2020 American Control Conference
Denver, CO, USA, July 1-3, 2020

Recent Advances in Reinforcement Learning

M. Vidyasagar

***Abstract*— In this paper, we give a brief review of Markov Decision Processes (MDPs), and how Reinforcement Learning (RL) can be viewed as MDP where the parameters are unknown. Specific topics discussed include the Bellman equation and the Bellman operator, and value and policy iterations for MDPs, together with recent "empirical" approaches to solving the Bellman equation and applying the Bellman iteration. In addition to the well-established method of Q-learning, we also discuss the more recent approach known as Zap Q-learning.**

I. Introduction

Reinforcement learning (RL) is a rather vague phrase that is supposed to capture the human mode of learning in an uncertain environment. Commonly used phrases in RL are exploration and exploitation. Exploration refers to the learner exploring previously unseen situations to see what happens, while exploitation refers to the learner taking advantage of knowledge already gained. An early and common mathematical model for this flavor of RL is the multi-arm bandit problem, whereby a user attempts to choose between several options (often thought of as "slot machines" which explains the nomenclature "bandits"), which have unknown payoffs, so as to maximize the expected reward. An early paper that clearly lays out the tradeoffs between these two facets of exploration and exploitation can be found in [1].

A more general class of learning problems are Markov Decision Processes (MDPs). It is possible to think of a multi-arm bandit problem as a MDP; see [2, Section 3.6]. Over the years there has been a lot of interest in formulating RL as a MDP where the underlying parameters are unknown, and must somehow be "inferred" on the fly. In this paper, we begin by reviewing MDPs, and then RL viewed as a MDP with unknown parameters. The reader is directed to [2], [3], [4] for background material and additional details.

II. Review of Markov Decision Processes

In this section we will briefly review Markov decision processes, often abbreviated as MDP. The emphasis is on the case where the parameters of the MDP are completely known. As shown in the next section, one approach to RL is to view it as a MDP where the underlying parameters are unknown. In the interests of simplicity, the discussion is limited to the situation where the state space underlying the MDP is a finite set. However, MDPs where the underlying state space is a subset of $\mathbb{R}^d$ for some d are also sometimes of interest. Two recent papers [5], [6] present some new techniques for addressing such problems. The latter paper also contains an extensive and relevant bibliography.

M. Vidyasagar is a National Science Chair, and is with the Indian Institute of Technology Hyderabad, India, This research is supported by the Science and Engineering Research Board (SERB), Government of India. Email: m.vidyasagar@iith.ac.in.

A. *Review of Markov Processes*

Though much of the material in this section is standard, it serves to introduce the notation used in the paper. Relevant facts about Markov processes can be found in [7]. Suppose $\mathcal{X}$ is a finite set of cardinality n, written as $\{x_1, \ldots, x_n\}$. Strictly speaking, the order in which the elements of $\mathcal{X}$ are arranged does not matter, but it is assumed that some order is specified and is used throughout. If $\{X_t\}_{t\geq 0}$ is a stationary Markov process assuming values in $\mathcal{X}$, then the corresponding state transition matrix A is defined by

$$a_{ij} = \Pr\{X_{t+1} = x_j | X_t = x_i\}. \tag{1}$$

Thus the i-th row of A is the conditional probability vector of X_{t+1} when $X_t = x_i$. Clearly the row sums of the matrix A are all equal to one. Therefore the induced norm $\|A\|_{\infty\to\infty}$ also equals one. Now suppose that there is a "reward" function $R : \mathcal{X} \to \mathbb{R}$ associated with each state. Choose a "discount factor" $\gamma \in (0, 1)$. Then, along a sample path $\{X_t\}_{t\geq 0}$, the total discounted reward is just the sum of $\gamma^t R(X_t)$. Now, for each state $x_i \in \mathcal{X}$, define the **expected discounted future reward** $V(x_i)$ as

$$V(x_i) = E\left[\sum_{t=0}^{\infty} \gamma^t R(X_t) | X_0 = x_i\right]. \tag{2}$$

Define the vectors

$$\mathbf{v} = [\ V(x_1) \ \ \cdots \ \ V(x_n)\ ]^\top, \tag{3}$$

$$\mathbf{r} = [\ R(x_1) \ \ \cdots \ \ R(x_n)\ ]^\top. \tag{4}$$

Then it is easy to show that $\mathbf{v}$ satisfies the recursive relationship

$$\mathbf{v} = \mathbf{r} + \gamma A\mathbf{v}. \tag{5}$$

Since the induced matrix norm $\|A\|_{\infty\to\infty} \leq 1$ and $\gamma < 1$, it follows that if we equip $\mathbb{R}^n$ with the norm $\|\cdot\|_\infty$, the map $\mathbf{z} \mapsto \mathbf{r} + \gamma A\mathbf{z}$ is a contraction. Therefore, for every fixed assignment of rewards to states, there is a unique $\mathbf{v}$ that satisfies (5). In principle one could solve (5) by repeated application of the contraction map. Note that the faster future rewards are discounted (i.e., the smaller γ is), the faster the iterations will converge.

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on November 16,2022 at 07:29:00 UTC from IEEE Xplore. Restrictions apply.

B. Markov Decision Processes: Formulation

In a Markov process, the state X_t evolves on its own. In contrast, in a MDP, there is also another variable called the "action" which affects the dynamics. Specifically, in addition to the state space $\mathcal{X}$, there is also a finite set of actions $\mathcal{U}$. At time t, when the state is X_t, an action $U_t \in \mathcal{U}$ is applied. The state transition matrix of a MDP is defined via

$$a_{ij}^u = \Pr\{X_{t+1} = x_j | X_t = x_i, U_t = u\}. \tag{6}$$

Obviously, for each fixed $u \in \mathcal{U}$, the corresponding state transition matrix A^u is column-stochastic. There is also a "reward" function $R : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$.

The most important aspect of a MDP is the concept of a "policy," which is just a systematic way of choosing U_t given X_t. One can make a distinction between deterministic and probabilistic policies. A deterministic policy is just a map from $\mathcal{X}$ to $\mathcal{U}$. A probabilistic policy is a map from $\mathcal{X}$ from the set of probability distributions on $\mathcal{U}$. Let Π_d, Π_p denote respectively the set of deterministic, and the set of probabilistic, policies. If $\pi \in \Pi_d$, then for each $x_i \in \mathcal{X}$, we have that $\pi(x_i) \in \mathcal{U}$. If $\pi \in \Pi_p$, then for each $x_i \in \mathcal{X}$, $\pi(x_i)$ belongs to the $|\mathcal{U}|$-dimensional simplex (consisting of probability distributions on $\mathcal{U}$). Clearly the number of deterministic policies is $|\mathcal{U}|^{|\mathcal{X}|}$, while Π_p is uncountable.

Whether a policy π is deterministic or probabilistic, the resulting stochastic process $\{X_t\}$ is Markov with the state transition matrix determined as follows: If $\pi \in \Pi_d$, then

$$\Pr\{X_{t+1} = x_j | X_t = x_i, \pi\} = a_{ij}^{\pi(x_i)}. \tag{7}$$

If $\pi \in \Pi_s$ and

$$\pi(x_i) = [\ (\pi(x_i))_1 \quad \cdots \quad (\pi(x_i))_m\ ], \tag{8}$$

where $m = |\mathcal{U}|$, then

$$\Pr\{X_{t+1} = x_j | X_t = x_i, \pi\} = \sum_{u \in \mathcal{U}} (\pi(x_i))_u a_{ij}^u. \tag{9}$$

(Note that the notation in (9) is a little imprecise.) In either case, we can define A^π to be the state transition matrix that results from applying the policy π. In a similar manner, for every policy π, the reward function $R : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ can be converted into a reward map $R_\pi : \mathcal{X} \to \mathbb{R}$, as follows: If $\pi \in \Pi_d$, then

$$R_\pi(x_i) = R(x_i, \pi(x_i)), \tag{10}$$

whereas if $\pi \in \Pi_s$, then

$$R_\pi(x_i) = \sum_{u \in \mathcal{U}} (\pi(x_i))_u R(x_i, u). \tag{11}$$

For a MDP, one can pose three questions of increasing difficulty:

1) **Policy evaluation:** For a given policy π, define the "value" associated with the policy π and initial state x_i as the expected discounted future reward with $X_0 = x_i$, and denote it by $V_\pi(x_i)$. How can $V_\pi(x_i)$ be computed for each $x_i \in \mathcal{X}$?
2) **Optimal Value Determination:** For a specified initial state x_i, define

$$V^*(x_i) := \max_{\pi \in \Pi_d} V_\pi(x_i), \tag{12}$$

to be the **optimal value** over all policies. How can $V^*(x_i)$ be computed? Note that in (12), the optimum is taken over all *deterministic* policies. In principle one could also seek the optimum value over all *probabilistic* policies; but this case does not seem to have been studied very much.
3) **Optimal Policy Determination:** Define the **optimal policy** map $\mathcal{X} \to \Pi_d$ via

$$\pi^*(x_i) := \arg\max_{\pi \in \Pi_d} V_\pi(x_i). \tag{13}$$

How can the optimal policy map π^* be determined?

C. Markov Decision Processes: Solution

In this subsection we present answers to the three questions above.

Policy Evaluation:

Suppose a policy $\pi \in \Pi_d$ is specified. Then the corresponding state transition matrix and reward are given by (7) and (10) respectively. Now suppose we define the vector $\mathbf{v}_\pi$ by

$$\mathbf{v}_\pi = [\ V_\pi(x_1) \quad \ldots \quad V_\pi(x_n)\ ], \tag{14}$$

and the reward vector $\mathbf{r}_\pi$ by

$$\mathbf{r}_\pi = [\ R_\pi(x_1) \quad \ldots \quad R_\pi(x_n)\ ]. \tag{15}$$

Then it is easy to see that $\mathbf{v}_\pi$ satisfies an equation analogous to (5), namely

$$\mathbf{v}_\pi = \mathbf{r}_\pi + \gamma A^\pi \mathbf{v}_\pi. \tag{16}$$

Optimal Value Determination:

Let $V^*(x_i)$ denote the maximum value of the discounted future reward, over all policies $\pi \in \Pi_d$. The key to computing $V^*(x_i)$ is provided by the **Bellman equation**, namely

$$V^*(x_i) = \max_{u \in \mathcal{U}} \left[R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u V^*(x_j) \right]. \tag{17}$$

Note that the above equation is also referred by other names such as the renewal equation, or the principle of optimality. A derivation of (17) can be found in [4, (3.19)]. Note that (17) is *recursive* in that the unknown function V^* occurs on both sides of the equation. Therefore (17) can be thought of as a *characteriation* of V^*, and some iterative procedure is needed to *solve* the equation.

A common approach to solving (17) is known as "value iteration." To present it, let us define the vector $\mathbf{v}^*$ in a manner analogous to earlier notation , namely

$$\mathbf{v}^* = [\ V^*(x_1) \quad \cdots \quad V^*(x_n)\ ]. \tag{18}$$

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on November 16,2022 at 07:29:00 UTC from IEEE Xplore. Restrictions apply.

Next, we define a "value update" map $T : \mathbb{R}^n \to \mathbb{R}^n$, as follows:

$$(T\mathbf{v})_i := \max_{u \in \mathcal{U}} \left[R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u v_j \right]. \qquad (19)$$

One can think of $\mathbf{v}$ as the current guess for the vector $\mathbf{v}^*$, and of $T\mathbf{v}$ as an updated guess. Indeed, the next theorem shows that this intuition is valid.

Theorem 1: For all $\mathbf{v}_0 \in \mathbb{R}^n$, the sequence of iterations $\{T^k \mathbf{v}_0\}$ approaches $\mathbf{v}^*$ as $k \to \infty$.

As before, the proof consists of showing that the map T is a contraction with constant γ. Hence, the more future rewards are discounted, the faster the iterations converge.

Optimal Policy Determination:

Suppose that the optimal value vector $\mathbf{v}^*$ has been determined, either through Theorem 1 of some other means. How can this information be used to determine the optimal policy?

Theorem 2: Suppose the optimal value vector $\mathbf{v}^*$ is known, and define, for each $x_i \in \mathcal{X}$, the policy $\pi^* : \mathcal{X} \to \mathcal{U}$ via

$$\pi^*(x_i) = \arg\min_{u \in \mathcal{U}} \left[R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u v_j^* \right]. \qquad (20)$$

Then π^* is an optimal policy.

Therefore, in principle one could use Theorem 1 to compute the optimal value $\mathbf{v}^*$, and then use Theorem 2 to determine the optimal policy. Note that the determination of the optimal value using Theorem 1 requires an iteration on the guessed value function. The next approach combines this with finding the optimal π and is known as "policy iteration."

Let $\pi \in \Pi_d$ be a (deterministic) policy, and define an associated map $T_\pi : \mathbb{R}^n \to \mathbb{R}^n$ by

$$(T_\pi \mathbf{v})_i := R_\pi(x_i) + \gamma \sum_{j \in [n]} a_{ij}^\pi v_j, \qquad (21)$$

or more compactly

$$T_\pi \mathbf{v} := \mathbf{r}_\pi + \gamma A^\pi \mathbf{v}. \qquad (22)$$

So, if we start with any initial guess $\mathbf{v}_0$, the sequence $\{T_\pi^k \mathbf{v}_0\}$ converges to $\mathbf{v}_\pi$, the unique solution of

$$\mathbf{v}_\pi = \mathbf{r}_\pi + \gamma A^\pi \mathbf{v}_\pi. \qquad (23)$$

In order to find the optimal policy π^*, we apply the following iterative procedure: Start by setting the iteration counter to zero, and choose some arbitrary initial policy π_0. Then at iteration k,

1) Compute the value vector $\mathbf{v}_{\pi_k}$ such that $\mathbf{v}_{\pi_k} = T_{\pi_k} \mathbf{v}_{\pi_k}$. Note that computing $\mathbf{v}_{\pi_k}$ using Theorem 1 would require infinitely many applications of the map T_{π_k} to some arbitrary initial vector.
2) Use this value vector $\mathbf{v}_{\pi_k}$ to compute an updated policy π_{k+1}, via

$$\pi_{k+1}(x_i) = \arg\min_{u \in \mathcal{U}} \left[R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u (\mathbf{v}_{\pi_k})_j \right]. \qquad (24)$$

Note that (24) implies that

$$T_{\pi_{k+1}} \mathbf{v}_{\pi_k} = T_{\pi_k} \mathbf{v}_{\pi_k}. \qquad (25)$$

Theorem 3: We have that

$$\mathbf{v}_{\pi_k} \leq \mathbf{v}_{\pi_{k+1}}, \qquad (26)$$

where the dominance is componentwise. Consequently, there exists a finite iteger k_0 such that $\mathbf{v}_{\pi_k} = \mathbf{v}^*$ for all $k \geq k_0$.

III. Reinforcement Learning as a MDP with Unknown Parameters

The results of the previous section are applicable to the case where the transition probabilities A^u and the reward function : $R : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ are all known. Reinforcement learning can be thought of as the problem of determining the optimal policy π^* when these parameters are unknown.

A. Monte Carlo Methods

The phrase "Monte Carlo" methods is used nowadays to refer to almost any technique wherein an expected value of a random variable is approximated by its empirical average, that is, an average of its observed values. In this subsection, we introduce one approach to approximating the value of a policy for a MDP where the underlying parameters are unknown.

All of the discussion here assumes that some policy π has been chosen and implemented, and that we observe a time series of triplets $\{(X_t, U_t, W_t\}_{t \geq 0}$ where $U_t = \pi(X_t)$ for the known policy π, $W_t = R_\pi(X_t)$ where the policy reward R_π is unknown, and the state transtion matrix A^π resulting from the policy is also unknown. Because π is fixed throughout, we drop the subscript and superscript π.

The discussion in this subsection applies to the case where the underlying Markov process contains one or more absorbing, or terminal, states. Recall that a state x_i is said to be "absorbing" if

$$\Pr\{X_{t+1} = x_i | X_t = x_i\} = 1,$$

or equivalently, the row of the state transition matrix corresponding to the state x_i consists of a 1 in column i and zeros in other columns. The Markov process can have more than one absorbing state. While the dynamics of the MDP are otherwise assumed to be unknown, it is assumed that the learner knows which states are absorbing. By tradition, it is assued that the reward $R(x_i) = 0$ whenever x_i is an absorbing state.

In this setting, an **episode** refers to any sample path $\{(X_t, U_t, W_t\}_{t \geq 0}$. Since the policy π is presumably chosen by the learner, it is always the case that $U_t = \pi(X_t)$. Therefore U_t does not add any new information. Once

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on November 16,2022 at 07:29:00 UTC from IEEE Xplore. Restrictions apply.

X_t reaches an absorbing state, the episode terminates. The underlying assumption is that, once the Markov process reaches an absorbing state, it can be restarted with the initial state distributed according to its stationary (or some other) distribution. This assumption may not always hold in practice.

Define

$$G_t = \sum_{i=0}^{\infty} \gamma^i W_{t+i}. \tag{27}$$

If an absorbing state is reached after a finite time, say T, then the summation can be truncated at time T, because $W_t = 0$ for $t \geq T$. Now it is known that, for a state $x_i \in \mathcal{X}$, we have

$$V_\pi(x_i) = E[G_t | X_t = x_i].$$

Accordingly, suppose that an episode contains the state of interest x_i at time τ, that is, $X_\tau = x_i$. Let us also suppose that the episode terminates at time T. Then the quantity

$$\sum_{i=0}^{T-\tau} \gamma^i W_{\tau+i}$$

provides *one* approximation to $V_\pi(x_i)$. Now suppose we have L episodes, call them $\mathcal{E}_1, \cdots, \mathcal{E}_L$. Let k the number of these episodes in which the state of interest x_i occurs. Without loss of generality, renumber the episodes so that these are $\mathcal{E}_1$ through $\mathcal{E}_k$. For each such episode, let τ denote the *first* time at which x_i appears in the state sequence, and T the time at which the episode terminates.[1] Further, define

$$H_l := \sum_{i=0}^{T-\tau} \gamma^i W_{\tau+i}.$$

Then

$$\frac{1}{k} \sum_{l=1}^{k} H_l \tag{28}$$

provides an estimate for $V_\pi(x_i)$, known as the **first-time estimate**. It is possible that the state of interest x_i occurs multiple times within the same episode. In such a case, one can form multiple estimates H_l and then average them. This called the **everytime** estimate.

Example 1: Suppose $n = 3$, and for convenience label the states as A, B, C. Suppose further that $R(A) = 3$, $R(B) = 2$ and C is an absorbing state for the policy under study. Suppose $L = 3$ and that the three episodes (all terminating at C) are:

$$\mathcal{E}_1 = ABABBC, \mathcal{E}_2 = BBC, \mathcal{E}_3 = BAABC.$$

Now suppose we wish to estimate the value $V(A)$. Then $\mathcal{E}_2$ does not interest us because A does not occur in it. If the discount factor γ equals 0.9, then we can form the following quantities:

$$H_{11} = 3 + 2 \cdot (0.9) + 3 \cdot (0.9)^2 + 2 \cdot (0.9)^3 + 2 \cdot (0.9)^4,$$

$$H_{12} = 3 + 2 \cdot (0.9) + 2 \cdot (0.9)^2,$$

$$H_{31} = 3 + 3 \cdot (0.9) + 2 \cdot (0.9)^2, H_{32} = 3 + 2 \cdot (0.9).$$

Then $(H_{11} + H_{31})/2$ is the first-time estimate for $V(A)$, while $(H_{11} + H_{12} + H_{31} + H_{32})/4$ is the everytime estimate for $V(A)$.

[1]Strictly speaking we should use the notation τ_1, T_1 etc., but we do not do this in the interests of clarity.

B. Temporal Difference Methods

Unlike Monte Carlo methods that make use of episodes, Temporal Difference (TD) methods update various estimates at each time step. As before it is assumed that a particular policy π has been chosen and implemented, and that a sample path $\{(X_t, U_t, W_t\}_{t \geq 0}$ is observed. Let $\hat{\mathbf{v}}_t \in \mathbb{R}^n$ denote the estimated value vector at time t. For convenience, for $x_i \in \mathcal{X}$, we write $\hat{\mathbf{v}}(x_i)$ to denote the i-th component of $\hat{\mathbf{v}}$. The iterative process commences by setting $\hat{\mathbf{v}}_0 = \mathbf{0}$. A sequence of step sizes $\{\alpha_t\}$ is selected beforehand. At time t, the following computations and updates are carried out.

$$\delta_{t+1} = W_{t+1} + \gamma \hat{\mathbf{v}}_t(X_{t+1}) - \hat{\mathbf{v}}_t(X_t), \tag{29}$$

$$\hat{\mathbf{v}}_{t+1}(X_{t+1}) = \hat{\mathbf{v}}_t(X_{t+1}) + \alpha_t \delta_{t+1}, \tag{30}$$

$$\hat{\mathbf{v}}_{t+1}(x_j) = \hat{\mathbf{v}}_t(x_j) \text{ for } x_j \neq X_{t+1}. \tag{31}$$

In other words, at time $t+1$, if $X_{t+1} = x_i$, then the i-th component of the estimated value vector $\hat{\mathbf{v}}$ is updated by adding $\alpha_t \delta_{t+1}$, while all other components remain the same.

C. Q-Iteration Methods

One of the significant advances in RL is that, instead of learning the value function, one learns the "Q" function, defined next. Note that this technique is introduced in [8].

$$Q_\pi(x_i, u) = R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u V_\pi(x_j). \tag{32}$$

The optimal function Q^* is given by

$$Q^*(x_i, u) := R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u V^*(x_j). \tag{33}$$

Recall that the optimal value function V^* satisfies the Bellman equation (17), recalled here for the reader's convenience:

$$V^*(x_i) = \max_{u \in \mathcal{U}} \left[R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u V^*(x_j) \right].$$

Therefore it is clear that

$$V(x_i) = \max_{u \in \mathcal{U}} Q^*(x_i, u).$$

In view of this, (33) can be rewritten as follows, which can be thought of as the Bellman equation for Q^*.

$$Q^*(x_i, u) := R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u \max_{w \in \mathcal{U}} Q^*(x_j, w). \tag{34}$$

Similarly, once we know Q^*, it is easy to determine the optimal policy, via

$$\pi^*(x_i) = \arg\max_{u \in \mathcal{U}} Q^*(x_i, u). \tag{35}$$

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on November 16,2022 at 07:29:00 UTC from IEEE Xplore. Restrictions apply.

Recall the "value iteration" map T defined in (19) for computing the optimal value function recursively, namely

$$(T\mathbf{v})_i := \max_{u \in \mathcal{U}} \left[R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u v_j \right].$$

Not surprisingly, there is an analous iterative map for computing Q as well. Define $F : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ by

$$(FQ)(x_i, u) := R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u \max_{w \in \mathcal{U}} Q(x_j, w). \quad (36)$$

With this definition, we can state several useful results.

Theorem 4: The map F is monotone and is a contraction. Therefore, for any initial guess Q_0, the sequence $\{F^k Q_0\}$ converges to Q^*.

Now let us turn to the question of *estimating* FQ. As before, suppose that a sample path $\{(X_t, U_t, W_t\}_{t \geq 0}$ is observed. Then the quantity

$$R(X_t, U_t) + \gamma \max_{w \in \mathcal{U}} Q(X_{t+1}, w)$$

is an unbiased sample of $FQ(X_t, U_t)$. In contrast,

$$\max_{u \in \mathcal{U}} [R(X_t, u) + \gamma V(X_{t+1})]$$

is *not* an unbiased sample of $TV(X_t)$. On the basis of this reasoning, it is possible to propose an updating rule for learning Q. Choose a sequence of "step sizes" $\{\alpha_t\}_{t \geq 0}$. Start with some initial guess $Q_0(\cdot, \cdot)$ on $\mathcal{X} \times \mathcal{U}$. Then, at time t, if $X_t = x_i$, then update $Q_{t+1}(x_i, U_t)$ via

$$\begin{aligned} Q_{t+1}(x_i, U_t) &:= (1 - \alpha_t) Q_t(x_i, U_t) \\ &+ \alpha_t \left[W_t + \gamma \max_{w \in \mathcal{U}} Q_t(X_{t+1}, w) \right], \end{aligned} \quad (37)$$

and do not update any other $Q_{t+1}(x_j, u)$ for $(x_j, u) \neq (x_i, U_t)$.

For this updating rule, it is possible to prove the following convergence result, which is a type of stochastic approximation. introduced in [9]. See [10] for a survey of stochastic approximation.

Theorem 5: Suppose that each pair in $\mathcal{X} \times \mathcal{U}$ is samples infinitely often, and that for each $(x_i, u) \in \mathcal{X} \times \mathcal{U}$, the following conditions hold:

$$\sum_{t : (X_t, U_t) = (x_i, u)} \alpha_t = \infty, \quad (38)$$

$$\sum_{t : (X_t, U_t) = (x_i, u)} \alpha_t^2 < \infty. \quad (39)$$

Then $Q_t \to Q^*$ almost surely.

IV. Some Recent Advances and Future Areas

In this section, we briefly discuss two of the many advances that have been made in recent years.

A. Empirical Dynamic Programming

Note that it is possible to rewrite the Bellman equation (17) as

$$V^*(x_i) = \max_{u \in \mathcal{U}} \{ R(x_i, u) + \gamma E[V^*(X_{t+1}) | x_i, u] \}. \quad (40)$$

Similarly, it is possible to rewrite the Bellman (contraction) operator as

$$(T\mathbf{v})_i = \max_{u \in \mathcal{U}} \{ R(x_i, u) + \gamma E[V^*(X_{t+1}) | x_i, u] \}. \quad (41)$$

Here and below, we use the shorthand

$$E[\cdot | x_i, u] = E[\cdot | X_t = x_i, U_t = u].$$

The advantage of these reformulations is that they apply even when $\mathcal{X}, \mathcal{U}$ are infinite sets. However, computing the expected value, even when $\mathcal{X}$ is finite, can be time-consuming. Instead, one can approximate the expected value with an **empirical average**, where, starting with $X_t = x_i, U_t = u$, some samples for X_{t+1} are generated, and the value of V over these samples is taken as an approximation for the expected value. This is the approach adopted in [5], [6].

B. Zap Q-Learning

A recent set of papers proposes a new variant of Q-learning called "Zap" Q-learning. In addition to guaranteeing the asymptotic almost sure convergence of an estimated Q-function to Q^*, this variant also leads to the asymptotic covariance being optimal.

To begin, recall that the Bellman equation for the Q-function is given in (34), as follows:

$$Q^*(x_i, u) := R(x_i, u) + \gamma \sum_{j \in [n]} a_{ij}^u \max_{w \in \mathcal{U}} Q^*(x_j, w). \quad (42)$$

Observe that Q^* is a function mapping $\mathcal{X} \times \mathcal{U}$ into $\mathbb{R}$. Therefore it is possible to think of Q^* as a $|\mathcal{X}| \cdot |\mathcal{U}|$-dimensional vector. However, it is more economical to *choose a set of basis functions* $\psi_l, l \in [d]$ where each $\psi_l : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, and to approximate Q^* as a linear combination, in the form

$$Q^{\boldsymbol{\theta}}(x_i, u) = \sum_{l \in [d]} \theta_l \psi(x_i, u) = \boldsymbol{\theta}^\top \boldsymbol{\psi}(x_i, u). \quad (43)$$

Obviously this approach makes sense only if $d \ll |\mathcal{X}| \cdot |\mathcal{U}|$. Having indicated that both $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ are d-dimensional vectors, we now dispense with the bold-face, and use just θ and ψ. Define the **Bellman error** as

$$\begin{aligned} \mathcal{B}^\theta(x_i, u) &:= -Q^\theta(x_i, u) + R(x_i, u) \\ &+ \gamma E\left[\max_{w \in \mathcal{U}} Q^\theta(X_{t+1}, w) | x_i, u \right], \end{aligned} \quad (44)$$

where, as before, we use the shorthand

$$E[\cdot | x_i, u] = E[\cdot | X_t = x_i, U_t = u].$$

Clearly Q^θ is optimal if and only its associated Bellman error is the zero function. The original stochastic approximation approach [9], [10] addresses the problem of finding the zeros of a function with noisy measurements. This suggests that an

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on November 16,2022 at 07:29:00 UTC from IEEE Xplore. Restrictions apply.

approach similar to stochastic approximation may be used to find Q^θ so as to make the Bellman error equal to zero.

Now we present the Zap Q-learning algorithm. The original paper is [11], while a nice survey is presented in [12]. Define

$$\begin{aligned} \tilde{\mathcal{B}}^\theta_{t+1} &:= -Q^\theta(X_t, U_t) + R(X_t, U_t) \\ &+ \gamma E\left[\max_{w \in \mathcal{U}} Q^\theta(X_{t+1}, w), h\right], \end{aligned} \qquad (45)$$

where h is the stationary distribution of the Markov chain. Choose a constant $\lambda \in (0, 1)$, and define sequences

$$\alpha_t = \frac{1}{t+1}, \beta_t = \frac{1}{(t+1)^\rho}, \rho \in (0.5, 1).$$

Note that more general choices for these sequences are possible; see [12]. Next, define

$$\theta_{t+1} = \theta_t - \alpha_t \hat{M}^{-1}_{t+1} d_{t+1} z_t,$$

$$d_{t+1} = \tilde{\mathcal{B}}^{\theta_t}_{t+1},$$

$$z_{t+1} = \lambda\gamma z_t + \psi(X_t, U_t),$$

$$\hat{M}_{t+1} = \hat{M}_t + \beta_t(M_{t+1} - \hat{M}_t),$$

$$M_{t+1} = z_t[\gamma\psi(X_{t+1}, \phi_t(X_{t+1})) - \psi(X_t, U_t)]^\top,$$

where

$$\phi_t(x_i) = \arg\max_{w \in \mathcal{U}} Q^{\theta_t}(x_i, w).$$

With the above definitions, it is shown in, for example, [12, Theorem 2.2], that the sequence $\{Q^{\theta_t}\}$ converges amost surely to Q^*. Moreover, the resulting asymptotic covariance is optimal.

C. Some Areas for Future Research

Given a sample path of a Markov Decision Process with unknown parameters, in principle it is possible to learn these parameters on the basis of observations. Therefore a promising area for future research is to explore the connections between Reinforcement Learning, and statistical learning theory as studied in, for example, [13], [14]. This approach is also reminiscent of what used to be called "Direct Adaptive Control" during the 1960s. It is however too early to say whether this approach offers any advantages over existing methods.

References

[1] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[2] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley, 2005.

[3] C. Szepesvári, *Algorithms for Reinforcement Learning*. Morgan and Claypool, 2010.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction (Second Edition)*. MIT Press, 2018.

[5] W. B. Haskell, R. Jain, and D. Kalathil, "Empirical dynamic programming," *Mathematics of Operations Research*, vol. 41, no. 2, pp. 402–429, 2016.

[6] W. B. Haskell, R. Jain, H. Sharma, and P. Yu, "A universal empirical dynamic programming algorithm for continuous state MDPs," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 115–129, January 2020.

[7] M. Vidyasagar, *Hidden Markov Processes: Theory and Applications to Biology*. Princeton University Press, 2014.

[8] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[9] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[10] T. L. Lai, "Stochastic pproximation (invited paper)," *The Annals of Statistics*, vol. 31, no. 2, pp. 391–406, 2003.

[11] A. M. Devraj and S. Meyn, "Zap Q-learning," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, pp. 2235–2244.

[12] A. M. Devraj, A. Busić, and S. Meyn, "Zap Q-learning– a users guide," in *Proceedings of the 2019 Fifth Indian Control Conference (ICC)*, 2019, pp. 10–15.

[13] V. N. Vapnik, *Statistical Learning Theory*. John Wiley, 1998.

[14] M. Vidyasagar, *Learning and Generalization: With Applications to Neural Networks and Control Systems*. Springer-Verlag, 2002.

Authorized licensed use limited to: Indian Institute of Technology Hyderabad. Downloaded on November 16,2022 at 07:29:00 UTC from IEEE Xplore. Restrictions apply.