursively generates sequences $\{r_t | t = 1, 2, 3, \ldots\}$ and $\{\mu_t | t = 1, 2, 3, \ldots\}$. The latter sequence is generated according to

$$\mu_{t+1} = (1 - \eta_t)\mu_t + \eta_t g(x_t).$$

For each $t = 0, 1, 2, \ldots$, given $r_t$, a temporal difference $d_t$ is defined by

$$d_t = g(x_t) - \mu_t + (\Phi r_t)(x_{t+1}) - (\Phi r_t)(x_t).$$

This temporal difference is then used in generating $r_{t+1}$ according to

$$r_{t+1} = r_t + \gamma_t d_t z_t,$$

where $\{\gamma_t | t = 0, 1, 2, \ldots\}$ is a sequence of scalar step sizes and $\{z_t | t = 0, 1, 2, \ldots\}$ is a sequence of eligibility vectors taking on values in $\Re^K$, defined by

$$z_t = \sum_{\tau=0}^{t} \lambda^{t-\tau} \phi(x_\tau).$$

Unlike the discounted case, we will only consider values of $\lambda$ in $[0, 1)$, because the eligibility vector becomes unstable when $\lambda = 1$.

We make the following assumption concerning the step sizes.

**Assumption 7.4** *The sequence $\{\gamma_t | t = 0, 1, 2, \ldots\}$ is prespecified (deterministic), nonincreasing, and satisfies*

$$\sum_{t=0}^{\infty} \gamma_t = \infty \qquad \text{and} \qquad \sum_{t=0}^{\infty} \gamma_t^2 < \infty.$$

*Furthermore, there exists a positive scalar $C$ such that $\eta_t = C\gamma_t$ for all $t$.*

Our final assumption imposes stability requirements and is exactly the same as Assumption 4.8.

**Assumption 7.5** *The following conditions hold:*

1. *There exists scalars $C$ and $q$ such that, for all $x \in \Re^d$,*

$$|g(x)| \leq C(1 + \|x\|_2^q) \qquad \text{and} \qquad \|\phi(x)\|_2 \leq C(1 + \|x\|_2^q).$$

2. *For any $q > 0$, there exists a scalar $\mu_q$ such that for all $x \in \Re^d$ and $t = 0, 1, 2, \ldots$,*

$$\mathrm{E}\Big[\|x_t\|_2^q | x_0\Big] \leq \mu_q(1 + \|x_0\|_2^q) \qquad \text{and} \qquad \mathrm{E}\Big[\|\phi(x_t)\|_2^q | x_0\Big] \leq \mu_q(1 + \|\phi(x_0)\|_2^q).$$

3. *There exists scalars $C$ and $q$ such that, for all $x \in \Re^d$ and $m = 0, 1, 2, \ldots$,*

$$\sum_{t=0}^{\infty} \Big\| \mathrm{E}[\phi(x_t)\phi'(x_{t+m}) | x_0 = x] - \lim_{\tau \to \infty} \mathrm{E}[\phi(x_\tau)\phi'(x_{\tau+m})] \Big\|_2 \leq C(1 + \|x\|_2^q),$$

*and*

$$\sum_{t=0}^{\infty} \Big\| \mathrm{E}[\phi(x_t)g(x_{t+m}) | x_0 = x] - \lim_{\tau \to \infty} \mathrm{E}[\phi(x_\tau)g(x_{\tau+m})] \Big\|_2 \leq C(1 + \|x\|_2^q).$$

(We use $\|\cdot\|_2$ to denote the standard Euclidean norm on finite dimensional vectors and the Euclidean–induced norm on finite matrices.)

We define a variant of the TD($\lambda$) operator that is analogous to that employed in the analysis of discounted reward temporal–difference learning. In particular, for any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ that is orthogonal to $e$, let

$$(T^{(\lambda)}J)(x) = (1 - \lambda) \sum_{m=0}^{\infty} \lambda^m \mathrm{E}\left[ \sum_{t=0}^{m} (g(x_t) - \mu^*) + J(x_{m+1}) \Big| x_0 = x \right].$$

We will establish later that $T^{(\lambda)}$ maps the space orthogonal to $e$ into itself.

In assessing the approximation error, we will employ a different metric from that used in previous chapters. We will be concerned only with the relative values of a function. In particular, given an approximation $\tilde{J}$ to the differential value function $J^*$, error will be interpreted as

$$\|\Upsilon(\tilde{J} - J^*)\|_\pi,$$

where $\Upsilon$ denotes the projection operator that projects onto the space orthogonal to $e$. Note that this operator satisfies

$$\Upsilon J = J - \langle J, e \rangle_\pi e,$$

for all $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$.

We define a set of functions $\overline{\phi}_k = \Upsilon\phi_k$ for $k = 1, \ldots, K$. Also, let $\overline{\Pi}$ be the projection operator that projects onto the span of $\overline{\phi}_1, \ldots, \overline{\phi}_K$.

We now state the main result of this chapter.

**Theorem 7.6** *Under Assumptions 7.1–7.5, for any $\lambda \in [0, 1)$,*

1. *The value function $J^*$ is in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ and is orthogonal to $e$.*

2. *The sequences $\{\mu_t | t = 0, 1, 2, \ldots\}$ and $\{r_t | t = 0, 1, 2, \ldots\}$ converge to the average reward $\mu^*$ and some vector $r^* \in \Re^K$ almost surely.*

3. *The limit of convergence $r^*$ is the unique solution of the equation*

$$\overline{\Pi} T^{(\lambda)} \overline{\Phi} r^* = \overline{\Phi} r^*.$$

4. *The limit of convergence $r^*$ satisfies*

$$\|\Upsilon(\Phi r^* - J^*)\|_\pi \leq \frac{1}{\sqrt{1 - \kappa^2}} \|\overline{\Pi} J^* - J^*\|_\pi,$$

*where $\kappa$ is the contraction factor of $\overline{\Pi} T^{(\lambda)}$ and satisfies*

$$\kappa \leq \frac{\beta(1 - \lambda)}{1 - \lambda\beta} \leq \beta.$$

Except for the fact that we use a new error metric, statements (1)–(3) are entirely analogous to statements (1)–(3) of Theorem 4.9, pertaining to the case of discounted rewards. Statement (4) implies that if each basis function $\phi_k$ is decomposed into a multiple of $e$ plus an orthogonal component $\overline{\phi}_k$, only the latter influences the limit of convergence $r^*$. Finally, statement (5) looks like the error bound of Theorem 4.9, except that the scalar $\beta$, which represents a "mixing factor," substitutes for the role of a discount factor.

## 7.2 Proof of Theorem 7.6

We now provide a proof of the result. We begin with some preliminary lemmas that characterize the value function, the TD($\lambda$) operator, and the "steady–state" behavior of the updates. We then state a corollary to Theorem 3.10. Section 7.2.2 integrates these items with results from Chapter 3 in order to prove Theorem 4.9.

### 7.2.1 Preliminary Lemmas

Let $\mathcal{J}$ be the subset of $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ that is orthogonal to $e$. It is easy to verify that $(\mathcal{J}, \langle \cdot, \cdot \rangle_\pi)$ is a Hilbert space. Our first lemma establishes that $J^*$ is in $\mathcal{J}$.

**Lemma 7.7** *Under Assumptions 7.1 and 7.2, $J^*$ is well-defined and in $\mathcal{J}$. Furthermore,*

$$J^* = \sum_{t=0}^{\infty} P^t(g - \mu^*).$$

**Proof:** To establish that $J^*$ is well defined (for almost all $x$),

$$\begin{aligned}
\mathrm{E}\left[\sum_{t=0}^{\infty} |g(x_t) - \mu^*|\right] &= \sum_{t=0}^{\infty} \mathrm{E}[(P^t|g - \mu^* e|)(x_0)] \\
&\leq \sum_{t=0}^{\infty} \left(\mathrm{E}[(P^t|g - \mu^* e|)^2(x_0)]\right)^{\frac{1}{2}} \\
&= \sum_{t=0}^{\infty} \|P^t|g - \mu^* e|\|_\pi \\
&\leq \frac{1}{1-\beta}\|g - \mu^* e\|_\pi,
\end{aligned}$$

where the Tonelli–Fubini theorem, the definition of $P$, Jensen's inequality, and Assumption 7.2, have been used. Note that Assumption 7.2 could be used because

$$\langle g - \mu^* e, e\rangle_\pi = \mathrm{E}[g(x_0) - \mu^*] = 0,$$

as a consequence of the definition of $\mu^*$.

By another application of the Tonell–Fubini theorem,

$$J^*(x) = \mathrm{E}\left[\sum_{t=0}^{\infty}(g(x_t) - \mu^*)\Big|x_0 = x\right] = \sum_{t=0}^{\infty}\mathrm{E}[g(x_t) - \mu^*|x_0 = x] = \sum_{t=0}^{\infty}(P^t(g - \mu^* e))(x).$$

It then follows from Assumption 7.2 that

$$\|J^*\|_\pi \leq \frac{1}{1-\beta}\|g - \mu^* e\|_\pi,$$

and therefore, $J^*$ is in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. Furthermore, since for any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ and any nonnegative integer $t$,

$$\langle e, P^t J\rangle_\pi = \mathrm{E}[(P^t J)(x_0)] = \mathrm{E}[J(x_0)] = \langle e, J\rangle_\pi,$$

it follows from the fact that $g - \mu^* e$ is in $\mathcal{J}$ that $J^*$ is also in $\mathcal{J}$. **Q.E.D.**

The next lemma characterizes relevant properties of the TD($\lambda$) operator. Unlike in the discounted case, the operator corresponding to the average reward setting is not a contraction on $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. However, it is a contraction on $(\mathcal{J}, \langle\cdot,\cdot\rangle_\pi)$.

**Lemma 7.8** *Let Assumptions 7.1 and 7.2 and hold. Then,*

1. *For any* $\lambda \in [0,1)$ *and any* $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$T^{(\lambda)}J = (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\left(\sum_{t=0}^{m}P^t(g-\mu^*e)+P^{m+1}J\right),$$

*which is in* $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$.

2. *For any* $\lambda \in [0,1)$ *and any* $J_1, J_2 \in \mathcal{J}$,

$$\|T^{(\lambda)}J_1 - T^{(\lambda)}J_2\|_\pi \leq \frac{\beta(1-\lambda)}{1-\beta\lambda}\|J_1 - J_2\|_\pi \leq \beta\|J_1 - J_2\|_\pi.$$

3. *For any* $\lambda \in [0,1)$, $J^*$ *is the unique solution (in* $(\mathcal{J}, \langle\cdot,\cdot\rangle_\pi)$*) to* $T^{(\lambda)}J^* = J^*$.

**Proof:** For any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$,

$$\begin{aligned}
(T^{(\lambda)}J)(x) &= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\mathrm{E}\left[\sum_{t=0}^{m}(g(x_t)-\mu^*)+J(x_{m+1})\Big|x_0=x\right] \\
&= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\left(\sum_{t=0}^{m}\mathrm{E}[g(x_t)-\mu^*|x_0=x]+\mathrm{E}[J(x_{m+1})|x_0=x]\right) \\
&= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\left(\sum_{t=0}^{m}(P^t(g-\mu^*e))(x)+(P^{m+1}J)(x)\right).
\end{aligned}$$

The fact that $T^{(\lambda)}J$ is in $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ follows easily from Assumption 7.2 and Lemma 4.12, and Statement (1) follows.

Statement (2) follows from Assumption 7.2. In particular, for any $J_1, J_2 \in \mathcal{J}$,

$$\begin{aligned}
\|T^{(\lambda)}J_1 - T^{(\lambda)}J_2\|_\pi &= \left\|(1-\lambda)\sum_{m=0}^{\infty}\lambda^m P^{m+1}(J_1-J_2)\right\|_\pi \\
&\leq (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\beta^{m+1}\|J_1-J_2\|_\pi \\
&= \frac{\beta(1-\lambda)}{1-\beta\lambda}\|J_1-J_2\|_\pi.
\end{aligned}$$

Statement (3) follows from Lemma 7.7 and some simple algebra:

$$\begin{aligned}
T^{(\lambda)}J^* &= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\Big(\sum_{t=0}^{m}P^t(g-\mu^*e)+P^{m+1}J^*\Big) \\
&= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\Big(\sum_{t=0}^{m}P^t(g-\mu^*e)+P^{m+1}\sum_{t=0}^{\infty}P^t(g-\mu^*e)\Big) \\
&= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m\Big(\sum_{t=0}^{\infty}P^t(g-\mu^*e)\Big) \\
&= (1-\lambda)\sum_{m=0}^{\infty}\lambda^m J^*
\end{aligned}$$

$$= J^*.$$

The fact that $T^{(\lambda)}$ is a contraction on $(\mathcal{J}, \langle \cdot, \cdot \rangle_\pi)$ implies that the fixed point is unique (in the sense of $(\mathcal{J}, \langle \cdot, \cdot \rangle_\pi)$). **Q.E.D.**

Let a process $\{y_t | t = 0, 1, 2, \ldots\}$ be defined by $y_t = (x_t, x_{t+1}, z_t)$, and let a function $s$ be defined by

$$s_k(\theta, y) = \Big(g(x) - \mu + (\Phi r)(\overline{x}) - (\Phi r)(x)\Big) z_k,$$

for $\theta = (r, \mu)$ and $y = (x, \overline{x}, z)$ and $k = 1, \ldots, K$. Furthermore, let

$$s_{K+1}(\theta, y) = C(g(x) - \mu),$$

for $\theta = (r, \mu)$ and $y = (x, \overline{x}, z)$, where $C = \eta_t / \gamma_t$ (this constant is independent of time by Assumption 7.4). The average reward temporal–difference learning algorithm can then be rewritten in the form

$$\theta_{t+1} = \theta_t + \gamma_t s(\theta_t, y_t),$$

where we let $\theta_t = (r_t, \mu_t)$.

Our next Lemma establishes that the "steady–state" update direction is of the form related to but different from that studied in Section 3.4.2. We will employ the notation $\underset{t \to \infty}{\mathrm{E}}[\cdot]$ as shorthand for $\lim_{t \to \infty} \mathrm{E}[\cdot]$.

**Lemma 7.9** *Under Assumptions 7.1 and 7.2,*

$$\underset{t \to \infty}{\mathrm{E}}[s_k(\theta, y_t)] = \Big\langle \overline{\phi}_k, T^{(\lambda)} \overline{\Phi} r - \overline{\Phi} r \Big\rangle_\pi + \frac{1}{1 - \lambda} \langle \phi_k, (\mu^* - \mu) e \rangle_\pi,$$

*for all $k = 1, \ldots, K$, $r \in \Re^K$, $\mu \in \Re$, and $\lambda \in [0, 1)$, where $\theta = (r, \mu)$. Furthermore,*

$$\underset{t \to \infty}{\mathrm{E}}[s_{K+1}(\theta, y_t)] = C(\mu^* - \mu),$$

*for all $r \in \Re^K$, $\mu \in \Re$, and $\lambda \in [0, 1)$, where $\theta = (r, \mu)$.*

**Proof:** By arguments from the proof of Lemma 4.15, for any $r \in \Re^K$, $\mu \in \Re$, $k = 1, \ldots, K$, and $\lambda \in [0, 1)$, letting $\theta = (r, \mu)$,

$$\begin{aligned} \underset{t \to \infty}{\mathrm{E}}[s_k(\theta, y_t)] &= \sum_{m=0}^{\infty} \lambda^m \Big\langle \phi_k, P^m (g - \mu e + P \Phi r - \Phi r) \Big\rangle_\pi \\ &= \sum_{m=0}^{\infty} \lambda^m \Big\langle \phi_k, P^m (g - \mu^* e + P \Phi r - \Phi r) \Big\rangle_\pi + \frac{1}{1 - \lambda} \langle \phi_k, (\mu^* - \mu) e \rangle_\pi. \end{aligned}$$

Note that $g - \mu^* e$ and $P \Phi r - \Phi r$ are both orthogonal to $e$ (the latter follows from the fact that $\langle e, PJ \rangle_\pi = \langle e, J \rangle_\pi$ for all $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$). It follows that

$$\underset{t \to \infty}{\mathrm{E}}[s_k(\theta, y_t)] = \sum_{m=0}^{\infty} \lambda^m \Big\langle \overline{\phi}_k, P^m (g - \mu^* e + P \overline{\Phi} r - \overline{\Phi} r) \Big\rangle_\pi + \frac{1}{1 - \lambda} \langle \phi_k, (\mu^* - \mu) e \rangle_\pi.$$

Note that for any $\lambda \in [0,1)$ and $J \in \mathcal{J}$,

$$\sum_{m=0}^{\infty} \lambda^m P^m J = (1-\lambda) \sum_{m=0}^{\infty} \lambda^m \sum_{t=0}^{m} P^t J.$$

We therefore have

$$\begin{aligned} \mathop{\mathrm{E}}_{t\to\infty}[s_k(\theta, y_t)] &= \left\langle \overline{\phi}_k, (1-\lambda) \sum_{m=0}^{\infty} \lambda^m \sum_{t=0}^{m} P^t \Big( g - \mu^* e + P\overline{\Phi} r - (\overline{\Phi} r) \Big) \right\rangle_\pi \\ &\quad + \frac{1}{1-\lambda} \langle \phi_k, (\mu^* - \mu) e \rangle_\pi \\ &= \left\langle \overline{\phi}_k, T^{(\lambda)} \overline{\Phi} r - \overline{\Phi} r \right\rangle_\pi + \frac{1}{1-\lambda} \langle \phi_k, (\mu^* - \mu) e \rangle_\pi. \end{aligned}$$

The remaining part of the lemma, stating that

$$\mathop{\mathrm{E}}_{t\to\infty}[s_{K+1}(\theta, y_t)] = C(\mu^* - \mu),$$

simply follows from the fact that $\mu^* = \mathop{\mathrm{E}}_{t\to\infty}[g(x_t)]$. **Q.E.D.**

The expectation $\mathop{\mathrm{E}}_{t\to\infty}[s(\theta, y_t)]$ characterized by Lemma 7.9 does not take on the form studied in Chapter 3, and consequently, Theorem 3.10 may not apply directly to our algorithm. However, a simple corollary of that theorem suits our purposes.

**Corollary 7.10** *The conclusions of Theorem 3.10 remain valid if condition (1) is replaced by*

- *There exists some $r^* \in \Re^K$ and a diagonal matrix $D$ with positive diagonal entries such that $(r - r^*)' D \overline{s}(r) < 0$, for all $r \neq r^*$, and $\overline{s}(r^*) = 0$.*

It is not difficult to show that this corollary follows from Theorem 3.10. However, like Theorem 3.10, it is also a special case of Theorem 17 of [7]. We therefore omit the proof.

### 7.2.2 Proof of the Theorem

We will establish convergence using Corollary 7.2.1. Let the function $\overline{s} : \Re^{K+1} \mapsto \Re^{K+1}$ required by the conditions be defined by $\overline{s}(\theta) = \mathop{\mathrm{E}}_{t\to\infty}[s(\theta, y_t)]$. (Note that the variables $r$ and $K$ from the corollary corresponds to $\theta$ and $K+1$ for our current setting.) Also, though we have considered until this point a process $y_t = (x_t, x_{t+1}, z_t)$, for the purposes of validating conditions of Corollary 7.2.1, we let $y_t = (x_t, x_{t+1}, z_t, \phi(x_{t+1}))$. Note that there is a one–to–one mapping between the two versions of $y_t$, so the update direction $s$ is still a function of $\theta_t$ and $y_t$.

Validity of conditions (2)–(6) can be verified using arguments analogous to those employed in the proof of Theorem 4.9. To avoid repetition, we will omit proofs that these conditions are valid. We are left, however, with condition (1), which we now address.

Since $T^{(\lambda)}$ is a contraction on $(\mathcal{J}, \langle\cdot,\cdot\rangle_\pi)$ and $\overline{\Pi}$ is a projection whose range is in $\mathcal{J}$, it follows from Theorem 3.9 that $\overline{\Pi}T^{(\lambda)}$ is a contraction on $(\mathcal{J}, \langle\cdot,\cdot\rangle_\pi)$. Furthermore, since $e, \phi_1, \ldots, \phi_K \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ are linearly independent (Assumption 7.3), $\overline{\phi}_1, \cdots, \overline{\phi}_K \in \mathcal{J}$ are linearly independent, and it follows that there exists a unique vector $r^* \in \Re^K$ such that $\overline{\Phi}r^*$ is the unique fixed point of $\overline{\Pi}T^{(\lambda)}$ (in $(\mathcal{J}, \langle\cdot,\cdot\rangle_\pi)$).

Let $\theta^* = (r^*, \mu^*)$. It is easy to see that $\overline{s}(\theta^*) = 0$. Let $D \in \Re^{(K+1)\times(K+1)}$ be a diagonal matrix with the $(K+1)$th entry (i.e., that corresponding to the average reward estimate) equal to a positive scalar $\ell$ and every other diagonal entry equal to one. Then, given a vector $\theta = (r, \mu)$,

$$\begin{aligned}(\theta - \theta^*)' D\overline{s}(\theta) &= \sum_{k=1}^{K}(\theta(k) - \theta^*(k))\left(\left\langle\overline{\phi}_k, T^{(\lambda)}\overline{\Phi}r - \overline{\Phi}r\right\rangle_\pi - \frac{1}{1-\lambda}\langle\phi, (\mu^* - \mu)e\rangle_\pi\right) \\ &\quad -C(\mu^* - \mu)^2 \\ &= \left\langle\overline{\Phi}(r - r^*), T^{(\lambda)}\overline{\Phi}r - \overline{\Phi}r\right\rangle_\pi - \frac{1}{1-\lambda}\langle\phi, (\mu^* - \mu)e\rangle_\pi - C(\mu^* - \mu)^2.\end{aligned}$$

Note that

$$\frac{1}{1-\lambda}|\langle\overline{\Phi}(r - r^*), (\mu^* - \mu)e\rangle_\pi| \leq \frac{|\mu^* - \mu|}{1-\lambda}\|\overline{\Phi}(r - r^*)\|_\pi \leq C_1|\mu^* - \mu| \cdot \|r - r^*\|_2,$$

for some constant $C_1 > 0$. Furthermore, by Theorem 3.11(4),

$$\begin{aligned}\left\langle\overline{\Phi}(r - r^*), T^{(\lambda)}\overline{\Phi}r - \overline{\Phi}r\right\rangle_\pi &= \sum_{k=1}^{K}(r_k - r_k^*)\langle\overline{\phi}_k, T^{(\lambda)}\overline{\Phi}r - \overline{\Phi}r\rangle_\pi \\ &\leq -C_2\|r - r^*\|_2^2,\end{aligned}$$

for some constant $C_2 > 0$. It follows that

$$(\theta - \theta^*)' D\overline{s}(\theta) \leq -\ell C(\mu^* - \mu)^2 + C_1|\mu^* - \mu| \cdot \|r - r^*\|_2 - C_2\|r - r^*\|_2^2,$$

and by setting $\ell$ to a value satisfying $c_1^2 \leq 4\ell C C_2$,

$$(\theta - \theta^*)' D\overline{s}(\theta) < 0,$$

for all $\theta \neq \theta^*$. Condition (1) is therefore valid.

Corollary 7.2.1 implies that $\theta_t$ converges to $\theta^*$ (almost surely). Therefore, $\mu_t$ and $r_t$ both converge, as stated in Theorem 7.6(2). Statement (1) of the theorem is established by Lemma 7.7.

We have already established Statement (3) earlier ($r^*$ is the unique solution to $\overline{\Pi}T^{(\lambda)}\overline{\Phi}r^* = \overline{\Phi}r^*$). Statement (4) follows from Theorem 3.9, since $\overline{\Phi}r^*$ is the unique fixed point in $(\mathcal{J}, \langle\cdot,\cdot\rangle_\pi)$ of a composition between a projection $\overline{\Pi}$ and a contraction $T^{(\lambda)}$ (the bound of $\beta(1-\lambda)/(1-\beta\lambda)$ on the contraction factor is from Lemma 7.8). **Q.E.D.**

## 7.3 Discounted Versus Averaged Rewards

Let $J^{(\alpha)}$ denote the value function corresponding to a discount factor $\alpha \in [0, 1)$. In particular,

$$J^{(\alpha)}(x) = \mathrm{E}\left[\sum_{t=0}^{\infty} \alpha^t g(x_t) \middle| x_0 = x\right].$$

Then, as $\alpha$ approaches 1, the projection $\Upsilon J^{(\alpha)}$ converges to the differential value function $J^*$ (in the sense of $L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$. To see this, note that $\Upsilon P^t g = P^t(g - \mu e)$ for all $t$, and therefore,

$$\Upsilon J^{(\alpha)} = \Upsilon\left(\sum_{t=0}^{\infty} \alpha^t P^t g\right) = \sum_{t=0}^{\infty} \alpha^t P^t (g - \mu e).$$

Under Assumption 7.2, it follows that

$$\|\Upsilon J^{(\alpha)} - J^*\|_\pi \leq \sum_{t=0}^{\infty} (1 - \alpha^t) \|P^t(g - \mu e)\|_\pi \leq \sum_{t=0}^{\infty} (1 - \alpha^t)\beta^t,$$

which approaches 0 as $\alpha$ approaches 1.

Another interesting fact is that $\langle e, J^{(\alpha)} \rangle_\pi = \mu^*/(1 - \alpha)$. To see this, note that $\langle e, P^t g \rangle_\pi = \mu^*$ for all $t$, and therefore

$$\langle e, J^{(\alpha)} \rangle_\pi = \left\langle e, \sum_{t=0}^{\infty} \alpha^t P^t g \right\rangle_\pi = \sum_{t=0}^{\infty} \alpha^t \langle e, g \rangle_\pi = \frac{\mu^*}{1 - \alpha}.$$

Based on the observations we have made, it is natural to think of the value function $J^{(\alpha)}$ as providing the average reward and an approximation to the differential value function in the following sense:

$$J^* \approx \Upsilon J^{(\alpha)} \qquad \text{and} \qquad \mu^* = (1 - \alpha)\langle e, J^{(\alpha)} \rangle_\pi.$$

Average and discounted temporal–difference learning exhibit some related relationships, which we explore in this section.

Recall that, in average reward temporal–difference learning, $e$ is not within the span of the basis functions. In some sense, the average reward estimate $\mu$ plays a role that, in the case of discounted reward temporal–difference learning, would be played by a basis function that is aligned with $e$. Hence, when we compare average and discounted temporal–difference learning, we include an extra basis function in the latter case. In particular we study the two following situations.

1. Executing average reward temporal–difference learning with basis functions $\phi_1, \ldots, \phi_K$ that satisfy Assumption 7.3.

2. Executing discounted reward temporal–difference learning with basis functions $\phi_1, \ldots, \phi_K, e$ (a total of $K + 1$ basis functions). Note that, by Assumption 7.3, these basis functions are linearly independent, as required by Assumption 4.6.

The next subsection compares the ultimate approximations generated by these two approaches, while Section 7.3.2 discusses differences in the incremental updates that occur during execution of the algorithms. It turns out that the ultimate approximations are very similar (if $\alpha$ is close to 1), but the transient behavior of the algorithms can be significantly different, and in fact, there may be computational advantages to the average reward version of temporal–difference learning.

To keep the exposition simple, we will focus on the case of $\lambda = 0$. Entirely analogous arguments apply in the more general setting of $\lambda \in [0, 1)$.

### 7.3.1 Limits of Convergence

Let us denote the parameterizations employed in average and discounted temporal–difference learning by $\tilde{J}^a$ and $\tilde{J}^d$, respectively. In particular, given weight vectors $r \in \Re^K$ and $\overline{r} \in \Re^{K+1}$, let

$$\tilde{J}^a(x, r) = \sum_{k=1}^{K} r(k)\phi_k \qquad \text{and} \qquad \tilde{J}^d(x, \overline{r}) = \sum_{k=1}^{K} \overline{r}(k)\phi_k + \overline{r}(K+1)e.$$

Let $r^* \in \Re^K$ and $r^{(\alpha)} \in \Re^{K+1}$ denote the limits of convergence given these parameterizations (with $\lambda = 0$). For shorthand, let $\tilde{J} = \tilde{J}^a(\cdot, r^*)$ and $\tilde{J}^{(\alpha)} = \tilde{J}^d(\cdot, r^{(\alpha)})$. Then, it turns out that $\lim_{\alpha \uparrow 1} \Upsilon \tilde{J}^{(\alpha)} = \Upsilon \tilde{J}$ and $\langle e, \tilde{J}^{(\alpha)} \rangle_\pi = \mu^*/(1-\alpha)$. Hence, we can think of discounted temporal–difference learning as approximating the results of average reward temporal difference learning in the following sense:

$$\Upsilon \tilde{J} \approx \Upsilon \tilde{J}^{(\alpha)} \qquad \text{and} \qquad \mu^* = (1-\alpha)\langle e, \tilde{J}^{(\alpha)} \rangle_\pi.$$

Let us now establish that our claims are indeed true. We denote the TD(0) operators for the average and discounted cases by $T$ and $T^{(\alpha)}$, respectively. Note that for any $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$ and $\alpha \in [0, 1)$,

$$\left\| \Upsilon T J - \Upsilon T^{(\alpha)} J \right\|_\pi = (1-\alpha) \left\| \Upsilon P J \right\|_\pi = (1-\alpha) \left\| J \right\|_\pi,$$

which approaches 0 as $\alpha$ approaches 1. Hence, $\lim_{\alpha \uparrow 1} \Upsilon T^{(\alpha)} J = \Upsilon T J$.

Let $\overline{\Pi}$ be a projection onto the span of $\overline{\phi}_1, \ldots, \overline{\phi}_K$ (recall that $\overline{\phi}_k = \Upsilon \phi_k$). Hence the projection of a function $J$ onto the span of $\phi_1, \ldots, \phi_K, e$ is given by $\overline{\Pi} J + e\langle e, J \rangle_\pi$. By Theorem 4.9(3), $\tilde{J}^{(\alpha)}$ is the unique fixed point of $\overline{\Pi} T^{(\alpha)}(\cdot) + e\langle e, T^{(\alpha)}(\cdot) \rangle_\pi$. It follows that $\Upsilon \tilde{J}^{(\alpha)}$ is the unique fixed point of $\overline{\Pi} T^{(\alpha)}$ and $\langle e, \tilde{J}^{(\alpha)} \rangle_\pi = \langle e, T^{(\alpha)} \tilde{J}^{(\alpha)} \rangle_\pi$ Furthermore, it follows from Theorem 7.6(3)–(4), that $\Upsilon \tilde{J}$ is the unique fixed point of $\overline{\Pi} T$.

By the triangle inequality,

$$\begin{aligned}
\left\| \Upsilon \tilde{J} - \Upsilon \tilde{J}^{(\alpha)} \right\|_\pi &\leq \left\| \Upsilon \tilde{J} - \overline{\Pi} T^{(\alpha)} \Upsilon \tilde{J} \right\|_\pi + \left\| \overline{\Pi} T^{(\alpha)} \Upsilon \tilde{J} - \Upsilon \tilde{J}^{(\alpha)} \right\|_\pi \\
&= \left\| \overline{\Pi} T \Upsilon \tilde{J} - \overline{\Pi} T^{(\alpha)} \Upsilon \tilde{J} \right\|_\pi + \left\| \overline{\Pi} T^{(\alpha)} \Upsilon \tilde{J} - \Upsilon \tilde{J}^{(\alpha)} \right\|_\pi \\
&\leq \left\| \Upsilon T \Upsilon \tilde{J} - \Upsilon T^{(\alpha)} \Upsilon \tilde{J} \right\|_\pi + \beta \left\| \Upsilon \tilde{J} - \Upsilon \tilde{J}^{(\alpha)} \right\|_\pi,
\end{aligned}$$

where the final inequality relies on two facts: (1) the range of the projection $\overline{\Pi}$ is a subspace of the range of the projection $\Upsilon$ and (2) $\overline{\Pi}T^{(\alpha)}$ is a contractions with contraction factor less than $\beta$ and fixed point $\Upsilon\tilde{J}^{(\alpha)}$. It follows that

$$\|\Upsilon\tilde{J} - \Upsilon\tilde{J}^{(\alpha)}\|_\pi \leq \frac{1}{1-\beta}\|\Upsilon T\Upsilon\tilde{J} - \Upsilon T^{(\alpha)}\Upsilon\tilde{J}\|_\pi.$$

Since $\lim_{\alpha\uparrow 1}\Upsilon T^{(\alpha)}J = \Upsilon TJ$ for all $J \in L_2(\Re^d, \mathcal{B}(\Re^d), \pi)$, it follows that $\lim_{\alpha\uparrow 1}\Upsilon\tilde{J}^{(\alpha)} = \Upsilon\tilde{J}$.

To complete the arguments validating our earlier claims, we have

$$\langle e, \tilde{J}^{(\alpha)}\rangle_\pi = \langle e, T^{(\alpha)}\tilde{J}^{(\alpha)}\rangle_\pi = \langle e, g + \alpha P\tilde{J}^{(\alpha)}\rangle_\pi = \mu^* + \alpha\langle e, \tilde{J}^{(\alpha)}\rangle_\pi,$$

and it follows that

$$\langle e, \tilde{J}^{(\alpha)}\rangle_\pi = \frac{\mu^*}{1-\alpha}.$$

### 7.3.2 Transient Behavior

As shown in the previous subsection, if $\alpha$ is close to 1, the limit of convergence of discounted temporal–difference learning approximates that of average reward temporal–difference learning. It turns out that the same is not true of the iterates generated during the course of computation. In this subsection, discuss the similarities between the iterations as well as the cause of significant differences in transient behavior.

Let the sequences generated by average reward temporal–difference learning be denoted by $\{r_t|t = 0, 1, 2, \ldots\}$ and $\{\mu_t|t = 0, 1, 2, \ldots\}$, and let the sequence generated by discounted temporal–difference learning be denoted by $\{r_t^{(\alpha)}|t = 0, 1, 2, \ldots\}$. Hence, $\lim_{t\to\infty} r_t = r^*$, $\lim_{t\to\infty}\mu_t = \mu^*$, and $\lim_{t\to\infty} r_t^{(\alpha)} = r^{(\alpha)}$. We assume that the sequences are initialized such that $r_0(k) = r_0^{(\alpha)}(k)$ for $k = 1, \ldots, K$ and

$$\mu_0 = (1-\alpha)\left(\sum_{k=1}^{K} r_0^{(\alpha)}(k)\langle\phi_k, e\rangle_\pi + r_0^{(\alpha)}(K+1)\right).$$

Let $\tilde{J}_t = \tilde{J}^a(\cdot, r_t)$ and $\tilde{J}_t^{(\alpha)} = \tilde{J}^d(\cdot, r_t^{(\alpha)})$. Then, the constraints on initial weights imply that $\Upsilon\tilde{J}_0 = \Upsilon\tilde{J}_0^{(\alpha)}$ and $\mu_0 = (1-\alpha)\langle e, \tilde{J}_0^{(\alpha)}\rangle_\pi$. In the previous subsection, we showed that, when $\alpha$ is close to 1, $\lim_{t\to\infty}\Upsilon\tilde{J}_t \approx \lim_{t\to\infty}\Upsilon\tilde{J}_t^{(\alpha)}$ and $\lim_{t\to\infty}\mu_t = (1-\alpha)\lim_{t\to\infty}\langle e, \tilde{J}_t^{(\alpha)}\rangle_\pi$. The question we will address in this subsection is whether $\Upsilon\tilde{J}_t \approx \Upsilon\tilde{J}_t^{(\alpha)}$ and $\mu_t \approx (1-\alpha)\langle e, \tilde{J}_t^{(\alpha)}\rangle_\pi$ for all $t$, when $\alpha$ is close to 1.

Recall that $\overline{\phi}_k = \Upsilon\phi_k$ for $k = 1, \ldots, K$. The average reward temporal–difference learning update (for $\lambda = 0$) is given by

$$\begin{aligned} r_{t+1}(k) &= r_t(k) + \gamma_t\phi(x_t)\big(g(x_t) - \mu_t + \tilde{J}^a(x_{t+1}, r_t) - \tilde{J}^a(x_t, r_t)\big) \\ &= r_t(k) + \gamma_t\phi(x_t)\left(g(x_t) - \mu_t + \sum_{j=1}^{K}\phi_j(x_{t+1})r_t(j) - \sum_{j=1}^{K}\phi_j(x_t)r_t(j)\right) \end{aligned}$$

$$= r_t(k) + \gamma_t \phi(x_t) \left( g(x_t) - \mu_t + \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t(j) \right),$$

since the component of each $\phi_k$ that is aligned with $e$ evaluates to the same value for both $x_t$ and $x_{t+1}$.

Suppose that $\mu_t = (1-\alpha)\langle e, \tilde{J}_t^{(\alpha)} \rangle_\pi$. The update in the discounted case is then given by

$$\begin{aligned} r_{t+1}^{(\alpha)}(k) &= r_t^{(\alpha)}(k) + \gamma_t \phi_k(x_t) \big( g(x_t) + \alpha \tilde{J}^d(x_{t+1}, r_t^{(\alpha)}) - \tilde{J}^d(x_t, r_t^{(\alpha)}) \big) \\ &= r_t^{(\alpha)}(k) + \gamma_t \phi_k(x_t) \left( g(x_t) - \mu_t + \alpha \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t^{(\alpha)}(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t^{(\alpha)}(j) \right) \end{aligned}$$

for $k = 1, \ldots, K$. Note that this update becomes identical to that of average reward temporal–difference learning as $\alpha$ approaches 1.

The difference in the dynamics of the two algorithms comes with the update equations relating to the average reward estimates. In average reward temporal–difference learning, we have

$$\mu_{t+1} = \mu_t + C\gamma_t (g(x_t) - \mu_t),$$

for some positive constant $C$. In the discounted case, the change in the average reward estimate $\langle e, \tilde{J}_t^{(\alpha)} \rangle_\pi$ takes on a a significantly different form. In particular, supposing that $\mu_t = (1-\alpha)\langle e, \tilde{J}_t^{(\alpha)} \rangle_\pi$, we have

$$\begin{aligned} (1-\alpha)\langle e, \tilde{J}_{t+1}^{(\alpha)} \rangle_\pi &= (1-\alpha) \left\langle e, \sum_{k=1}^{K} r_{t+1}^{(\alpha)}(k)\phi_k + r_{t+1}^{(\alpha)}(K+1)e \right\rangle_\pi \\ &= (1-\alpha) \sum_{k=1}^{K} r_{t+1}^{(\alpha)}(k) \langle e, \phi_k \rangle_\pi + (1-\alpha) r_{t+1}^{(\alpha)}(K+1) \\ &= (1-\alpha) \sum_{k=1}^{K} \Bigg( r_t^{(\alpha)}(k) + \gamma_t \phi_k(x_t) \Big( g(x_t) - \mu_t \\ &\quad + \alpha \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t^{(\alpha)}(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t^{(\alpha)}(j) \Big) \Bigg) \langle e, \phi_k \rangle_\pi \\ &\quad + (1-\alpha) \Bigg( r_t^{(\alpha)}(K+1) + \gamma_t \Big( g(x_t) - \mu_t \\ &\quad + \alpha \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t^{(\alpha)}(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t^{(\alpha)}(j) \Big) \Bigg) \\ &= (1-\alpha) \left\langle e, \sum_{k=1}^{K} r_t^{(\alpha)}(k)\phi_k \right\rangle_\pi + (1-\alpha) r_t^{(\alpha)}(K+1) \\ &\quad + (1-\alpha)\gamma_t \left( \sum_{k=1}^{K} \langle e, \phi_k \rangle_\pi \phi_k(x_t) + 1 \right) \end{aligned}$$

$$
\begin{aligned}
& \left( g(x_t) - \mu_t + \alpha \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t^{(\alpha)}(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t^{(\alpha)}(j) \right) \\
= \; & (1-\alpha) \left\langle e, \sum_{k=1}^{K} r_t^{(\alpha)}(k) \phi_k \right\rangle_\pi + (1-\alpha) r_t^{(\alpha)}(K+1) \\
& + (1-\alpha)\gamma_t \left( \sum_{k=1}^{K} \langle e, \phi_k \rangle_\pi^2 + 1 \right) \\
& \left( g(x_t) - \mu_t + \alpha \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t^{(\alpha)}(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t^{(\alpha)}(j) \right) \\
& + (1-\alpha)\gamma_t \sum_{k=1}^{K} \langle e, \phi_k \rangle_\pi \overline{\phi}_k(x_t) \\
& \left( g(x_t) - \mu_t + \alpha \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t^{(\alpha)}(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t^{(\alpha)}(j) \right) \\
= \; & \mu_t + C\gamma_t (g(x_t) - \mu_t) \\
& + C\gamma_t \left( \alpha \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t^{(\alpha)}(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t^{(\alpha)}(j) \right) \\
& + (1-\alpha)\gamma_t \sum_{k=1}^{K} \langle e, \phi_k \rangle_\pi \overline{\phi}_k(x_t) \\
& \left( g(x_t) - \mu_t + \alpha \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t^{(\alpha)}(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t^{(\alpha)}(j) \right),
\end{aligned}
$$

for a constant

$$
C = (1-\alpha) \left( \sum_{k=1}^{K} \langle e, \phi_k \rangle_\pi^2 + 1 \right).
$$

Let us discuss the three terms involved in updating $(1-\alpha)\langle e, \tilde{J}_t^{(\alpha)} \rangle_\pi$. The first term

$$
C\gamma_t (g(x_t) - \mu_t),
$$

is the same as that involved in the update equation for $\mu_t$ in average reward temporal–difference learning. The second term

$$
C\gamma_t \left( \alpha \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t^{(\alpha)}(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t^{(\alpha)}(j) \right),
$$

is absent in average reward temporal–difference learning. Its expectation (in steady–state) is zero, but it generally takes on nonzero values that add "noise" to the updates.

Finally, the third term

$$(1-\alpha)\gamma_t \sum_{k=1}^{K} \langle e, \phi_k \rangle_\pi \overline{\phi}_k(x_t) \left( g(x_t) - \mu_t + \alpha \sum_{j=1}^{K} \overline{\phi}_j(x_{t+1}) r_t^{(\alpha)}(j) - \sum_{j=1}^{K} \overline{\phi}_j(x_t) r_t^{(\alpha)}(j) \right),$$

adds additional interference which is not necessarily even zero–mean. It is interesting to note that this term is equal to zero in the event that the basis functions $\phi_1, \ldots, \phi_K$ are orthogonal to $e$, which unfortunately, is not generally the case.

In summary, the update equation for discounted temporal–difference learning involves "noise" that is not present in the average reward case. This interferes not only with the evolution of the average reward estimate $(1-\alpha)\langle e, \tilde{J}_t^{(\alpha)} \rangle_\pi$, but also with $\Upsilon \tilde{J}_t^{(\alpha)}$, since the average reward estimate enters into the computation of the latter. Consequently, there may be computational advantages to average reward temporal–difference learning, as observed in previous empirical work [49]. Similar observations have also been made in experiments involving related algorithms [77].

## 7.4 Closing Remarks

In order to place our results in perspective, let us discuss their relation to previous research. We are not the first to consider variants of temporal–difference learning that approximate differential value functions. However, the algorithms that have been studied in this context generally make use of look–up table representations, which involve storing and updating one value per state in the state space. We refer the reader to [65, 67, 48, 1] for work along these lines. The computational experiments of Tadepalli and Ok [77] and Marbach, Mihatsch, and Tsitsiklis [49] do employ parameterized approximations to the differential value function, but the authors do not provide convergence analyses.

It is known that the differential value function of a finite–state infinite–horizon Markov chain is the same as the value function of an auxiliary absorbing Markov chain [8, 13]. This relationship motivates one way of using temporal–difference learning to approximate a differential value function, namely, deriving the auxiliary absorbing Markov chain and then employing an existing version of temporal–difference learning. However, this reduction can affect approximations in undesirable ways, as we discuss next.

In temporal–difference learning, each weight update is dependent on a history of visited states. When temporal–difference learning is applied to an absorbing Markov chain, multiple finite trajectories (each terminating at an absorbing state) are simulated. Weight updates occur during these simulations, and the history of visited states is erased upon the termination of each trajectory. Even though restarting the record of visited states is appropriate for an absorbing Markov chain, it is unnatural for the original infinite horizon Markov chain. Due to this peculiarity introduced by the reduction, it is preferable to use a variant of temporal–difference learning designed specifically for approximating differential value functions, as the one we have introduced.

The error bound we have derived (Theorem 7.6(5)) is very similar in appearance to that pertaining to discounted reward temporal–difference learning (Theorem 4.9(4)). The only difference is the presence of a parameter $\beta$ that substitutes for the role of the discount factor $\alpha$. This parameter represents a "mixing factor," related to the mixing time of the Markov process. It is interesting to note that even if a given Markov chain takes a long time to reach steady state, which would imply that $\beta$ is close to 1, the contraction factor $\kappa$ of $\overline{\Pi}T^{(\lambda)}$ may be small due to the choice of basis functions. This may partially explain why small values of $\lambda$ seem to lead to good approximations even with Markov chains that converge to steady state rather slowly.

Our comparison of average reward and discounted reward temporal–difference learning algorithms has lead to an interesting conclusion – the asymptotic results are close (if the discount factor is close to 1), but the evolution of iterates during the course of the algorithms can be very different. In particular, the weight updates in average reward temporal–difference learning avoid some "noise" that influences the discounted reward counterpart.

On the technical side, we mention a few straightforward extensions to our results.

1. If we allow the reward per stage $g(x_t)$ to be dependent on the next state (i.e., employ a function $g(x_t, x_{t+1})$) or even to be noisy, as opposed to being a deterministic function of $x_t$ and $x_{t+1}$, our line of analysis still goes through. In particular, we can replace the Markov process $y_t = (x_t, x_{t+1}, z_t)$ that was constructed for the purposes of our analysis with a process $y_t = (x_t, x_{t+1}, z_t, g_t)$, where $g_t$ is the reward associated with the transition from $x_t$ to $x_{t+1}$. Then, as long as the distribution of the noise only depends on the current state, our proof can easily be modified to accommodate this situation.

2. The assumption that the step sizes $\eta_t$ are equal to $C\gamma_t$ was adopted for convenience, and weaker assumptions that allow greater flexibility in choosing step sizes $\eta_t$ will certainly suffice, although this might require a substantially more sophisticated proof.

3. If the basis functions $\phi, \ldots, \phi_K$ were allowed to contain $e$ within their span, then our line of analysis can be used to show that $\overline{\Phi}r_t$ still converges, but $\Phi r_t - \overline{\Phi}r_t$ is aligned to $e$ and need not converge.

4. The algorithm we analyzed simultaneously adapts approximations of average reward and the differential value function. In [85], the same line of analysis is used to establish convergence and error bounds for a related algorithm in which the average reward estimate $\mu$ is held fixed while the weights $r(1), \ldots, r(K)$ are updated as usual. The error bound in that case includes an extra term that is proportional to $\|\Pi e\|_\pi$ times the the error $|\mu - \mu^*|$ in the average reward estimate. It is interesting to note that $\|\Pi e\|_\pi$ is influenced by the orientation of the basis functions.

# Chapter 8

# Approximations Based on Representative Scenarios

There are two preconditions to effective value function approximation: an appropriate parameterization and an algorithm for computing parameters. In previous chapters, we have focused on algorithms that adjust basis function weights in a linear parameterization. Our discussion on approaches to selecting basis functions, however, has been limited to the context of a case study (Chapter 6), where the selection was based on intuition concerning the nature of the decision problem.

An interesting question concerns whether or not there are systematic and broadly applicable methods and/or guidelines for basis function selection. In this chapter, we explore one approach to generating basis functions that involves the use of "representative scenarios." In particular, rewards generated under policies and disturbance sequences from a select set are employed as basis functions. This notion is introduced in the context of a controlled system. Certain relevant analytical results pertaining to the autonomous case are then presented in Section 8.2. As an illustration, Section 8.3 provides a concrete application of the main result. Future research directions and related ideas in the literature are discussed in a closing section.

## 8.1 Generation of Basis Functions from Scenarios

As in Section 2.1, we consider a discrete–time dynamic system that, at each time $t$, takes on a state $x_t$ and evolves according to

$$x_{t+1} = f(x_t, u_t, w_t),$$

where $w_t$ is a disturbance and $u_t$ is a control decision. Though more general state spaces will be treated in the next section, we restrict attention for now to finite state, disturbance, and control spaces, denoted by $S$, $W$, and $U$, respectively. For each policy $\mu$, a value function $J^\mu : S \mapsto \Re$ is defined by

$$J^\mu(x) = \mathrm{E}\left[\sum_{t=0}^{\infty} \alpha^t g(x_t, \mu(x_t)) \Big| x_0 = x\right],$$

where $g : S \times U \mapsto \Re$ is a reward function, $\alpha \in [0, 1)$ is a discount factor and the state sequence is generated according to $x_0 = x$ and $x_{t+1} = f(x_t, \mu(x_t), w_t)$. The optimal value function $J^*$ is defined by

$$J^*(x) = \max_{\mu} J^{\mu}(x).$$

Let the process $\{w_t | t = 0, 1, 2, \ldots\}$ be defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$, so that each random variable $w_t$ is implicitly a function of a sample point $\omega$ taking on values in $\Omega$. We will refer to each $\omega \in \Omega$ as a "scenario." Intuitively, a scenario captures all relevant information about future events. For each policy $\mu$ and scenario $\omega$, we define a value function $J^{\mu,\omega} : S \mapsto \Re$ by

$$J^{\mu,\omega}(x) = \sum_{t=0}^{\infty} \alpha^t g(x_t, \mu(x_t)),$$

where $x_0 = x$ and $x_{t+1} = f(x_t, \mu(x_t), w_t)$. This function provides the future reward starting at state $x$ given that the policy $\mu$ is deployed and the scenario $\omega$ determines future disturbances. Note that

$$J^{\mu}(x) = \mathrm{E}\left[J^{\mu,\omega}(x)\right].$$

Suppose we believe that a set $(\mu_1, \omega_1), \ldots, (\mu_K, \omega_K)$ of policy–scenario pairs is "sufficiently representative" of the range of possibilities. One approach proposed by Bertsekas [10] for approximating the optimal value function $J^*$ involves the use of basis functions $\phi_k = J^{\mu_k,\omega_k}$ for $k = 1, \ldots, K$. Temporal–difference learning can then be applied to compute corresponding weights $r(1), \ldots, r(K)$. This approach essentially reduces the problem of basis function selection to one of scenario selection.

## 8.2 The Case of an Autonomous System

As a first step in further understanding issues involved in the use of scenarios, in this section, we develop some theory pertaining to the context of autonomous systems. The intention is to gain an understanding of how many scenarios it takes to constitute a "sufficiently representative" set and how difficult it is to find such scenarios.

We consider a stochastic dynamic system defined with respect to a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ that evolves in a state space $\Re^d$ according to

$$x_{t+1} = f(x_t, w_t),$$

where each disturbance $w_t$ is implicitly a function of a random scenario $\omega$ and takes on values in a measurable space $(W, \mathcal{W})$. (As in previous chapters, we implicitly assume that relevant functions such as $f$ are measurable.)

Let $g : \Re^d \mapsto \Re$ be a reward function, and let $\alpha \in [0, 1)$ be a discount factor. For

each scenario $\omega \in \Omega$, we define a function

$$J^{\omega}(x) = \sum_{t=0}^{\infty} \alpha^t g(x_t),$$

where $x_0 = x$ and $x_{t+1} = f(x_t, w_t)$. Note that the value function for this autonomous system is given by

$$J^*(x) = \mathrm{E}\left[J^{\omega}(x)\right].$$

We consider approximations of the form

$$\tilde{J}(x, r) = \sum_{k=1}^{K} r(k) J^{\omega_k}(x),$$

where $\omega_1, \ldots, \omega_K$ make up a set of representative scenarios and $r \in \Re^K$ is a weight vector. The hope is that, for many problems of practical interest, a relatively small set of scenarios is representative enough to generate an accurate approximation to the value function. In this spirit, we will derive bounds on the number $K$ of scenarios that would be sufficient for an approximation of some desired level of accuracy. We will consider two mechanisms for scenario selection:

1. Random sampling according to the distribution $\mathcal{P}$.
2. "Best–case" sampling.

The bounds will be contingent on properties of the system function $f$ and the reward function $g$.

### 8.2.1 Bounds on Numbers of Scenarios

Before providing bounds, let us state assumptions and define relevant terms. Let $\mathbf{d}$ be a pseudo–metric on $\Re^d$ (i.e., a nonnegative scalar function on $\Re^d \times \Re^d$ such that $\mathbf{d}(x, x) = 0$, $\mathbf{d}(x, y) = \mathbf{d}(y, x)$, and $\mathbf{d}(x, z) \leq \mathbf{d}(x, y) + \mathbf{d}(y, z)$ for all $x, y, z$). We make the following assumption concerning the system and reward functions.

**Assumption 8.1** *The function $g$ and each component function $f_i$ take on values in $[-M/2, M/2]$. Let the following conditions hold for the pseudo–metric $\mathbf{d}$.*

1. *There exists a scalar $\beta \in [0, \frac{1}{\alpha})$ such that*

$$\mathbf{d}(f(x, w), f(y, w)) \leq \beta \mathbf{d}(x, y),$$

*for all $w \in W$ and any $x, y \in \Re^d$.*

2. *There exists a scalar $L \in \Re$ such that*

$$|g(x) - g(y)| \leq L\mathbf{d}(x, y),$$

*for all $x, y \in \Re^d$.*

3. *There exists a positive scalar $\chi$ and a vector $\theta$ in the $d$–dimensional unit simplex such that*

$$\mathbf{d}(x, y) \leq \chi \sum_{i=1}^{d} \theta_i |x_i - y_i|,$$

*for all $x, y \in \Re^d$.*

Condition (1) can be viewed as a "stability requirement." Among other things, it implies that, given a particular disturbance sequence, state trajectories with different initial states do not diverge from one another at a rate greater than $1/\alpha^t$. The second condition is a Lipschitz bound that requires a degree of "smoothness" from the reward function. The final condition relates the pseudo–metric $\mathbf{d}$ to a weighted Manhattan norm.

Our bounds on numbers of scenarios will be based on the "complexity" of the system function. We now define the notion of complexity that will be employed.

**Definition 8.2** *Let $(\Omega, \mathcal{F})$ be a measurable space, and let $\mathcal{H}$ be a set of measurable functions mapping $\Omega$ to $\Re$. A set $\{\omega_1, \ldots, \omega_K\}$ is said to be* **shattered** *by $\mathcal{H}$ if there exists a vector $a \in \Re^K$ such that, for every binary vector $b \in \{0, 1\}^K$, there exists a function $h \in \mathcal{H}$ such that*

$$h(\omega_k) \begin{cases} \geq a_k, & \text{if } b_k = 1, \\ < a_k, & \text{if } b_k = 0. \end{cases}$$

*The* **pseudo–dimension** *of $\mathcal{H}$, denoted by $\mathbf{dim}(\mathcal{H})$, is defined as the largest integer $d$ such that there exists a set of cardinality $d$ that is shattered by $\mathcal{H}$.*

This notion is due to Pollard [58]. It is a generalization of the Vapnik–Chervonenkis dimension [88], and both notions of complexity have been applied extensively in the machine learning literature (see, e.g., [36] or [89]).

We are now ready to state the main result of this chapter. This result provides a sufficient condition for the approximation of the value function within a "tolerance" $\epsilon$ with "confidence" $1 - \delta$ in the event that the samples are drawn as independent random variables each distributed according to $\mathcal{P}$.

**Theorem 8.3** *Let Assumption 8.1 hold. Let $\beta$, $L$, $\chi$, and $\theta$ be variables satisfying the conditions of the assumption. For each $i \in \{1, \ldots, d\}$ let $\mathcal{H}_i = \{f_i(x, \cdot) | x \in S\}$. Let $\epsilon$ be in $(0, M/2]$, let $\delta$ be a positive scalar, and let $K$ be an integer satisfying*

$$K \geq \frac{32M^2}{\epsilon^2} \left( \ln \frac{4}{\delta} + \left( \sum_{i=1}^{n} \mathbf{dim}(\mathcal{H}_i) \right) \ln \left( \frac{16\alpha\chi LMe}{(1 - \alpha\beta)\epsilon} \ln \left( \frac{16\alpha\chi LMe}{(1 - \alpha\beta)\epsilon} \right) \right) \right).$$

*Then, given a set of independent samples $\omega_1, \ldots, \omega_K \in \Omega$ each drawn according to $\mathcal{P}$, we have*

$$\text{Prob} \left\{ \sup_{x \in S} \left| J^*(x) - \frac{1}{K} \sum_{k=1}^{K} J^{\omega_k}(x) \right| \leq \epsilon \right\} \geq 1 - \delta.$$

Note that, as presented in the theorem, each function $J^{\omega_k}$ receives an equal weighting of $r(k) = 1/K$. If the weights were allowed to be optimized in some way, the sample complexity requirements might decrease.

Let us mention two qualitative observations regarding Theorem 8.3:

1. Given a system, the sample complexity grows at most as a polynomial in $1/\epsilon$ and $\ln(1/\delta)$.

2. Given a class of systems and a particular $\epsilon, \delta > 0$, the sample complexity grows at most as a polynomial in the pseudo–dimensions $\mathbf{dim}(\mathcal{H}_i)$, the constants $\chi$ and $L$, the range $M$, and an "effective horizon" $1/(1-\alpha\beta)$.

The first observation simply states that the sample complexity does not grow at an unreasonable rate as greater accuracy is desired. The second observation provides a new perspective on the "complexity" of systems. In particular, one may naturally be inclined to associate the complexity of a system with the size of its state space. This might lead to a premonition that the number of scenarios required to summarize future possibilities for all states is a function of state space size, which is typically intractable. However, the result points out that it is instead the pseudo–dimensions associated with components of the system function that influence sample requirements. This is an important observation because, for many relevant classes of systems, the pseudo–dimension may grow at a tractable rate relative to the state space size.

A second case we consider involves a "best–case" choice of scenarios. In particular, we assume that the scenarios are chosen in a way that minimizes approximation error. The corresponding result is a corollary of Theorem 8.3.

**Corollary 8.4** *Let Assumption 8.1 hold. Let $\beta$, $L$, $\chi$, and $\theta$ be variables satisfying the conditions of the assumption. For each $i \in \{1, \ldots, d\}$ let $\mathcal{H}_i = \{f_i(x, \cdot) | x \in S\}$. Let $\epsilon$ be in $(0, M/2]$, let $\delta$ be positive scalars, and let $K$ be an integer satisfying*

$$K \geq \frac{32M^2}{\epsilon^2}\left(\ln 4 + \left(\sum_{i=1}^{n} \mathbf{dim}(\mathcal{H}_i)\right) \ln\left(\frac{16\alpha\chi LMe}{(1-\alpha\beta)\epsilon} \ln\left(\frac{16\alpha\chi LMe}{(1-\alpha\beta)\epsilon}\right)\right)\right).$$

*We then have,*

$$\inf_{\omega_1,\ldots,\omega_K \in \Omega} \sup_{x \in S} \left| J^*(x) - \frac{1}{K}\sum_{k=1}^{K} J^{\omega_k}(x) \right| \leq \epsilon.$$

In practice, the sample complexity associated with this case may be substantially lower than in the case of randomly sampled scenarios. This possibility is reflected in the bound by the elimination of the $1/\delta$ term. However, the bound does not reflect any dramatic change in the rate of growth of sample complexity with system parameters (pseudo–dimensions etc.). This may be because the ability to select key scenarios indeed does not grant such benefits, or it may simply be due to the fact that the bound is not sharp.

### 8.2.2 Preliminaries

In this section, we introduce some results from the literature on uniform laws of large numbers that will be used in our analysis. These results are based on the work of Pollard [58], but as an accessible source for the particular bounds we present, we refer the reader to Theorem 5.7 and Corollary 4.2 of the book by Vidyasagar [89]. We begin by defining some terms and notation.

**Definition 8.5** *Let $(Y, \rho)$ be a pseudo–metric space. Given a set $Z \subseteq Y$ and some $\epsilon > 0$, a set $\{y_1, \ldots, y_m\} \subseteq Y$ is said to be an* $\epsilon$**–cover** *of $Z$ if, for each $z \in Z$, there exists and index $i$ such that $\rho(z, y_i) \leq \epsilon$. The* $\epsilon$**–covering number** *of $Z$ (with respect to $\rho$) is defined as the smallest integer $m$ such that $Z$ has an $\epsilon$–cover of cardinality $m$, and is denoted by $\mathbf{N}(Z, \epsilon, \rho)$.*

One pseudo–metric that we will be working with is defined, for a given probability space $(\Omega, \mathcal{F}, \mathcal{P})$, by

$$\rho_{\mathcal{P}}(h_1, h_2) = \int |h_1(\omega) - h_2(\omega)| \mathcal{P}(d\omega),$$

for any measurable scalar functions $h_1$ and $h_2$. The following theorem presents a uniform law of large numbers.

**Theorem 8.6** *Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space and let $h : \Re^n \times \Omega \mapsto [0, 1]$ be a measurable function (with respect to the product of the Borel sets and $\mathcal{F}$). Let $\omega_1, \ldots, \omega_K$ be a sequence of $K$ independent samples drawn according to $\mathcal{P}$. Then,*

$$\text{Prob}\left\{ \sup_{x \in \Re^n} \left| \frac{1}{K} \sum_{k=1}^{K} h(x, \omega_k) - \int h(x, \omega) \mathcal{P}(d\omega) \right| > \epsilon \right\} \leq 2 \sup_{\mathcal{Q}} \mathbf{N}\left(\mathcal{H}, \epsilon/8, \rho_{\mathcal{Q}}\right) e^{-\epsilon^2 K/32},$$

*where the supremum is over all probability measures on $(\Omega, \mathcal{F})$.*

Covering numbers are somewhat unwieldy because they are contingent on a level of tolerance $\epsilon$ and a probability distribution $\mathcal{P}$. It is often more convenient to deal with the pseudo–dimension, which is a single number that can be used to bound covering numbers as illustrated by the next theorem.

**Theorem 8.7** *For any probability space $(\Omega, \mathcal{F}, \mathcal{P})$, any set $\mathcal{H}$ of measurable functions mapping $\Omega$ to $[0, 1]$, and any $\epsilon \in (0, 1/2]$, we have*

$$\mathbf{N}(\mathcal{H}, \epsilon, \rho_{\mathcal{P}}) \leq 2 \left( \frac{2e}{\epsilon} \ln \frac{2e}{\epsilon} \right)^{\mathbf{dim}(\mathcal{H})}.$$

Equipped with the tools of this section, we are ready to prove Theorem 8.3.

### 8.2.3 Proof of Theorem 8.3

We generalize to vector–valued functions the pseudo–metric from the previous subsection by letting

$$\rho_{\mathcal{P},\theta}(h,\overline{h}) = \int \sum_{i=1}^{d} \theta_i |h_i(\omega) - \overline{h}_i(\omega)| \mathcal{P}(d\omega),$$

for any probability space $(\Omega, \mathcal{F}, \mathcal{P})$, vector $\theta$ in the unit simplex, and measurable vector–valued functions $h$ and $\overline{h}$.

Let $\mathcal{H} = \{f(x,\cdot)|x \in S\}$. We will place a bound on the covering number of this set with respect to $\rho_{\mathcal{P},\theta}$. Fix an $\epsilon > 0$. For each set $\mathcal{H}_i$, let the $m_i = \mathbf{N}(\mathcal{H}_i, \epsilon, \rho_{\mathcal{P}})$. Then, for each $i$, there exists a set of functions $\overline{\mathcal{H}}_i$ consisting of $m_i$ functions such that for any $h \in \mathcal{H}_i$, there exists a function $\overline{h} \in \overline{\mathcal{H}}_i$ with $\rho_{\mathcal{P}}(h,\overline{h}) < \epsilon$. Let $\overline{\mathcal{H}} = \prod_{i=1}^{n} \overline{\mathcal{H}}_i$. Then, for each $h \in \mathcal{H}$ there exists some $\overline{h} \in \overline{\mathcal{H}}$ with component functions satisfying $\rho_{\mathcal{P}}(h_i,\overline{h}_i) \leq \epsilon$. Since $\theta$ is in the unit simplex, it follows that $\rho_{\mathcal{P},\theta}(h,\overline{h}) \leq \epsilon$. Hence, $\overline{\mathcal{H}}$ is an $\epsilon$–cover of $\mathcal{H}$, and therefore

$$\mathbf{N}\left(\mathcal{H}, \epsilon, \rho_{\mathcal{P},\theta}\right) \leq \prod_{i=1}^{n} \mathbf{N}\left(\mathcal{H}_i, \epsilon, \rho_{\mathcal{P}}\right).$$

A bound can also be placed on the covering number of $\mathcal{H}$ with respect to the pseudo–metric $\mathbf{d}$. In particular, it follows from Assumption 8.1(3) that $\overline{\mathcal{H}}$ is a $(\chi\epsilon)$–cover of $\mathcal{H}$, and therefore

$$\mathbf{N}\left(\mathcal{H}, \chi\epsilon, \mathbf{d}\right) \leq \mathbf{N}\left(\mathcal{H}, \epsilon, \rho_{\mathcal{P},\theta}\right) \leq \prod_{i=1}^{n} \mathbf{N}\left(\mathcal{H}_i, \epsilon, \rho_{\mathcal{P}}\right).$$

Let $\mathcal{J} = \{J^{(\cdot)}(x) - g(x)|x \in S\}$. We will now place a bound on the covering numbers of this set with respect to $\rho_{\mathcal{P}}$. Using the same $\epsilon > 0$ and $\overline{\mathcal{H}}$ as above, let us define a set $\Psi$ of scalar functions that contains, for each function $\overline{h} \in \overline{\mathcal{H}}$, a function of the form

$$\psi(\omega) = \sum_{t=1}^{\infty} \alpha^t g(y_t),$$

where the sequence $y_1, y_2, \ldots$ is defined by $y_1 = \overline{h}(\omega)$ and $y_{t+1} = f(y_t, w_t)$. Let $\eta$ be a function in $\mathcal{J}$ and let $x_\eta \in S$ be a state for which $\eta(\omega) = J^{\omega}(x_\eta) - g(x_\eta)$. Let $\overline{h}$ be a function in $\overline{\mathcal{H}}$ satisfying $\mathbf{d}(f(x_\eta,\cdot),\overline{h}) \leq \chi\epsilon$, and let $\psi$ be the corresponding element of $\Psi$. We then have

$$\rho_{\mathcal{P}}(\eta,\psi) = \int |\eta(\omega) - \psi(\omega)| \mathcal{P}(d\omega) \leq \int \left( \sum_{t=1}^{\infty} \alpha^t |g(x_t) - g(y_t)| \right) \mathcal{P}(d\omega).$$

For each $t$th term of the summation, by Assumption 8.1(1)–(2),

$$|g(x_t) - g(y_t)| \leq L\mathbf{d}(x_t, y_t) \leq L\beta^{t-1}\mathbf{d}(x_1, y_1) = L\beta^{t-1}\mathbf{d}(f(x_\eta, w_0), \overline{h}(\omega)).$$

It follows that

$$\begin{aligned}\rho_{\mathcal{P}}(\eta,\psi) &\leq L\sum_{t=1}^{\infty}\alpha^t\beta^{t-1}\int \mathbf{d}(f(x,w_0),\overline{h}(\omega))\mathcal{P}(d\omega)\\ &\leq L\sum_{t=1}^{\infty}\alpha^t\beta^{t-1}\chi\epsilon\\ &= \frac{\alpha L\chi\epsilon}{(1-\alpha\beta)}.\end{aligned}$$

Since $|\Psi| = |\overline{\mathcal{H}}|$,

$$\mathbf{N}\left(\mathcal{J}, \frac{\alpha L\chi\epsilon}{(1-\alpha\beta)}, \rho_{\mathcal{P},\theta}\right) \leq \mathbf{N}\left(\mathcal{H}, \epsilon, \rho_{\mathcal{P},\theta}\right) \leq \prod_{i=1}^{n}\mathbf{N}\left(\mathcal{H}_i, \epsilon, \rho_{\mathcal{P}}\right).$$

By Theorem 8.7, for $\epsilon \in (0, M/2]$

$$\mathbf{N}(\mathcal{H}_i, \epsilon, \rho_{\mathcal{P}}) \leq 2\left(\frac{2Me}{\epsilon}\ln\frac{2eM}{\epsilon}\right)^{\mathbf{dim}(\mathcal{H}_i)}.$$

(The tolerance $\epsilon$ is divided by $M$ because the range of the functions of interest is now $[-M/2, M/2]$ instead of $[0,1]$.) Combining this with the bound on the covering number of $\mathcal{J}$, we obtain

$$\mathbf{N}(\mathcal{J}, \epsilon, \rho_{\mathcal{P}}) \leq 2\left(\frac{2\alpha\chi LMe}{(1-\alpha\beta)\epsilon}\ln\left(\frac{2\alpha\chi LMe}{(1-\alpha\beta)\epsilon}\right)\right)^{\sum_{i=1}^{n}\mathbf{dim}(\mathcal{H}_i)}.$$

Since this expression is valid for any probability measure $\mathcal{P}$, by Theorem 8.6, we have

$$\begin{aligned}&\mathrm{Prob}\left\{\sup_{x\in S}\left|J^*(x) - \frac{1}{K}\sum_{k=1}^{K}J^{\omega_k}(x)\right| > \epsilon\right\}\\ &\qquad \leq 4\left(\frac{16\alpha\chi LMe}{(1-\alpha\beta)\epsilon}\ln\left(\frac{16\alpha\chi LMe}{(1-\alpha\beta)\epsilon}\right)\right)^{\sum_{i=1}^{n}\mathbf{dim}(\mathcal{H}_i)} e^{-\epsilon^2K/32M^2}.\end{aligned}$$

Some simple algebra then leads to the fact that

$$\mathrm{Prob}\left\{\sup_{x\in S}\left|J^*(x) - \frac{1}{K}\sum_{k=1}^{K}J^{\omega_k}(x)\right| > \epsilon\right\} \leq \delta,$$

for any number of samples

$$K \geq \frac{32M^2}{\epsilon^2}\left(\ln\frac{4}{\delta} + \left(\sum_{i=1}^{n}\mathbf{dim}(\mathcal{H}_i)\right)\ln\left(\frac{16\alpha\chi LMe}{(1-\alpha\beta)\epsilon}\ln\left(\frac{16\alpha\chi LMe}{(1-\alpha\beta)\epsilon}\right)\right)\right).$$

**Q.E.D.**

### 8.2.4 Proof of Corollary 8.4

This corollary follows almost immediately from the theorem. In particular, note that, for each $\delta < 1$, there is positive probability that the random sampling of $K$ scenarios with

$$K \geq \frac{32M^2}{\epsilon^2}\left(\ln\frac{4}{\delta} + \left(\sum_{i=1}^{n}\mathbf{dim}(\mathcal{H}_i)\right)\ln\left(\frac{16\alpha\chi LMe}{(1-\alpha\beta)\epsilon}\ln\left(\frac{16\alpha\chi LMe}{(1-\alpha\beta)\epsilon}\right)\right)\right),$$

will result in positive probability for the event that

$$\sup_{x\in S}\left|J^*(x) - \frac{1}{K}\sum_{k=1}^{K}J^{\omega_k}(x)\right| \leq \epsilon.$$

Hence, for each $\delta < 1$ there exists at least one set of $K$ scenarios that generates the above event. By taking the limit as $\delta$ approaches 1, we arrive at the corollary. **Q.E.D.**

## 8.3 An Example

To enhance our understanding of Theorem 8.3, let us discuss a concrete example. Consider a stable linear system evolving in $\Re^d$ according to

$$x_{t+1} = \beta x_t + w_t,$$

where $\beta$ is a scalar in $[0, 1)$ and each disturbance $w_t$ takes on values in $\Re^d$ with Euclidean norm $\|w_t\|_2 \leq 1$. Note that, if $\|x_0\|_2 \leq 1/(1-\beta)$ then $\|x_t\|_2 \leq 1/(1-\beta)$ for all $t$. Hence, we can effectively think of the bounded set $S = \{x \in \Re | \|x\|_2 \leq 1/(1-\beta)\}$ as the state space.

The system function is given by $f(x, w) = \beta x + w$, so each set $\mathcal{H}_i = \{f_i(x, \cdot) | x \in S\}$ is a subset of a one–dimensional linear space. It is well known that the pseudo–dimension of a linear space of functions is equal to the linear dimension [26], and therefore, $\mathbf{dim}(\mathcal{H}_i) = 1$ for each $i = 1, \ldots, d$.

Letting $\mathbf{d}$ be the metric defined by the Euclidean norm on $\Re^d$, it is easy to see that parts (1) and (3) of Assumption 8.1 are satisfied with the value of $\beta$ from the definition of the system function, $M = 2/(1-\beta)$, $\theta_i = 1/d$ for $i = 1, \ldots, d$, and $\chi = \sqrt{d}$. Let us assume that the reward function $g$ is also bounded by $M$ and satisfies Assumption 8.1(2) for some scalar $L$. Then, the bound of Theorem 8.3 applies, and we have

$$\text{Prob}\left\{\sup_{x\in S}\left|J^*(x) - \frac{1}{K}\sum_{k=1}^{K}J^{\omega_k}(x)\right| \leq \epsilon\right\} \geq 1 - \delta,$$

for a set of scenarios of size

$$K \geq \frac{32M^2}{\epsilon^2}\left(\ln\frac{4}{\delta} + d\ln\left(\frac{16\alpha\sqrt{d}Le}{(1-\alpha\beta)(1-\beta)\epsilon}\ln\left(\frac{16\alpha\sqrt{d}Le}{(1-\alpha\beta)(1-\beta)\epsilon}\right)\right)\right).$$

Note that, for this class of systems, the sample complexity is bounded by a polynomial in the state space dimension, which unlike the state space size, is tractable.

## 8.4 Closing Remarks

Our analysis has focused on the case of autonomous systems, where we have derived a bound on the number of scenarios that leads to accurate approximation. This bound bears a dependence on pseudo–dimensions of components of the system function, which are measures of complexity that can grow at a rate much slower than the size of the state space.

The use of representative scenarios in the context of controlled systems presents an interesting and challenging direction for future research. In particular, an open question concerns the possibility of developing a measure of complexity for controlled systems that both grows slower than state space size and provides a bound on sufficient numbers of scenarios.

It is worth noting that Theorem 8.3 is also relevant to the study of "rollout algorithms" [78, 13, 11]. Such algorithms aim at approximating values $J^{\mu}(x)$ under a given policy $\mu$ by averaging results from simulations. In particular, when the value $J^{\mu}(x)$ associated with a state $x$ is desired, the rollout algorithm generates an approximation $(1/K)\sum_{k=1}^{K} J^{\mu,\omega_k}(x)$, where $\omega_1, \ldots, \omega_K$ are a set of scenarios produced via simulation. One may consider the possibility of fixing the set of scenarios, storing them in memory, and reusing them to compute $J^{\mu}(x)$ whenever the value at a particular state $x$ is desired. In this event, Theorem 8.3 provides a bound on the number of such scenarios needed to obtain uniformly low error with a certain degree of confidence.

One advantage to reusing scenarios in rollout algorithms rather than generating new ones for each state might be computational efficiency. There is another potential advantage, however, if uniformly low error is desired. In particular, if new scenarios are used at each state in a large state space, it is very likely that the error associated with approximations at some of the states will be very large. On the other hand, if the same scenarios are reused, there is a probability $1 - \delta$ that the approximation is accurate at every single state. This issue is related to the use of "common random numbers" for variance reduction in simulation, as studied, for example, in [29].

# Chapter 9

# Perspectives and Prospects

There is a lack of systematic approaches for dealing with the myriad of complex stochastic control problems arising in practical applications. Theoretical results (e.g., [21, 55]) suggest that such problems are fundamentally intractable. In particular, we believe that there is no general, fully–automated, and computationally efficient method that can address the range of stochastic control problems we have in mind.

The thrust of our effort has been in developing a methodology that, though not fully automated, may offer a vehicle for tackling many problems of interest. To bring the thesis to a close, let us attempt to place in perspective our philosophy pertaining to the nature of this methodology and to discuss what we envisage as potential prospects.

As discussed early in the thesis, the "curse of dimensionality" can be viewed as the primary obstacle prohibiting effective solution methods for stochastic control problems. It is interesting to note that an analogous impediment arises in statistical regression. In particular, given an ability to collect data pairs of the form $(x, J(x))$, the problem of producing an accurate approximation $\tilde{J}$ to the underlying function $J$ becomes computationally intractable as the dimension of the domain increases. Similarly with the context of stochastic control, difficulties arise due to the curse of dimensionality. In the setting of statistical regression, a common approach to dealing with this limitation involves selecting a set of basis functions $\phi_1, \ldots, \phi_K$, collecting a set of input–output pairs $\{(x_1, J(x_1)), \ldots, (x_m, J(x_m))\}$, and using the least–squares algorithm to compute weights $r(1), \ldots, r(K)$ that minimize

$$\sum_{i=1}^{m} \left( J(x_i) - \sum_{k=1}^{K} r(k)\phi_k(x_i) \right)^2 .$$

The result is an approximation of the form

$$\tilde{J}(x) = \sum_{k=1}^{K} r(k)\phi_k(x).$$

Though there is no systematic and generally applicable method for choosing basis functions, a combination of intuition, analysis, guesswork, and experimentation often leads to a useful selection. In fact, the combination of basis function selection and

least–squares is a valuable tool that has met prevalent application.

The utility of least–squares statistical regression provides inspiration for the flavor of methods we study. In particular, temporal–difference learning can be viewed as an analog to the least–squares algorithm that is applicable to stochastic control rather than statistical regression – given a stochastic control problem and a selection of basis functions $\phi_1, \ldots, \phi_K$, the intent is to compute weights $r(1), \ldots, r(K)$ such that the function

$$\tilde{J}(x) = \sum_{k=1}^{K} r(k)\phi_k(x)$$

approximates the value function. In special cases involving autonomous systems and optimal stopping problems, we have provided analyses that ensure desirable qualities for resulting approximations. In these settings, the streamlined character of the algorithms and results makes them accessible and useful as demonstrated in the computational study of Chapter 6.

Though our work provides a starting point, the development of streamlined methods and analyses for general classes of stochastic control problems remains largely open. Our hope, however, is that the range of problems we can address in such a manner will broaden with future research. A goal might be to eventually produce an algorithm that is as useful and widely accessible in the context of stochastic control as is least–squares in the context of statistical regression.

An additional area of research pursued in this thesis involves an indirect approach to basis function selection via use of "representative scenarios." There are many open problems in the context of this approach, and the study of basis function selection for stochastic control problems in general poses a broad and interesting topic for future research.

# Bibliography


[1] J. Abounadi. *Stochastic Approximation for Non–Expansive Maps: Application to Q–Learning Algorithms.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1998.

[2] M. Adams and V. Guillemin. *Measure Theory and Probability.* Birkhäuser, Boston, MA, 1996.

[3] L. C. Baird. Residual Algorithms: Reinforcement Learning with Function Approximation. In Prieditis and Russell, editors, *Machine Learning: Proceedings of the Twelfth International Conference*, San Francisco, CA, 1995. Morgan Kaufman.

[4] J. Barraquand and D. Martineau. Numerical Valuation of High Dimensional Multivariate American Securities. *Journal of Financial and Quantitative Analysis*, 30(3):383–405, 1997.

[5] A. G. Barto, S. J. Bradtke, and S. P. Singh. Real–Time Learning and Control Using Asynchronous Dynamic Programming. *Artificial Intelligence*, 72:81–138, 1995.

[6] A. G. Barto and R. S. Sutton. Simulation of Anticipatory Responses in Classical Conditioning by a Neuron–Like Adaptive Element. *Behavioural Brain Research*, 4:221–235, 1982.

[7] A. Benveniste, M. Métivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations.* Springer–Verlag, Berlin, 1990.

[8] D. P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, Bellmont, MA, 1995.

[9] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Bellmont, MA, 1995.

[10] D. P. Bertsekas, 1997. Private communication.

[11] D. P. Bertsekas and D. A. Castanon. Rollout Algorithms for Stochastic Scheduling Problems. preprint, 1998.

[12] D. P. Bertsekas and S. Ioffe. Temporal Differences–Based Policy Iteration and Applications in Neuro–Dynamic Programming. Technical Report LIDS–P–2349, MIT Laboratory for Information and Decision Systems, 1996.

[13] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Bellmont, MA, 1996.

[14] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81:637–654, 1973.

[15] V. S. Borkar. *Probability Theory: An Advanced Course.* Springer–Verlag, New York, NY, 1995.

[16] J. A. Boyan and A. W. Moore. Generalization in Reinforcement Learning: Safely Approximating the Value Function. In *Advances in Neural Information Processing Systems 7*, Cambridge, MA, 1995. MIT Press.

[17] M. J. Brennan, E. S. Shwartz, and R. Lagnado. Strategic Asset Allocation. *Journal of Economic Dynamics and Control*, 21:1377–1403, 1997.

[18] M. Broadie and P. Glasserman. A Stochastic Mesh Method for Pricing High–Dimensional American Options. working paper, 1997.

[19] M. Broadie and P. Glasserman. Pricing American–Style Securities Using Simulation. *Journal of Economic Dynamics and Control*, 21:1323–1352, 1997.

[20] E. K. P. Chong and P. J. Ramadge. Stochastic Optimization of Regenerative Systems Using Infinitesimal Perturbation Analysis. *IEEE Transactions on Automatic Control*, 39:1400–1410, 1994.

[21] C.-S. Chow and J. N. Tsitsiklis. The Complexity of Dynamic Programming. *Journal of Complexity*, 5:466–488, 1989.

[22] R. H. Crites. *Large–Scale Dynamic Optimization Using Teams of Reinforcement Learning Agents.* PhD thesis, University of Massachusetts, Amherst, MA, 1996.

[23] R. H. Crites and A. G. Barto. Improving Elevator Performance Using Reinforcement Learning. In *Advances in Neural Information Processing Systems 8*, Cambridge, MA, 1995. MIT Press.

[24] P. D. Dayan. The Convergence of TD($\lambda$) for General $\lambda$. *Machine Learning*, 8:341–362, 1992.

[25] P. D. Dayan and T. J. Sejnowski. TD($\lambda$) Converges with Probability 1. *Machine Learning*, 14:295–301, 1994.

[26] R. M. Dudley. Central Limit Theorems for Empirical Measures. *Annals of Probability*, 6(6):899–929, 1978.

[27] R. M. Dudley. *Real Analysis and Probability.* Wadsworth, Belmont, CA, 1989.

[28] R. G. Gallager. *Discrete Stochastic Processes.* Kluwer Academic Publishers, Boston, MA, 1996.

[29] P. Glasserman and D. D. Yao. Some Guidelines and Guarantees for Common Random Numbers. *Management Science*, 38(6):884–908, 1992.

[30] P. W. Glynn. Diffusion Approximations. In *Handbooks in Operations Research and Management Science Vol. 2: Stochastic Models*, pages 145–198, Amsterdam, 1990. North Holland.

[31] G. J. Gordon. Stable Function Approximation in Dynamic Programming. Technical Report CMU-CS-95-103, Carnegie Mellon University, 1995.

[32] L. Gurvits, 1996. Private communication.

[33] L. Gurvits, L. J. Lin, and S. J. Hanson. Incremental Learning of Evaluation Functions for Absorbing Markov Chains: New Methods and Theorems. working paper, 1994.

[34] J. M. Harrison and D. Kreps. Martingales and Arbitrage in Multiperiod Securities Markets. *Journal of Economic Theory*, 20:381–408, 1979.

[35] J. M. Harrison and S. Pliska. Martingales and Stochastic Integrals in the Theory of Continuous Trading. *Stochastic Processes and Their Applications*, 11:215–260, 1981.

[36] D. Haussler. Decision Theoretic Generalizations of the PAC Model for Neural Net and Other Learning Applications. *Information and Computation*, 100:78–150, 1992.

[37] D. Heath, R. Jarrow, and A. Morton. Bond Pricing and the Term Structure of Interest Rates: A New Methodology for Contingent Claims Valuation. *Econometrica*, 60(1):77–105, 1992.

[38] R. Howard. *Dynamic Programming and Markov Processes.* M.I.T. Press, Cambridge, MA, 1960.

[39] T. Jaakkola, M. I. Jordan, and S. P. Singh. On the Convergence of Stochastic Iterative Dynamic Programming Algorithms. *Neural Computation*, 6:1185–1201, 1994.

[40] H. Johnson. Options on the Maximum or the Minimum of Several Assets. *Journal of Financial and Quantitative Analysis*, 22(3):277–283, 1987.

[41] N. Ju. Pricing an American Option by Approximating Its Early Exercise Boundary As a Multi–Piece Exponential Function. To appear in *Review of Financial Studies*, 1998.

[42] I. Karatzas. On the Pricing of American Options. *Applied Mathematics and Operations Research*, pages 37–60, 1988.

[43] P. Konstantopoulos and F. Baccelli. On the Cut-Off Phenomenon in Some Queueing Systems. *Journal of Applied Probability*, 28:683–694, 1991.

[44] P. R. Kumar. Re–Entrant Lines. *Queueing Systems: Theory and Applications*, 13:87–110, 1993.

[45] H. L. Lee and C. Billington. Material Management in Decentralized Supply Chains. *Operations Research*, 41(5):835–847, 1993.

[46] M. L. Littman. *Algorithms for Sequential Decision Making.* PhD thesis, Brown University, Providence, RI, 1996.

[47] D. G. Luenberger. *Optimization by Vector Space Methods.* John Wiley & Sons, New York, NY, 1969.

[48] S. Mahadevan. Average Reward Reinforcement Learning: Foundations, Algorithms, and Empirical Results. *Machine Learning*, 22:1–38, 1996.

[49] P. Marbach, O. Mihatsch, and J.N. Tsitsiklis. Call Admission Control and Routing in Integrated Service Networks Using Reinforcement Learning. Submitted to the IEEE Conference on Decision and Control, 1998.

[50] R. C. Merton. Theory of Rational Option Pricing. *Bell Journal of Economics and Management Science*, 4:141–183, 1973.

[51] R. C. Merton. *Continuous–Time Finance.* Basil Blackwell, Oxford, UK, 1992.

[52] S. Meyn, 1997. Private communication.

[53] S. Meyn and R. Tweedie. *Markov Chains and Stochastic Stability.* Springer–Verlag, London, UK, 1993.

[54] W. D. Nordhaus. *Managing the Global Commons: the Economics of Climate Change.* MIT Press, Cambridge, MA, 1994.

[55] C. H. Papadimitriou and J. N. Tsitsiklis. The Complexity of Optimal Queueing Network Control. preprint, 1995.

[56] M. Parsley. Exotics Enter the Mainstream. *Euromoney*, March:127–130, 1997.

[57] F. Pineda. Mean–Field Analysis for Batched TD($\lambda$). To appear in *Neural Computation*, 1996.

[58] D. Pollard. *Empirical Processes: Theory and Applications*, volume 2. Institute of Mathematical Statistics and American Statistical Association, Hayward, CA, 1990.

[59] M. Reed and S. Simon. *Methods of Modern Mathematical Physics I: Functional Analysis.* Academic Press, New York, NY, 1980.

[60] M. Rosenblatt. *Random Processes.* Springer–Verlag, New York, NY, 1974.

[61] G. A. Rummery. *Problem Solving with Reinforcement Learning.* PhD thesis, Cambridge University, Cambridge, UK, 1995.

[62] G. A. Rummery and M. Niranjan. On–Line $Q$–Learning Using Connectionist Systems. Technical Report CUED/F–INFENG/TR 166, Engineering Department, Cambridge University, 1994.

[63] J. Rust. Using Randomization to Break the Curse of Dimensionality. To appear in *Econometrica*, 1996.

[64] R. E. Schapire and M. K. Warmuth. On the Worst–Case Analysis of Temporal–Difference Learning Algorithms. *Machine Learning*, 22:95–122, 1996.

[65] A. Schwartz. A Reinforcement Learning Method for Maximizing Undiscounted Rewards. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 298–305, 1993.

[66] A. N. Shiryaev. *Optimal Stopping Rules.* Springer–Verlag, New York, NY, 1978.

[67] S. P. Singh. Reinforcement Learning Algorithms for Average Payoff Markovian Decision Processes. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 202–207, 1994.

[68] S. P. Singh and D. P. Bertsekas. Reinforcement Learning for Dynamic Channel Allocation in Cellular Telephone Systems. In *Advances in Neural Information Processing Systems 10*, Cambridge, MA, 1997. MIT Press.

[69] S. P. Singh, T. Jaakkola, and M. I. Jordan. Reinforcement Learning with Soft State Aggregation. In *Advances in Neural Information Processing Systems 7*, Cambridge, MA, 1994. MIT Press.

[70] S. P. Singh and R. S. Sutton. Reinforcement Learning with Replacing Eligibility Traces. *Machine Learning*, 22:123–158, 1996.

[71] G. D. Stamoulis and J. N. Tsitsiklis. On the Settling Time of the Congested $GI/G/1$ Queue. *Advances in Applied Probability*, 22:929–956, 1990.

[72] R. S. Sutton. Learning to Predict by the Methods of Temporal Differences. *Machine Learning*, 3:9–44, 1988.

[73] R. S. Sutton. On the Virtues of Linear Learning and Trajectory Distributions. In *Proceedings of the Workshop on Value Function Approximation, Machine Learning Conference 1995.* Technical Report CMU-CS-95-206, Carnegie Mellon University, 1995.

[74] R. S. Sutton. Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding. In *Advances in Neural Information Processing Systems 8*, Cambridge, MA, 1996. MIT Press.

[75] R. S. Sutton and A. G. Barto. Toward a Modern Theory of Adaptive Networks: Expectation and Prediction. *Psychological Review*, 88:135–170, 1981.

[76] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[77] P. Tadepalli and D. Ok. Model–Based Average Reward Reinforcement Learning. To appear in *Artificial Intelligence*, 1998.

[78] G. Tesauro and G. R. Galperin. On–Line Policy Improvement Using Monte Carlo Search. In *Advances in Neural Information Processing Systems 9*, Cambridge, MA, 1997. MIT Press.

[79] G. J. Tesauro. Practical Issues in Temporal Difference Learning. *Machine Learning*, 8:257–277, 1992.

[80] G. J. Tesauro. TD–Gammon, a Self–Teaching Backgammon Program, Achieves Master–Level Play. *Neural Computation*, 6(2):215–219, 1994.

[81] G. J. Tesauro. Temporal Difference Learning and TD–Gammon. *Communications of the ACM*, 38:58–68, 1995.

[82] J. N. Tsitsiklis. Asynchronous Stochastic Approximation and Q–Learning. *Machine Learning*, 16:185–202, 1994.

[83] J. N. Tsitsiklis and B. Van Roy. Feature–Based Methods for Large Scale Dynamic Programming. *Machine Learning*, 22:59–94, 1996.

[84] J. N. Tsitsiklis and B. Van Roy. An Analysis of Temporal–Difference Learning with Function Approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

[85] J. N. Tsitsiklis and B. Van Roy. Average–Cost Temporal–Difference Learning. Technical Report LIDS-P-2390, MIT Laboratory for Information and Decision Systems, 1997.

[86] J. N. Tsitsiklis and B. Van Roy. Optimal Stopping of Markov Processes: Hilbert Space Theory, Approximation Algorithms, and an Application to Pricing High-Dimensional Financial Derivatives. Technical Report LIDS-P-2389, MIT Laboratory for Information and Decision Systems, 1997.

[87] B. Van Roy, D. P. Bertsekas, Y. Lee, and J. N. Tsitsiklis. A Neuro–Dynamic Programming Approach to Retailer Inventory Management. preprint, 1997.

[88] V. N. Vapnik and A. Y. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to their Probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[89] M. Vidyasagar. *A Theory of Learning and Generalization with Applications to Neural Networks and Control Systems.* Springer–Verlag, London, UK, 1997.

[90] J. Walrand. *An Introduction to Queueing Networks.* Prentice Hall, Englewood Cliffs, NJ, 1988.

[91] C. J. C. H. Watkins. *Learning From Delayed Rewards.* PhD thesis, Cambridge University, Cambridge, UK, 1989.

[92] C. J. C. H. Watkins and P. Dayan. Q–learning. *Machine Learning*, 8:279–292, 1992.

[93] P. J. Werbos. Building and Understanding Adaptive Systems: a Statistical/Numerical Approach to Factory Automation and Brain Research. *IEEE Transactions on Systems, Man, and Cybernetics*, 17:7–20, 1987.

[94] P. J. Werbos. Approximate Dynamic Programming for Real–Time Control and Neural Modeling. In D. A. White and D. A. Sofge, editors, *Handbook of Intelligent Control*, 1992.

[95] P. J. Werbos. Neurocontrol and Supervised Learning: An Overview and Evaluation. In D. A. White and D. A. Sofge, editors, *Handbook of Intelligent Control*, 1992.

[96] W. Zhang and T. G. Dietterich. A Reinforcement Learning Approach to Job Shop Scheduling. In *Proceeding of the IJCAI*, 1995.