Approximate dynamic programming with $(\min, +)$ linear function approximation for Markov decision processes.

Chandrashekar L[†] Shalabh Bhatnagar[$]

IISc-CSA-SSL-TR-2014-17

http://stochastic.iisc.ernet.in/www/research/files/IISc-CSA-SSL-TR-2014-17.pdf

Stochastic Systems Lab
Computer Science and Automation
Indian Institute of Science, India

March 2014

Approximate dynamic programming with $(\min, +)$ linear function approximation for Markov decision processes.

Chandrashekar L[†] Shalabh Bhatnagar[$]

March 17, 2014

Abstract

Markov Decision Processes (MDP) is an useful framework to cast optimal sequential decision making problems. Given any MDP the aim is to find the optimal action selection mechanism i.e., the optimal policy. Typically, the optimal policy (u^*) is obtained by substituting the optimal value-function (J^*) in the Bellman equation. Alternately u^* is also obtained by learning the optimal state-action value function Q^* known as the Q value-function. However, it is difficult to compute the exact values of J^* or Q^* for MDPs with large number of states. Approximate Dynamic Programming (ADP) methods address this difficulty by computing lower dimensional approximations of J^*/Q^*. Most ADP methods employ linear function approximation (LFA), i.e., the approximate solution lies in a subspace spanned by a family of pre-selected basis functions. The approximation is obtain via a linear least squares projection of higher dimensional quantities and the L_2 norm plays an important role in convergence and error analysis. In this paper, we discuss ADP methods for MDPs based on LFAs in $(\min, +)$ algebra. Here the approximate solution is a $(\min, +)$ linear combination of a set of basis functions whose span constitutes a subsemimodule. Approximation is obtained via a projection operator onto the subsemimodule which is different from linear least squares projection used in ADP methods based on conventional LFAs. MDPs are not $(\min, +)$ linear systems, nevertheless, we show that the monotonicity property of the projection operator helps us to establish the convergence of our ADP schemes. We also discuss future directions in ADP methods for MDPs based on the $(\min, +)$ LFAs.

1 Introduction

Optimal sequential decision making problems in science, engineering and economics can be cast in the framework of Markov Decision Processes (MDP). Given an MDP, it is of interest to compute the optimal value-function ($J^* \in \mathrm{R}^n$) and/or the optimal-policy(u^*), or $Q^* \in \mathrm{R}^{n \times d}$ known as the Q value-function which encodes both J^* and u^*. The Bellman operator and Bellman equation play a central role in computing optimal value-function ($J^* \in \mathrm{R}^n$) and optimal policy (u^*). In particular, $J^* = TJ^*$ and $Q^* = HQ^*$, where $T\colon \mathrm{R}^n \to \mathrm{R}^n$, $H\colon \mathrm{R}^{n\times d} \to \mathrm{R}^{n \times d}$ are the Bellman and Q-Bellman operators respectively. Most methods to solve MDP such as value/policy iteration (Bertsekas [2007]) exploit the fact that J^* and Q^* are fixed points of the T and H.

[†]Dept Of CSA, IISc chandrul@csa.iisc.ernet.in, [$]Dept Of CSA, IISc shalabh@csa.iisc.ernet.in

Most problems arising in practice have large number of states and it is expensive to compute exact values of J^*/Q^* and u^*. A practical way to tackle the issue is by resorting to approximate methods. Approximate Dynamic Programming (ADP) refers to an entire spectrum of methods that aim to obtain approximate value-functions and/or policies. In most cases, ADP methods consider a family of functions and pick a function that approximates the value function well. Usually, the family of functions considered is the linear span of a set of basis functions. This is known as linear function approximation (LFA) wherein the value-function of a policy u is approximated as $J_u \approx \tilde{J}_u = \Phi r^*$. Here Φ is an $n \times k$ *feature* matrix and $r^* \in \mathrm{R}^k$ ($k << n$) is the weight vector to be computed.
Given Φ, ADP methods vary in the way they compute r^*. In this paper, we focus on ADP methods that solve the following Projected Bellman equation (PBE),

$$\Phi r^* = \Pi T_u \Phi r^*, \tag{1}$$

where Π is the projection matrix, $\Pi = \Phi(\Phi^\top D \Phi)^{-1}\Phi^\top$ and D is any positive definite matrix. Once can show that (1) has a solution by showing that the projected Bellman operator (PBO) ΠT_u is a contraction map in the L_2 norm. Solving (1) only address the problem of policy-*evaluation/prediction*, and to address the problem of *control* a policy improvement step is required. In order to guarantee an improvement in the policy, the prediction error $||J_u - \tilde{J}_u||_\infty$ needs to be bounded. Due to the use of linear least squares projection operator Π, we can bound only $||J_u - \tilde{J}_u||_D$, where $||x||_D = x^\top D x$. Consequently policy improvement is not guaranteed and an approximate policy iteration scheme will fail to produce a convergent sequence of policies. Thus the problems of prediction and control are addressed only partially. In particular, convergence and performance bounds are unsatisfactory. Also there is no convergent scheme using conventional LFAs that can approximate Q^*.

The $(\min, +)$ algebra differs from conventional algebra, in that $+$ and $\times$ operators are replaced by $\min$ and $+$ respectively. Similarly $(\max, +)$ algebra replaces $+$ with $\max$ and $\times$ with $+$. It is known that finite horizon deterministic optimal control problems with reward criterion are $(\max, +)$ linear transformations which map the cost function to the optimal-value function. A lot of work has been done in literature Akian et al. [2008], McEneaney et al. [2008], McEneaney and Kluberg [2009], Gaubert et al. [2011] that make use of $(\max, +)$ basis to solve to compute approximate value-functions. However, in the case of infinite horizon discounted reward MDP, due to the stochastic nature of the state evolution the Bellman operator (T/H) is neither $(\max, +)$ nor $(\min, +)$ linear, which is a key difference from the aforementioned works that apply $(\max, +)$ basis. The primary focus of the paper it so explore $(\min, +)$ LFAs in ADP schemes as opposed to conventional LFAs. Our specific contributions in this paper are listed below.

1. We argue that the Bellman operator arising in MDPs are neither $(\max, +)$/$(\min, +)$ linear. We justify our choice of $(\min, +)$ linear basis for value-function approximation in infinite horizon discounted reward MDPs.

2. We show that the projected Bellman operator in the $(\min, +)$ basis is a contraction map in the L_∞ norm. This enables us establish convergence and error bounds for the ADP schemes developed in this paper.

3. We develop a convergent ADP scheme called Approximate Q Iteration (AQI) to compute $\tilde{Q} = Q^*$. Thus we solve both *prediction* and *control* problems which was a shortcoming in ADP with conventional LFAs.

4. We also present another convergent ADP scheme called Variational Approximate Q Iteration (VAQI), which is based on the Variational or weak formulation of the PBE.

5. We present the error analysis of AQI and VAQI.

Since the main focus of this paper was to study ADP in $(\min,+)$ LFAs and the properties of the associated PBE, we have left out details on algorithmic implementation and analysis of the computational efficiency. Nevertheless, we present experimental results on random MDPs.

2 Markov Decision Processes

Markov Decision Processes (MDP) are characterized by their state space S, action space A, the reward function $g\colon S\times A\to \mathrm{R}$, and the probability of transition from state s to s' under action a denoted by $p_a(s,s')$. The reward for selecting an action a in state s is denoted by $g_a(s)$. We consider MDP with state space $S=\{1,2,\ldots,n\}$ and action set $A=\{1,2,\ldots,d\}$. For simplicity, we assume that all actions $a\in A$ are feasible in every state $s\in S$. A policy is a map $u\colon S\to A$, and it describes the action selection mechanism[2]. Under a policy u the MDP is a Markov chain and we denote its probability transition kernel by $P_u=(p_{u(i)}(i,j), i=1 \text{ to } n, j=1 \text{ to } n)$. The discounted reward starting from state s following policy u is denoted by $J_u(s)$ and is defined as

$$J_u(s)=\mathbf{E}[\sum_{t=0}^{\infty}\alpha^t g_{a_t}(s_t)|s_0=s,u]. \tag{2}$$

Here $\{s_t\}$ is the trajectory of the Markov chain under u, and $a_t=u(s_t),\forall t\geq 0$. We call $J_u=(J_u(s),\forall s\in S)\in \mathrm{R}^n$ the value-function for policy u. The optimal policy denoted as u^* is given by

$$u^*=\arg\max_{u\in U} J_u(s),\forall s\in S. \tag{3}$$

The optimal value function is then $J^*(s)=J_{u^*}(s),\forall s\in S$. The optimal value function and optimal policy are related by the Bellman Equation as below:

$$J^*(s)=\max_{a\in A}(g_a(s)+\alpha\sum_{s'=1}^{n}p_a(s,s')J^*(s')), \tag{4a}$$

$$u^*(s)=\arg\max_{a\in A}(g_a(s)+\alpha\sum_{s'=1}^{n}p_a(s,s')J^*(s')). \tag{4b}$$

Usually, J^* computed is first and u^* is obtained by substituting J^* in (4b). One can also define the state-action value-function of a policy u known as the Q value-function as follows:

$$Q_u(s,a)=\mathbf{E}[\sum_{t=0}^{\infty}\alpha^t g_{a_t}(s_t)|s_0=s,a_0=a], a_t=u(s_t)\forall t>0. \tag{5}$$

The optimal Q values obeys the Q Bellman equation given below:

$$Q^*(s,a)=g_a(s)+\alpha\sum_{s'}p(s,s')\max_{a'}Q^*(s',a'). \tag{6}$$

[2]The policy thus defined are known as stationary deterministic policy (SDP). Policies can also be non-stationary and randomized. However since there exists a optimal policy that is SDP (Puterman [1994]) we restrict our treatment to SDPs.

It is also known that (Bertsekas [2007]) $J^*(s) = \max_a Q^*(s, a)$. The optimal policy can be computed as $u^*(s) = \arg\max_a Q^*(s, a)$. Thus, in some cases it is beneficial to find Q^* since it encodes both J^* and u^*.

2.1 Basic Solution Methods

It is important to note that J^* and Q^* are fixed points of maps $T\colon \mathbf{R}^n \to \mathbf{R}^n$, $H\colon \mathbf{R}^{n\times d} \to \mathbf{R}^{n\times d}$ respectively defined as follows:

$$(TJ)(s) = \max_{a\in A}(g_a(s) + \alpha \sum_{j=1}^{n} p_a(s, s')J(s')), J \in \mathbf{R}^n. \tag{7a}$$

$$(HQ)(s, a) = (g_a(s) + \alpha \sum_{j=1}^{n} p_a(s, s') \max_{a\in A} Q(s', a)), Q \in \mathbf{R}^{n\times d}. \tag{7b}$$

T and H are called the Bellman and Q-Bellman operators respectively. Given $J \in \mathrm{R}^n, Q \in \mathrm{R}^{n\times d}$, TJ and HQ are the 'one-step' greedy value-functions. We summarize certain useful properties of H in the following Lemmas (see Bertsekas [2007] for proofs).

Lemma 1 *H is a* $\max$*-norm contraction operator, i.e., given $Q_1, Q_2 \in \mathbf{R}^{n\times d}$*

$$||HQ_1 - HQ_2||_\infty \leq \alpha||Q_1 - Q_2||_\infty \tag{8}$$

Corollary 1 *Q^* is a unique fixed point of H.*

Lemma 2 *H is a monotone map, i.e., given $Q_1, Q_2 \in \mathrm{R}^{n\times d} n \times d$ is such that $Q \geq HQ$, it follows that $Q \geq Q^*$.*

Lemma 3 *Given $Q \in \mathrm{R}^{n\times d}$, and $k \in \mathrm{R}$ and $\mathbf{1} \in \mathrm{R}^{n\times d}$ a vector with all entries 1, then*

$$H(Q + k\mathbf{1}) = HQ + \alpha k\mathbf{1}. \tag{9}$$

It is easy to check that Lemmas1, 2, 3 hold for T as well (Bertsekas [2007]).
Value iteration (VI) is the most basic method to compute J^*/Q^* and is given by

$$J_{n+1} = TJ_n, \tag{10a}$$
$$Q_{n+1} = HQ_n. \tag{10b}$$

Iterations in (10) are exact methods, and the contraction property of the Bellman operator ensures that $J_n \to J^*$ in (10a), $Q_n \to Q^*$ in (10b) as $n \to \infty$. They are also referred to as *look-up-table* methods or full state representation methods, as opposed to methods employing function approximation. u^* can be computed by substituting J^* in (4b). Another basic solution method is Policy Iteration (PI) presented in Algorithm 1. VI and PI form the basis of ADP methods explained in the next section.

Algorithm 1 Policy Iteration

Start with any policy u_0
for $i = 0, 1, \ldots, n$ **do**
  Evaluate policy u_i by computing J_{u_i}.
  Improve policy $u_{i+1}(s) = \arg\max_a (g_a(s) + \alpha \sum_{s'} p_a(s,s') J_{u_i}(s'))$.
end for
return u_n

3 Approximate Dynamic Programming in conventional LFAs

The phenomenon called *curse-of-dimensionality* denotes the fact that the number of states grows exponentially in the number of state variables. Due to the *curse*, as the number of variables increase, it is hard to compute exact values of J^*/Q^* and u^*. Approximate Dynamic Programming (ADP) methods make use of (4) and dimensionality reduction techniques to compute an approximate value-function $\tilde{J}$. Then $\tilde{J}$ can be used to obtain an approximate policy $\tilde{u}$ which is greedy with respect to $\tilde{J}$ as follows:

$$\tilde{u}(s) = \arg\max_a (g_a(s) + \alpha \sum_{s'} p_a(s,s') \tilde{J}(s')). \tag{11}$$

The sub-optimality of the greedy policy is given by the following result.

Lemma 4 *Let $\tilde{J} = \Phi r^*$ be the approximate value function and $\tilde{u}$ be as in* (11)*, then*

$$||J_{\tilde{u}} - J^*||_\infty \leq \frac{2}{1-\alpha} ||J^* - \tilde{J}||_\infty \tag{12}$$

Proof: See Bertsekas [2007]. Thus a good ADP method is one that address both *prediction* (i.e., computing $\tilde{J}$) and the *control* (i.e., computing $\tilde{u}$) problems with desirable approximation guarantees.
Linear function approximators (LFA) have been widely employed for their simplicity and ease of computation. LFAs typically let $\tilde{J} \in V \subset \mathrm{R}^n$, where V is the subspace spanned by a set of preselected basis functions $\{\phi_i \in \mathrm{R}^n, i = 1, \ldots, k\}$. Let Φ be the $n \times k$ matrix with columns $\{\phi_i\}$, and $V = \{\Phi r | r \in \mathrm{R}^k\}$, then the approximate value function $\tilde{J}$ is of the form $\tilde{J} = \Phi r^*$ for some $r^* \in \mathrm{R}^k$. r^* is a weight vector to be learnt, and due to dimensionality reduction ($k << n$) computing $r^* \in \mathrm{R}^k$ is easier than computing $J^* \in \mathrm{R}^n$.
We now discuss popular ADP methods namely approximate policy evaluation (APE) and approximate policy iteration (API), which are approximation analogues of VI and PI. APE and API are based on linear least squares projection of higher dimensional quantities onto V.

3.1 Approximate Policy Evaluation

J^*/Q^* are not known and hence projecting them onto V is impossible. Nevertheless, one can use the Bellman operator and linear least squares projection operator to write down a projected Bellman equation (PBE) as below:

$$\Phi r^* = \Pi T_u \Phi r^* \tag{13}$$

where $\Pi = \Phi(\Phi^\top D\Phi)^{-1}\Phi^\top$ is the projection operator, D is any diagonal matrix with all entries strictly greater than 0 and T_u is the Bellman operator restricted to a policy u and is given by

$$(T_u J)(s) = g_{u(s)}(s) + \alpha \sum_{s'} p_{u(s)}(s, s')J(s'), J \in \mathrm{R}^n.$$

The approximate policy evaluation (APE) is the following iteration:

$$\Phi r_{n+1} = \Pi T_u \Phi r_n. \tag{14}$$

The following Lemma 5 establishes the convergence of (14).

Lemma 5 *If $\Pi = \Phi(\Phi^\top D\Phi)^{-1}\Phi^\top$, and D be a diagonal matrix with the i^{th} diagonal entry being the stationary probability of visiting state i under policy u, then ΠT_u is a contraction map with factor α.*

Proof: Let P_u be the probability transition matrix corresponding to policy u. Then one can show that (Bertsekas [2007]) for $z \in \mathrm{R}^n$ and $||z||_D^2 = z^\top Dz$, $||P_u z||_D^2 \leq ||z||_D^2$. Also, we know that Π is a non-expansive map, because

$$\begin{aligned} ||\Pi x - \Pi y||_D &= ||\Pi(x - y)||_D \\ &\leq ||\Pi(x - y)||_D + ||(I - \Pi)(x - y)||_D \\ &= ||x - y||_D \end{aligned}$$

Then

$$||\Pi T J_1 - \Pi T J_2||_D \leq ||T J_1 - T J_2||_D \leq \alpha ||J_1 - J_2||_D$$

Corollary 2 *Then the iteration in (14) converges to Φr^* such that $\Phi r^* = \Pi T_u \Phi r^*$.*

The error bound for Φr^* is given by

$$||J_u - \Phi r^*||_D \leq \frac{1}{\sqrt{1 - \alpha^2}} ||J_u - \Pi J_u||_D, \tag{15}$$

One is inclined to think that an approximation analogue of (10a) would yield an approximation to J^*. It is important to note that (14) only computes $\tilde{J}_u$ because (14) contains T_u and not T. However, since operator ΠT might not be a contraction map in the L_2 norm, and T_u cannot be replaced by T in iteration (14), and the PBE in (13).

3.2 Approximate Policy Iteration

Approximate Policy Iteration (Algorithm 2) tackles both prediction and control problems, by performing APE and policy improvement at each step. The performance guarantee of API can be stated as follows:

Lemma 6 *If at each step i one can guarantee that $||\tilde{J}_i - J_{u_i}||_\infty \leq \delta$, then one can show that $\lim_{n\to\infty} ||J_{u_i} - J^*|| \leq \frac{2\delta\alpha}{(1 - \alpha)^2}$.*

Note that the error bound required by Lemma 6 is in the L_∞ norm, whereas (15) is only in the L_2 norm. So API cannot guarantee an approximate policy improvement each step which is a shortcoming. Also, even-though each evaluation step (line 3 of Algorithm 2) converges, the sequence $u_n, n \geq 0$ is not guaranteed to converge. This is known as policy *chattering* and is another important shortcoming of conventional LFAs. Thus the problem of *control* is only partially addressed by API.

Algorithm 2 Approximate Policy Iteration (API)

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

3.3 LFAs for Q value-function

To alleviate the shortcomings in API, it is then natural to look for an approximation method that computes a policy in more direct fashion. Since we know that by computing Q^* we obtain the optimal policy directly, it is a good idea to approximate Q^*. The PBE version of (10b) is a plausible candidate and the iterations are given by,

$$\Phi r_{n+1} = \Pi H \Phi r_n. \tag{16}$$

The above scheme will run into the following problems:

1. The H operator (7b) contains a $\max$ term, and it is not straightforward show that ΠH is a contraction map in L_2 norm, and consequently one cannot establish the convergence of iterates in (16).

2. The issue pertaining to $\max$ operator can be alleviated by restricting H to a policy u, i.e., consider iterations of form

$$\Phi r_{n+1} = \Pi H_u \Phi r_{n+1}. \tag{17}$$

   But iterates in (17) attempt to approximate Q_u and not Q^*, which means the problem of *control* is unaddressed.

We conclude this section with the observation that the important shortcomings of conventional LFAs related to convergence and error bound arise due to the L_2 norm. The main result of the paper is that ADP methods based on $(\min, +)$ LFAs don't suffer from such shortcomings.

4 $(\min, +)$/$(\max, +)$ non-linearity of MDPs

We introduce the $\mathbf{R}_{\min}$ semiring and show that MDPs are neither $(\min, +)$ nor $(\max, +)$ linear. The $\mathbf{R}_{\min}$ semiring is obtained by replacing multiplication $(\times)$ by $+$, and addition $(+)$ by $\min$.

Definition 7

Addition: $x \oplus y = \min(x, y)$
Multiplication: $x \otimes y = x + y$

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

Definition 8

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

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

It is a well known fact that deterministic optimal control problems with cost/reward criterion are $(\min,+)$/$(\max,+)$ linear. However, the Bellman operator T in (7a) (as well as H in (7b)) corresponding to infinite horizon discounted reward MDPs are neither $(\min,+)$ linear nor $(\max,+)$ linear systems. We illustrate this fact via the following example.

Example 1 *Consider an MDP with two states $S=\{s_1,s_2\}$, only one action, and a reward function g, and let the probability transition kernel be*

$$P=\begin{bmatrix}0.5 & 0.5\\ 0.5 & 0.5\end{bmatrix} \tag{18}$$

For any $J\in\mathrm{R}^2$ the Bellman operator $T\colon\mathrm{R}^2\to\mathrm{R}^2$ can be written as

$$(TJ)(s)=g(s)+\alpha\times(0.5\times J(1)+0.5\times J(2)) \tag{19}$$

Consider vectors $J_1,J_2\in\mathrm{R}^2$ such that $J_1=(1,2)$ and $J_2=(2,1)$ and $J_3=\max(J_1,J_2)=(2,2)$. Let $g(1)=g(2)=1$, and $\alpha=0.9$, then, it is easy to check that $TJ_1=TJ_2=(2.35,2.35)$, and $TJ_3=(2.8,2.8)$. However $TJ_3\neq\max(TJ_1,TJ_2)$, i.e., $TJ_3\neq(2.35,2.35)$. Similarly one can show that T is neither a $(\min,+)$ linear operator.

■

5 $(\min,+)$ linear functions

Even-though the Bellman operator is not $(\min,+)$/$(\max,+)$ linear, the motivation behind developing ADP methods based on $(\min,+)$ LFAs is to explore them as an alternative to the conventional basis representation. Thus the aim to understand the kind of convergence guarantees and error bounds that are possible in the $(\min,+)$ LFAs.

Given a set of basis function $\{\phi_i,i=1,\ldots,k\}$, we define its $(\min,+)$ linear span to be $\mathcal{V}=\{v|v=\Phi\otimes r\overset{def}{=}\min(\phi_1+r(1),\ldots,\phi_k+r(k)),r\in\mathbf{R}^k_{\min}\}$. $\mathcal{V}$ is a subsemimodule. In the context value function approximation, we would want to project quantities in $\mathbf{R}^n_{\min}$ onto $\mathcal{V}$. The $(\min,+)$ projection operator Π_M is given by (Akian et al. [2008], Cohen et al. [1996], McEneaney et al. [2008])

$$\Pi_M u=\min\{v|v\in\mathcal{V},v\geq u\},\forall u\in\mathcal{M}. \tag{20}$$

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

$$\begin{aligned} v &= \Pi_M T v, v \in \mathcal{V} \\ \Phi \otimes r^* &= \min\{\Phi \otimes r^* \in \mathcal{V} | \Phi \otimes r^* \geq T\Phi \otimes r^*\} \end{aligned} \tag{21}$$

Our choice is justified by the similarity in the structure of (21) and the linear programming (LP) formulation of the infinite horizon discounted reward MDP.

$$\begin{aligned} &\min\ c^\top J \\ &s.t \quad J(s) \geq g(s,a) + \alpha \sum_{s'} p_a(s,s')J(s'), \forall s \in S, a \in A \end{aligned} \tag{22}$$

Thus by making use of $(\min, +)$ LFAs and the projection operator Π_M we aim to find the minimum upper bound to J^*/Q^*. A closely related ADP method is the approximate linear program (ALP) given by

$$\begin{aligned} &\min\ c^\top \Phi r \\ &s.t \quad \Phi r(s) \geq g(s,a) + \alpha \sum_{s'} p_a(s,s')(\Phi r)(s'), \forall s \in S, a \in A \end{aligned} \tag{23}$$

Though formulations (21) and (23) look similar, the former has its search space which is a subsemimodule formed by a $(\min, +)$ linear span, whereas the latter has search space which is the intersection of subspace and set of constraints. The two formulations differ in the algorithmic implementation and performance bounds which we discuss in a longer version of the paper.

6 $(\min, +)$ LFAs for Q value approximation

We now present an ADP scheme bases on solving the PBE in $(\min, +)$ basis to compute approximate $Q^* \approx \tilde{Q}$. Our ADP scheme successfully addresses the two shortcomings of the ADP scheme in conventional basis. First, we show establish contraction of the projected Bellman operator in the L_∞ norm. This enables us to show that our recursion to compute $\tilde{Q}$ converges. As discussed earlier, we can also obtain a greedy policy $\tilde{u}(s) = \max_a \tilde{Q}(s,a)$. Secondly, we also present an error bound for the $\tilde{Q}$ in the $\max$ norm and as a consequence we can also ascertain the performance of $\tilde{u}$.
The PBE we are interested in solving is[3]

$$\Phi \otimes r^* = \Pi_M H \Phi \otimes r^* \tag{24}$$

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

$$\Phi \otimes r_{n+1} = \Pi_M H \Phi \otimes r_n. \tag{25}$$

The following results help us to establish the fact that that the operator $\Pi_M H \colon \mathbf{R}^{n\times d}_{\min} \to \mathbf{R}^{n\times d}_{\min}$ is a contraction map in the L_∞ norm.

[3] We did not consider $\Phi \otimes r^* = \Pi_M T\Phi \otimes r^*$ since it approximates only J^* and is superseded by (24) which computes approximate $Q^* \approx \tilde{Q}$. Thus (24) addresses both *prediction* and *rol* problems. We wish the remind the reader of the issues with (24) in conventional basis discussed in section 3.3.

Lemma 9 *For $Q_1, Q_2 \in \mathbf{R}_{\min}^{n\times d}$, such that $Q_1 \geq Q_2$, then $\Pi_M Q_1 \geq \Pi_M Q_2$.*

Proof: Follows from definition of projection operator in (20).

Lemma 10 *Let $Q \in \mathbf{R}_{\min}^{n\times d}$, $V_1 = \Pi_M Q$ be its projection onto $\mathcal{V}$ and $k \in \mathrm{R}$, and $\mathbf{1} \in \mathrm{R}^{n\times d}$ be a vector with all components equal to 1. The projection of $Q + k\mathbf{1}$ in $V_2 = \Pi_M Q + k\mathbf{1}$.*

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

Theorem 11 *$\Pi_M H$ is a contraction map in L_∞ norm with factor α.*

Proof: Let $Q_1, Q_2 \in \mathbf{R}_{\min}^{n\times d}$, define $\epsilon \overset{def}{=} ||Q_1 - Q_2||_\infty$, then

$$\begin{aligned} \Pi_M H Q_1 - \Pi_M H Q_2 &\leq \Pi_M H(Q_2 + \epsilon\mathbf{1}) - \Pi_M H Q_2 \\ &= \Pi_M (HQ_2 + \alpha\epsilon\mathbf{1}) - \Pi_M H Q_2 \\ &= \alpha\epsilon\mathbf{1}. \end{aligned} \tag{26}$$

Similarly $\Pi_M H Q_2 - \Pi_M H Q_1 \leq \alpha\epsilon\mathbf{1}$, so it follows that $||\Pi_M H Q_1 - \Pi_M H Q_2||_\infty \leq \alpha ||Q_1 - Q_2||_\infty$.

Corollary 3 *The approximate Q iteration in (25) converges to a fixed point r^*.*

7 Variational formulation of PBE

The projection operator Π_M used in (16) is exact. Let $v = \Pi_M u$, then for $\{w_i \in \mathbf{R}_{\min}^n\}, i = 1, \ldots, m$ it follows from the definition of Π_M that

$$w_i^\top v \geq w_i^\top u \tag{27}$$

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

$$\Pi_M^W u = \min\{v \in \mathcal{V} | W^\top v \geq W^\top u\}. \tag{28}$$

The superscript in Π_M^W denotes the test matrix W. The iteration to compute approximate Q values using Π_M^W is given by

$$\Phi \otimes r_{n+1} = \Pi_M^W H \Phi \otimes r_n. \tag{29}$$

Lemmas 9, 10, and Theorem 11 continue to hold if Π_M is replaced with Π_M^W. Thus by Corollary 3, we know that (29) converges to an unique fixed point r_W^* such that $\Phi \otimes r_W^* = \Pi_M^W H \Phi \otimes r_W^*$.

Theorem 12 *Let $\tilde{r}$ be such that $\tilde{r} = \arg\min_r ||Q^* - \Phi \otimes r||_\infty$. Let r^* be the fixed point of the iterates in (29), then*

$$\begin{aligned} ||Q^* - \Phi \otimes r^*||_\infty \leq \frac{2}{1+\alpha}(&||Q^* - \Phi \otimes \tilde{r}||_\infty \\ &+ ||\Phi \otimes \tilde{r} - \Pi_M^W \Phi \otimes \tilde{r}||_\infty) \end{aligned} \tag{30}$$

Proof: Let $\epsilon = ||Q^* - \Phi \otimes \tilde{r}||_\infty$, by contraction property of H (Lemma 1) we know that

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

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

$$\begin{aligned}||\Phi \otimes \tilde{r} - \Pi_M^W H\Phi \otimes \tilde{r}||_\infty &= ||\Phi \otimes \tilde{r} - \Pi_M^W \Phi \otimes \tilde{r}||_\infty \\ &+ ||\Pi_M^W \Phi \otimes \tilde{r} - \Pi_M^W H\Phi \otimes \tilde{r}||_\infty\end{aligned}$$

Now

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

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

$$||\Phi \otimes \tilde{r} - \Pi_M^W H\Phi \otimes \tilde{r}||_\infty \leq (1+\alpha)\epsilon + \beta, \tag{31}$$

where $\beta = ||\Phi \otimes \tilde{r} - \Pi_M^W \Phi \otimes \tilde{r}||_\infty$. Now consider the iterative scheme in (29) with $r_0 = \tilde{r}$, and

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

The term β in the error bound in Theorem 12 is the error due to the usage of Π_M^W. Thus for solution to (24) $\beta = 0$.

8 Experiments

We test our ADP schemes on a randomly generated MDP with 100 states, i.e., $S = \{1, 2, \ldots, 100\}$, and action set $A = \{1, \ldots, 5\}$. The reward $g_a(s)$ is a random integer between 1 and 10, and discount factor $\alpha = 0.9$. We now describe feature selection, where $\{\phi_j, j = 1, \ldots, k\}, \phi_j \in \mathbf{R}_{\min}^{n \times d}$ and $\{\phi^i, i = 1, \ldots, n\}, \phi^i \in \mathbf{R}_{\min}^k$ denote the columns and rows respectively of the feature matrix Φ. The feature corresponding to a state-action pair (s, a) is given by $\phi^{(s-1)\times d + a}$. Let ϕ^x, ϕ^y be features corresponding to state action pairs (s_x, a_x) and (s_y, a_y) respectively, then

$$< \phi^x, \phi^y >= \phi^x(1) \otimes \phi^y(1) \oplus \ldots \oplus \phi^x(k) \otimes \phi^y(k). \tag{32}$$

We desire the following in the feature matrix Φ.

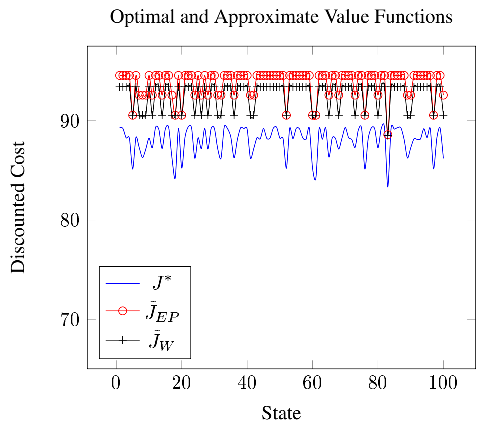

Figure 1: $||J^* - \tilde{J}_{EP}||_\infty = 6.47$,$||J^* - \tilde{J}_W||_\infty = 6.35$

1. Features ϕ^i should have unit norm, i.e., $||\phi^i|| =< \phi^i, \phi^i >= \mathbf{0}$, since $\mathbf{0}$ is the multiplicative identity in the $(\min, +)$ algebra.
2. For dissimilar states-action pairs, we prefer $< \phi^x, \phi^y >= +\infty$, since $+\infty$ is the additive identity in $(\min, +)$ algebra.

Keeping these in mind, we design the feature matrix Φ for the random MDP as in (33). For state-action pair (s, a) let $x = (s-1) \times d + a$, then the feature

$$\phi^x(i) = \begin{cases} 0 & : g_a(s) \in [g_{\min} + \frac{(i-1)L}{k}, g_{\min} + \frac{(i)L}{k}] \\ 1000 & : g_a(s) \notin [g_{\min} + \frac{(i-1)L}{k}, g_{\min} + \frac{(i)L}{k}], \end{cases} \quad \forall i = 1, \ldots, k. \tag{33}$$

We use 1000 in place of $+\infty$. It is easy to verify that Φ in (33) has the enumerated properties. The results are plotted in Figure 2. Here J^* is the optimal value-function, $\tilde{J}_{EP}(s) = \max_a \tilde{Q}_{EP}(s, a)$, where $\tilde{Q}_{EP}$ is the value obtained via the iterative scheme in (25) and subscript EP denotes the fact that the projection employed in *exact* (Π_M). $\tilde{J}_W(s) = \max_a \tilde{Q}_W(s, a)$, where $\tilde{Q}_W$ is the value obtained via the iterative scheme in (29) and subscript W denotes the fact that the projection employed is Π_M^W. $u_{EP}(s) = \arg\max_a \tilde{Q}_{EP}(s, a)$ and $u_W(s) = \arg\max_a \tilde{Q}_W(s, a)$ are greedy policies and $J_{u_{EP}}$, J_{u_W} are their respective value functions. $\tilde{J}_{u_{arbt}}$ is the value function of an *arbitrary* policy, wherein a fixed arbitrary action is chosen for each state.

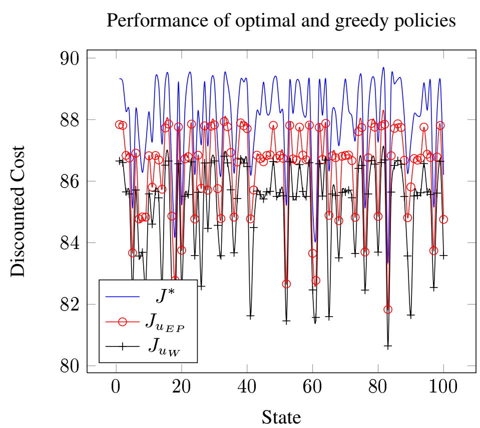

Figure 2: $||J^* - J_{u_{EP}}||_\infty = 2.61$,$||J^* - J_{u_W}||_\infty = 5.61$, $||J^* - \tilde{J}_{u_{arbt}}||_\infty = 40.49$

References

Marianne Akian, Stéphane Gaubert, and Asma Lakhoua. The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis. *SIAM Journal on Control and Optimization*, 47(2):817–848, 2008.

D.P. Bertsekas. *Dynamic Programming and Optimal Control*, volume II. Athena Scientific, Belmont,MA, 3 edition, 2007.

Guy Cohen, Stéphane Gaubert, and Jean-Pierre Quadrat. Kernels, images and projections in dioids. In *Proceedings of WODES96*, pages 151–158, 1996.

Stephane Gaubert, William McEneaney, and Zheng Qu. Curse of dimensionality reduction in max-plus based approximation methods: Theoretical estimates and improved pruning algorithms. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 1054–1061. IEEE, 2011.

William M McEneaney and L Jonathan Kluberg. Convergence rate for a curse-of-dimensionality-free method for a class of hjb pdes. *SIAM Journal on Control and Optimization*, 48(5):3052–3079, 2009.

William M McEneaney, Ameet Deshpande, and Stephane Gaubert. Curse-of-complexity attenuation in the curse-of-dimensionality-free method for hjb pdes. In *American Control Conference, 2008*, pages 4684–4690. IEEE, 2008.

M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Programming*. John Wiley, New York, 1994.