Machine Learning Reading Group: Reinforcement Learning Notes

Nicholas Denis

April 14, 2019

Contents

1 Introduction and overview **3**

2 How to make a decision: stochastic multi armed bandit **3**
2.1 Upper Confidence Bounds (UCB) Algorithm (Auer et al. (2002) 3

3 How to make a sequence of decisions: RL and Markov Decision Processes **6**
3.1 Markov Decision Processes 6
3.1.1 Returns 7
3.1.2 Policies 7
3.2 Value Functions 7
3.2.1 Bellman Equations 8
3.3 Optimality 9
3.3.1 Finite Horizon Setting 9
3.3.2 Infinite Horizon Setting 10
3.3.3 Bellman Operators 11

4 Solution Approaches I: Dynamic Programming **13**
4.1 Value Iteration 13
4.2 Policy Evaluation 14
4.3 Policy Improvement 14
4.4 Policy Iteration 15

5 Solution Approaches II: RL **15**
5.1 Monte Carlo Methods 15
5.2 Temporal Difference Learning: Q-learning 15
5.2.1 Incremental updates and tracking non-stationarity 15
5.3 PAC-MDP 16

6 Deep RL **16**
6.1 Deep Q-networks (DQN) 16
6.2 Policy Gradients 16
6.3 Actor Critic 16

1 Introduction and overview

2 How to make a decision: stochastic multi armed bandit

We begin by setting the stage of the stochastic MAB with some definitions and notation.

- $\exists k < \infty$ arms (choices)
- Arm i has unknown reward distribution $\mathcal{R}_i$ (e.g. $r_{i,t} \sim \mathcal{R}_i$), with $\mathbb{E}[r_{i,t}] = \mu_i$
- $\exists \mu^* = \max_{j \in \{1,2,...,k\}} \mu_j$
- Reward gap $\Delta_i := \mu^* - \mu_i$
- Number of pulls of arm i after time t, $T_i(t) = \sum_{k=1}^{t} \mathbb{1}[a_k = i]$
- Empirical mean reward of arm i after time t, $\hat{\mu}_{i,t}$
- Pseudo-regret: $R(T) = T\mu^* - \mathbb{E}\big[\sum_{t=1}^{T} r_{i,t}\big] = \sum_{i=1}^{k} \mathbb{E}[T_i(T)\Delta_i]$

Next, we introduce our trusty friend for distribution free analysis: The Hoeffding inequality.

Lemma 1. *(Hoeffding's Inequality)Let $Z_i, ..., Z_n$ be iid random variables s.t. $Z_i \in [a_i, b_i]$ almost surely. Then $\forall \epsilon > 0$, $P\Big(|\frac{1}{n}\sum_{i=1}^{n} Z_i - \mathbb{E}[Z]| \geq \epsilon\Big) \leq 2e^{-\frac{2\epsilon^2 n^2}{\sum_i (b_i - a_i)^2}}$ Moreover, if $a_i = 0, b_i = 1$, we get bounds of $2e^{-2n\epsilon^2}$, for $\alpha > 2$.*

2.1 Upper Confidence Bounds (UCB) Algorithm (Auer et al. (2002)

UCB is a simple algorithm that has stood the test of time, and is honored with many future algorithms use as their base. The idea is that the algorithm first samples each of the k arms once each, which acts to initialize the estimated mean reward of each arm $\hat{\mu}_{i,t}$. Then, after having sampled each arm exactly once,

$$UCB(t) := \underset{i \in [k]}{arg\,max}\ \hat{\mu}_{i,T_i(t)} + \sqrt{\frac{\alpha \log(t)}{2T_i(t)}}$$

Theorem 2. *Regret bound for the UCB algorithm after $T \geq 1$ is*

$$R(T) \leq \sum_{i:\Delta_i > 0} 4\alpha \Delta_i^{-1} log(T) + \frac{2\alpha}{\alpha - 1}\Delta_i$$

Proof. WLOG we assume that the first arm is optimal. Hence arm $i \neq 1$ is played if either arms 1 or i have been sampled insufficiently to distinguish their means, or the the upper confidence bound fails for either arm. Define events:

- A_t - $\hat{\mu}_{i,T_i(t)} \leq \mu_i + \sqrt{\frac{\alpha \log(t)}{2T_i(t)}}$
- B_t - $\hat{\mu}_{1,T_1(t)} \geq \mu_1 - + \sqrt{\frac{\alpha \log(t)}{2T_1(t)}}$

We bound the probabilities of these events not occurring. For A_t to fail, we have

$$\hat{\mu}_{i,T_i(t)} - \mu_i > + \sqrt{\frac{\alpha \log(t)}{2T_i(t)}}$$

Then using Hoeffding inequality we see that

$$P(\hat{\mu}_{i,T_i(t)} - \mu_i > \epsilon) \leq e^{-2t\epsilon^2}$$

For $\epsilon = \sqrt{\frac{\alpha \log(t)}{2T_i(t)}}$

$$\begin{aligned} P\Big(\hat{\mu}_{i,T_i(t)} - \mu_i > \sqrt{\frac{\alpha \log(t)}{2T_i(t)}}\Big) &\leq e^{\frac{-2t\alpha \log(t)}{2T_i(T)}} \\ &= e^{\frac{-t\alpha \log(t)}{T_i(T)}} \\ &\leq e^{\frac{-t\alpha \log(t)}{t}} \\ &= e^{-\alpha \log(t)} \\ &= t^{-\alpha} \end{aligned}$$

The same is true for the event B_t Now suppose a suboptimal arm i is pulled when both A_t, B_t hold, but the upper confidence bounds of arm i exceeds the upper confidence bounds of arm 1. Hence,

$$\hat{\mu}_{i,T_i(t)} + \sqrt{\frac{\alpha \log(t)}{2T_i(t)}} > \hat{\mu}_{1,T_1(t)} + \sqrt{\frac{\alpha \log(t)}{2T_1(t)}}$$

Since A_t is true, then

$$\mu_i + 2\sqrt{\frac{\alpha \log(t)}{2T_i(t)}} > \hat{\mu}_{1,T_1(t)} + \sqrt{\frac{\alpha \log(t)}{2T_1(t)}}$$

Since B_t is true, then

$$\hat{\mu}_{1,T_1(t)} + \sqrt{\frac{\alpha \log(t)}{2T_1(t)}} \geq \mu_1$$

and together we have

$$\mu_i + 2\sqrt{\frac{\alpha \log(t)}{2T_i(t)}} > \mu_1$$

Which we can express as

$$\sqrt{\frac{\alpha \log(t)}{T_i(t)}} \geq \frac{\mu_1 - \mu_i}{2} = \frac{\Delta_i}{2}$$

Hence, we arrive at

$$\begin{aligned} T_i(t) &\leq 4\Delta_i^{-2} \alpha \log(t) \\ &\leq 4\Delta_i^{-2} \alpha \log(T) \end{aligned}$$

Hence, combining all 3 possible sources of error (pulling the wrong arm), we have:

$$\begin{aligned} \mathbb{E}[T_i(t)] &= \sum_{t=1}^{T} \mathbb{E}[\mathbb{1}[arm_t = i]] \\ &\leq 4\alpha\Delta_i^{-2}\log(T) + \sum_{t=1}^{T} \mathbb{E}\Big[\mathbb{1}[A_t^c \cup B_t^c]\Big] \\ &\leq 4\alpha\Delta_i^{-2}\log(T) + \sum_{t=1}^{T} \Big(\mathbb{E}[\mathbb{1}[A_t^c] + \mathbb{E}[\mathbb{1}[B_t^c]]\Big) \\ &\leq 4\alpha\Delta_i^{-2}\log(T) + \sum_{t=1}^{T} (t^{-\alpha} + t^{-\alpha}) \\ &= 4\alpha\Delta_i^{-2}\log(T) + 2\sum_{t=1}^{T} t^{-\alpha} \end{aligned}$$

We note that $\sum_{t=1}^{T} t^{-\alpha} \leq 1 + \int_1^{\infty} x^{-\alpha} dx = 1 + \frac{-1}{1-\alpha} = \frac{-\alpha}{1-\alpha}$, but with $\alpha > 2$, we have $\mathbb{E}[T_i(t)] \leq 4\alpha\Delta_i^{-2}\log(T) + \frac{2\alpha}{\alpha-1}$. Then summing over all suboptimal arms provides the result. □

Moreover, (Lai-Robins (1985)), show that regret is lower bounded by $\log(T)$. Hence any algorithm achieiving $\mathcal{O}\log(T)$ regret is considered efficient.

3 How to make a sequence of decisions: RL and Markov Decision Processes

Wheras an agent faced with a MAB problem is sequentially faces the exact same decision over and over in perpetuity, an agent within an RL problem finds itself within an *environment*, where at each time point t, the agent is in a state s_t, and must take an action a_t. Upon taking an action, the environment evolves according to the environment dynamics, and the agent receives both a reward r_t and its state is updated to s_{t+1}. This process continues. Generally speaking, the goal of the agent is to learn from its experiences to arrive at a *policy* or action selection strategy/behaviour so as to maximize the expected cumulative sum of rewards. That is, to make a sequence of decisions that leads to the best possible future life. In order to deal with this problem, we will be relying on representing this problem within a specific framework: Markov Decision Processes. We next introduce how to model the environment, look at what constitutes a sufficient representation from which we base the policy (decision making process), and introduce key players in the MDP and RL framework: value functions.

3.1 Markov Decision Processes

A MDP, $\mathcal{M}$, is a tuple

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, R, P, \gamma, p_0 \rangle$$

where

- $\mathcal{S}$ is the state space. Here we assume $|\mathcal{S}| < \infty$.
- $\mathcal{A}$ is the set of actions at the disposal of the agent. Here we assume $|\mathcal{A}| < \infty$.
- P is the transition kernel of the environment. It encodes the state transition dynamics. Since we are considering finite state and action spaces, $\forall a \in \mathcal{A}$, $\exists P_a \in \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$, where $P(s'|s,a)$ is the probabilty of the environment transitioning to state s' when the action a is taken from state s. Hence, $\exists\ |\mathcal{A}|$ transition matrices.
- The Markov property of the environment is such that $P(s_{t+1} = s|s_t, a_t, s_{t-1}, ..., a_1, s_1) = P(s_{t+1} = s|s_t, a_t)$.
- R is the reward function. The MDP framework is flexible and allow for
  - $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
  - $R : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$
  - $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$
  - Note: typically it is assumed $R(.) \in [R_{min}, R_{max}]$
- R, P are assumed to be time homegenous.

- $\gamma \in [0, 1]$, is the discount factor.
- p_0 is the initial state distribution (e.g. $p_0 \in [0, 1]^{\mathcal{S}}$, s.t. $\sum_{s \in \mathcal{S}} p_0(s) = 1$)

3.1.1 Returns

We will use the following notation,

- $R_t := r_t + r_{t+1} + ... + r_{t+T}$. R_t is a random variable representing the sum of future rewards up until a *horizon* of length T into the future. This is often used for *episodic* or *finize horizon* settings.
- $R_t := r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$. This is a similar representation, however for infinite horizon settings. Since $\forall t$, $r_t \in [R_{min}, R_{max}]$, it may be that this sum diverges to ∞. Hence, to make computations meaningful, a discount factor $\gamma \in [0, 1)$ is used, so that $R_t < \infty$ for the infinite horizon settings. There are utility theory/economic motivations, however in reality it is simply a tool to turn an infinite sum into a finite sum.
- We may use a unified view of:
- $R_t = \sum_{k=0}^{T} \gamma^k r_{t+k}$, where we allow $\gamma \in [0, 1]$ and $T \leq \infty$, to capture both settings.

3.1.2 Policies

The goal of the agent is to solve for a policy within some class of policies, $\pi^* \in \Pi$, such that $\mathbb{E}_\pi[R_t]$ is maximized. A policy (class) can be a function of the current state only (Markovian),e.g. $\pi : \mathcal{S} \to \mathcal{A}$, or a function of the entire history of the agent, $\pi : \mathcal{H}_t \to \mathcal{A}$, where $\mathcal{H}_1 := \mathcal{S}_1$, and $\mathcal{H}_t := \mathcal{H}_{t-1} \times (\mathcal{A}_{t-1} \times \mathcal{S}_t)$. A policy (class) can be deterministic (as stated above) or randomized/stochastic: $\pi : \mathcal{S} \to \Delta(A)$, where $\pi(a|s) \in [0, 1]$, $\sum_{a \in \mathcal{A}} \pi(a|s) = 1$.

We note that $\Pi^{MD} \subset \Pi^{HD} \subset \Pi^{HR}$, and $\Pi^{MD} \subset \Pi^{MR} \subset \Pi^{HR}$.

3.2 Value Functions

We now turn to key players in our story: value functions. There are three types of value functions, two of which we introduce here. The first represents the value of a state, with respect to a given policy. It measures how good it is to be in that state, given the policy (action selection strategy) of the agent. The second is the value of a state-action pair, and measures how good it is to be in a given state *and* take a specific action.

- For fixed π, the state value function: $V^\pi(s) := \mathbb{E}_\pi[R_t|s_t = s]$.
- For fixed π, the state-action value function: $Q^\pi(s, a) := \mathbb{E}_\pi[R_t|s_t = s, a_t = a]$.

- We define $V^* = \max_{\pi} \ V^\pi$, and $Q^* = \max_{\pi} \ Q^\pi$.
- For fixed π, we view $V^\pi \in \mathbb{R}^{\mathcal{S}} =: \mathscr{V}$, $Q^\pi \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}} =: \mathscr{Q}$.

3.2.1 Bellman Equations

Now with our mathematical machinery, we can take a better look into what V^π is:

$$\begin{aligned}
V^\pi(s) &= \mathbb{E}_\pi[R_t|s_t = s] \\
&= \mathbb{E}_\pi\Big[\sum_{k=0}^{\infty}\gamma^k r_{t+k}|s_t = s\Big] \\
&= \mathbb{E}_\pi\Big[r_t + \gamma\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s\Big] \\
&= \sum_{a\in\mathcal{A}}\pi(a|s)\sum_{s'\in\mathcal{S}}P(s'|s,a)\Big[R(s,a,s') + \gamma\mathbb{E}_\pi[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s]\Big] \\
&= \sum_{a\in\mathcal{A}}\pi(a|s)\sum_{s'\in\mathcal{S}}P(s'|s,a)\Big[R(s,a,s') + \gamma V^\pi(s')\Big]
\end{aligned}$$

The final line above is the *Bellman Evaluation Equation* for V^π. Similarly,

$$Q^*(s,a) = \mathbb{E}_\pi[r_t + \gamma V^*(s_{t+1})|s_t = s, a_t = a]$$

V and Q are related in that $V^*(s) = \max_{a\in\mathcal{A}} \ Q^*(s,a)$

For V^* we have

$$\begin{aligned}
V^\pi(s) &= \max_{a\in\mathcal{A}} \ Q^{\pi^*}(s,a) \\
&= \max_{a\in\mathcal{A}} \ \mathbb{E}_{\pi^*}\Big[\sum_{k=0}^{\infty}\gamma^k r_{t+k}|s_t = s\Big] \\
&= \max_{a\in\mathcal{A}} \ \mathbb{E}_{\pi^*}\Big[r_t + \gamma\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s\Big] \\
&= \max_{a\in\mathcal{A}} \ \mathbb{E}\Big[r_t + \gamma V^*(s_{t+1})|s_t = s, a_t = a\Big] \\
&= \max_{a\in\mathcal{A}} \ \sum_{s'\in\mathcal{S}}P(s'|s,a)[R(s,a,s') + \gamma V^*(s')]
\end{aligned}$$

This last line above is known as the *Bellman Optimality Equation.* Moreover, for Q^* we have

$$Q^*(s,a) = \mathbb{E}[r_t + \gamma \max_{a' \in \mathcal{A}} \; Q^*(s',a') | s_t = s, a_t = a]$$
$$= \sum_{s' \in \mathcal{S}} P(s'|s,a)[R(s,a,s') + \gamma \max_{a' \in \mathcal{A}} \; Q^*(s',a')]$$

which is the Bellman optimality equation for Q^*.

3.3 Optimality

Now that we have defined some key players, it is time to solve for π^*, and $V^* := V^{\pi^*}$, the optimal policies and value functions. We would ultimately like to solve for $\pi^* \in \Pi^{HR}$, the optimal policy in the largest and most expressive policy class. As it turns out, we will show that under extremely soft conditions, we will be able to optimize over the smaller policy class Π^{MD} and the supremum over this policy class is *i. realized*, and *ii. matches that supremum value function over* Π^{HR}. Hence, we can find the "best" policy and value function while limiting ourselves to deterministic Markovian policies. We will consider first the finite horizon setting, then the infinite horizon setting.

3.3.1 Finite Horizon Setting

The optimality equations for $V^*(h_t)$ are:

$$V^*(h_t) = \sup_{a \in \mathcal{A}} \; \mathbb{E}[r_t(s_t, a) + V^*(h_{t+1})] \tag{3.3.1}$$

for $t = 1, 2, ..., T-1$, and for $t = T$ we have $V^*(h_T) = r_T(s_T)$ (3.3.2)

We begin by showing if there is a history dependent policy that realizes the above equations, then this policy has the same value as the optimal history randomized policy.

Theorem 3. *(Puterman, Theorem 4.3.3 p. 86) Suppose $V_t^*, t = 1, 2, ..., T$ are solutions to (3.3.1, 3.3.2), and that the policy $\pi^* \in \Pi^{HD}$ satisfies:*

$$V^*(h_t) = \max_{a \in \mathcal{A}} \; \mathbb{E}[r_t(s_t, a) + V^*(h_{t+1})]$$

for $t = 1, 2, ..., T-1$. Then

- *a. $\forall t$, $V^{\pi^*}(h_t) = V^*(h_t)$, $h_t \in \mathcal{H}_t$.*
- *b. π^* is an optimal policy and $V^{\pi^*}(s_T) = V^*(s_T)$, $\forall s \in \mathcal{S}$.*

Hence, this theorem states that if $\exists$ a policy in Π^{HD} s.t. at each step $\exists$ an action that realizes the supremum, then this policy is optimal and has the same value as the optimal policy within Π^{HR}. Moreover, there is another theorem that states that whenever there exists *any* policy where the actions taken realize the supremum (e.g. max = sup), then there exists a deterministic history-dependent policy that is optimal. Next we show that for finite horizon problems, the optimal policy is always Markovian.

Theorem 4. *(Puterman, Thereom 4.4.2 pg 89)Let V^* be the solution of (3.3.1, 3.3.2), then*

- *a. $\forall t = 1, 2, ..., T$, $V^*(h_t)$ depends only on h_t via s_t.*
- *b. If $\exists\ a \in \mathcal{A}$ such that Theorem 3 (previous theorem) holds true $\forall s_t \in \mathcal{S}, t \in [T]$, then $\exists\ \pi^* \in \Pi^{MD}$.*

Proof. For (a) we prove by reverse induction. Note $V^*(h_T) = \mathbb{E}(r(s_T)), \forall\, h_{T-1} \in \mathcal{H}_{T-1}$. Hence we have a base case for induction. Assume (a) is true for $k = T, T-1, ..., t+1$. Then

$$V^*(h_t) = \sup_{a \in \mathcal{A}} \ \mathbb{E}\big[r(s_t, a) + V^*(h_{t+1})\big]$$

which, by the induction hypothesis yields

$$V^*(h_t) = \sup_{a \in \mathcal{A}} \ \mathbb{E}\big[r(s_t, a) + V^*(s_{t+1})\big]$$

Since the above quantity depends on h_t only via s_t, we see that (a) holds $\forall t$. Hence, V^* is Markovian. Moreover, (b) is essentially a Corollary that follows from the previous theorem. □

We now want to consider when the supremum is obtained. This almost trivial, but we state anyway for the assumptions of our work.

Proposition 5. *Assume $\mathcal{S}$ is discrete and $\mathcal{A}$ is finite. Then $\exists\ \pi \in \Pi^{MD}$ that is optimal.*

Hence, it is true that $V^*(s) = \sup_{\pi \in \Pi^{HR}} V^\pi(s) = \sup_{\pi \in \Pi^{MD}} V^\pi(s), \forall s \in \mathcal{S}$.

3.3.2 Infinite Horizon Setting

Moving forward, we assume the following:

- Stationary rewards and transition probabilities
- Bounded rewards. e.g. $\exists M < \infty$ s.t. $|R(s, a, s')| \leq M$, $\forall (s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$.
- $\gamma \in [0, 1)$
- $|\mathcal{S}| \leq |\mathbb{N}|$

We will now make use of Banach spaces and fixed point theorems of contraction mappings to motivate the use of Bellman operators for performing value iteration and policy iteration. First, we note that $\mathscr{V} := \mathbb{R}^{\mathcal{S}}$ equipped with $||.||_\infty$ is a Banach space. Moreover, with bounded rewards, $\forall \pi$, $\forall s \in \mathcal{S}$, $V^\pi(s) \leq \sum_{k=0}^{\infty} \gamma^k M = \frac{M}{1-\gamma}$. Hence, $||V^\pi||_\infty \leq \frac{M}{1-\gamma}$. So the space of value functions are bounded in some ball.

Moreover, we may use the following compact notation $r(s)$ to mean either $r(s,a)$, or $r(s,a,s')$, when appropriate. Even more compactly, we may represent $r = r(s)$, or $r_a = r(.,a)$, and P_a to be the $|\mathcal{S}| \times |\mathcal{S}|$ matrix. r_a is the reward vector in $\mathbb{R}^{\mathcal{S}}$ (or $\mathbb{R}^{\mathcal{S}} \times \mathcal{S}$ when appropriate), and P_a the transition probability matrix. We may let $v \in \mathbb{R}^{\mathcal{S}}$, hence $P_a v \in \mathbb{R}^{\mathcal{S}}$.

Lemma 6. *(Puterman, 5.6.1, p.138) Let $\mathcal{S}$ be discrete, $|r(s,a)| \leq M < \infty$, $\forall a \in \mathcal{A}$, $s \in \mathcal{S}$, and let $\gamma \in [0,1]$. Then, $\forall v \in \mathscr{V}$, $\pi \in \Pi^{MR}$, $\mathbb{E}[r_\pi + \gamma P_\pi v] = v' \in \mathscr{V}$.*

Theorem 7. *(Puterman, Theorem 5.5.3 p.137) Suppose $\pi \in \Pi^{HR}$, then $\forall s \in \mathcal{S}$, $\exists \pi' \in \Pi^{MR}$ s.t. $V_\gamma^{\pi'}(s) = V_\gamma^\pi(s)$*

Hence, as a consequence of this theorem, we do not need to consider $\pi \in \Pi^{HR}$, since

$$V_\gamma^*(s) := \sup_{\pi \in \Pi^{HR}} V_\gamma^\pi(s) = \sup_{\pi \in \Pi^{MR}} V_\gamma^\pi(s)$$

3.3.3 Bellman Operators

Bellman Optimality Operator. Definition (Bellman Optimality Operator): Let $V \in (\mathscr{V}, ||\ ||_\infty)$. Define $B^* : \mathscr{V} \to \mathscr{V}$ pointwise, where

$$B^*V(s) := \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a)[R(s,a,s') + \gamma V(s')].$$

Then B^* is the Bellman optimality operator. Note that B^* is a bounded nonlinear operator. Note: Using the vector notation above,

$$B^*V(s) := \max_{a \in \mathcal{A}} \{r_a + \gamma P_a V(s')\}.$$

B^* is the Bellman optimality operator, and coincides with the recursive form of the optimal value function $V^* \in \mathscr{V}$. Normally B^* is defined using the *supremum*, and not the maximum. However, since we are considering $|\mathcal{A}| < \infty$, then the supremum is attained, and hence the maximum is equivalent. As well, we should note that using vector notation, this is equivalent to stating:

Hence, together, it is sufficient to find the optimal policy π^* within the class or deterministic Markovian policies, when the environment follows a MDP structure.

$$B^*V(s) := \max_{\pi \in \Pi^{MD}} \{r_\pi + \gamma P_\pi V(s')\}.$$

Since,

Proposition 8. *(Puterman, 6.2.1, p.147)* $\forall v \in \mathscr{V}$, *and* $\gamma \in [0,1]$

$$\sup_{\pi \in \Pi^{MD}} \{r_\pi + \gamma P_\pi v\} = \sup_{\pi \in \Pi^{MR}} \{r_\pi + \gamma P_\pi v\}.$$

Proof. Since $\Pi^{MD} \subset \Pi^{MR}$ it remains to show that $\sup_{\pi \in \Pi^{MD}} \{r_\pi + \gamma P_\pi v\} \geq \sup_{\pi \in \Pi^{MR}} \{r_\pi + \gamma P_\pi v\}$. Let $v \in \mathscr{V}$, $\delta \in \Pi^{MR}$, $s \in \mathcal{S}$ and see that

$$\begin{aligned}
\sup_{\pi \in \Pi^{MD}} \{r_\pi + \gamma P_\pi v\} &= \sup_{a \in \mathcal{A}} \left\{ r(s,a) + \sum_{s'} \gamma P(s'|s,a) V_\gamma(s') \right\} \\
&= \sum_{a' \in \mathcal{A}} \delta(a'|s) \sup_{a \in \mathcal{A}} \left\{ r(s,a) + \sum_{s'} \gamma P(s'|s,a) V_\gamma(s') \right\} \\
&\geq \sum_{a \in \mathcal{A}} \delta(a|s) \left\{ r(s,a) + \sum_{s'} \gamma P(s'|s,a) V_\gamma^\delta(s') \right\} \\
&= r_\delta + \gamma P_\delta v
\end{aligned}$$

□

Theorem 9. *(Banach fixed-point theorem)Suppose* U *is a Banach space and* $T : U \to U$ *is a contraction mapping (e.g.* $\exists \kappa < 1$ *s.t.* $\forall u, v \in U$, $||Tu - Tv|| \leq \kappa ||u - v||$*). Then*

- *i)* $\exists!$ $v^* \in U$ *s.t.* $Tv^* = v^*$*; and*
- *ii)* $\forall v_0 \in U$*, the sequence* $\{v_n\}$ *given by* $v_{n+1} = Tv_n = T^{n+1}v_0$ *converges to* v^**.*

Theorem 10. *Let* $\gamma \in [0,1)$*. Then* B^* *is a contraction mapping on* $\mathscr{V}$*.*

Proof. Let $U, V \in \mathscr{V}$, and fix $s \in \mathcal{S}$. WLOG assume $B^*U(s) \leq B^*V(s)$. Let

$a^* \in \underset{a \in \mathcal{A}}{arg\,max} \ \sum_{s' \in \mathcal{S}} P(s'|s,a)[R(s,a,s') + \gamma V(s')]$. Then,

$$
\begin{aligned}
0 &\leq B^*V(s) - B^*U(s) \\
&\leq \sum_{s' \in \mathcal{S}} P(s'|s,a^*)[R(s,a^*,s') + \gamma V(s')] - \sum_{s' \in \mathcal{S}} P(s'|s,a^*)[R(s,a^*,s') + \gamma U(s')] \\
&= \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a^*)[V(s') - U(s')] \\
&\leq \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a^*)||V - U||_\infty \\
&= \gamma ||V - U||_\infty
\end{aligned}
$$

Repeating the argument for the case that $B^*V(s) \leq B^*U(s)$ implies that

$$
\begin{aligned}
|B^*V(s) - B^*U(s)| \leq \gamma ||V - U||_\infty, \text{hence, taking sup over s yields} \\
||B^*V(s) - B^*U(s)||_\infty \leq \gamma ||V - U||_\infty
\end{aligned}
$$

□

Theorem 11. *(Puterman, Theorem 6.2.5, p151) Suppose $\gamma \in [0,1)$, $\mathcal{S}$ countable, $r(s,a)$ bounded. Then $\exists\ a\ V^* \in \mathscr{V}$ satisfying $B^*V^* = V^*$ and V^* is unique (the only element satisfying this).*

Hence, with just a little bit of machinery, we have now found a method to solve for the optimal value functions. Moreover, this approach can also be applied to the Q value functions. This simple bit of machinery allows us to do *Value Iteration*, a cornerstone solution method in RL. We can initialize a value function to any arbitrary value, and iteratively applying the Bellman optimality operator will converge on the optimal value function. Once the optimal value function is obtained, the optimal policy can be extracted by simply determining which action greedily maximizes the value at a given state. That is $\pi^*(s) = \underset{a \in \mathcal{A}}{arg\,max}\ Q^*(s,a)$. This provides us with our first set of solution methods, but as we will see, they require a strong assumption: we have oracle access to the dynamics of the environment P, R. In RL, this is never the case, though these solution methods will motivate other solution methods. Finally, the Bellman evaluation operator B defined without the max operation can also be shown to be a contraction mapping, whose fixed point solution is the value of a state with respect to the fixed policy. Hence, it is a way to evaluate the value of a policy, given any initialization of the value vector.

4 Solution Approaches I: Dynamic Programming

4.1 Value Iteration

- Initialize $V(s)$ arbitrarily $\forall s$.

- $\Delta := \epsilon$
- While $\Delta \geq \epsilon$
  - $\Delta := 0$
  - $\forall s \in \mathcal{S}$:
    - $v := V(s)$
    - $V(s) := \max_a \sum_{s'} P(s'|a,s)[R(s,a,s') + \gamma V(s')]$
    - $\Delta := max(\Delta, |v - V(s)|)$
- Output a deterministic policy, π, such that:
- $\pi(s) = \arg\max_{a \in \mathcal{A}} \sum_{s'} P(s'|a,s)[R(s,a,s') + \gamma V(s')]$

4.2 Policy Evaluation

- Initialize $V(s)$ arbitrarily $\forall s$.
- $\Delta := \epsilon$
- While $\Delta \geq \epsilon$
  - $\Delta := 0$
  - $\forall s \in \mathcal{S}$:
    - $v := V(s)$
    - $V(s) := \sum_a \pi(a|s) \sum_{s'} P(s'|a,s)[R(s,a,s') + \gamma V(s')]$
    - $\Delta := max(\Delta, |v - V(s)|)$
- Output $V \approx V^\pi$

4.3 Policy Improvement

Determining the value of a policy can allow us to improve upon it. Suppose we have evaluated a given policy and have V^π. Suppose furthermore that we evaluate $Q^\pi(s,a)$ and find that $\exists\ a' \neq \pi(s)$ s.t. $Q^\pi(s,a') \geq V^\pi(s)$. This means that once at state s, it is better to take the action a' once (thus straying from our policy π), then returning back to π for all further action selections. Moreover, this will be true *every* time we encounter this state s. Hence, we can update our policy so that we now take a' when in state s. For π' which is identical as π except at the single state s where it is an improvement, we have, roughly, the

policy improvement theorem which follows:

$$
\begin{aligned}
V^{\pi}(s_t) &\leq Q^{\pi}(s_t, \pi^{'}(s_t)) \\
&= \mathbb{E}_{\pi'}[r_t + \gamma V^{\pi}(s_{t+1})|s_t = s] \\
&\leq \mathbb{E}_{\pi'}[r_t + \gamma Q^{\pi}(s_{t+1}, \pi^{'}(s_{t+1}))|s_t = s] \\
&= \mathbb{E}_{\pi'}[r_t + \gamma \mathbb{E}_{\pi'}[r_{t+1} + \gamma V^{\pi}(s_{t+2})]|s_t = s] \\
&= \mathbb{E}_{\pi'}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} V^{\pi}(s_{t+2})|s_t = s] \\
&. \\
&. \\
&. \\
&\leq \mathbb{E}_{\pi'}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + ...|s_t = s] \\
&= V^{\pi^{'}}(s)
\end{aligned}
$$

4.4 Policy Iteration

5 Solution Approaches II: RL

5.1 Monte Carlo Methods

5.2 Temporal Difference Learning: Q-learning

5.2.1 Incremental updates and tracking non-stationarity

Maintaining a running average of some quantity after receiving $t+1$ samples

$$
\begin{aligned}
Q_{t+1} &:= \frac{1}{t+1}\sum_{k=1}^{t+1} r_k \\
&= \frac{1}{t+1}\Big(r_{t+1} + \sum_{k=1}^{t} r_k\Big) \qquad\qquad = \frac{1}{t+1}(r_{t+1} + tQ_t + Q_t - Q_t) \\
&= \frac{1}{t+1}(r_{t+1} + (t+1)Q_t - Q_t \\
&= Q_t + \frac{1}{t+1}\Big(r_{t+1} - Q_t\Big) \qquad\qquad = (1 - \frac{1}{t+1})Q_t + \frac{1}{t+1}r_{t+1}
\end{aligned}
$$

The last two lines above can be restated more abstractly as both:

- New Estimate ← Old Estimate + Step Size[Target - Old Estimate]
- New Estimate ← (1- Step Size)Old Estimate + (Step Size)Target

This general formulation is helpful for tracking nonstationary distributions, but also for when we must initialize Q_0 to random values. This is the exponentially weighted moving averages, and is the cornerstone for the updates

performed in almost all RL algorithms. We note

$$\begin{aligned} Q_t &= \alpha r_t + (1-\alpha)Q_{t-1} \\ &= \alpha r_t + (1-\alpha)\alpha r_{t-1} + (1-\alpha)^2 Q_{t-2} \\ &= \alpha r_t + (1-\alpha)\alpha r_{t-1} + (1-\alpha)^2 \alpha r_{t-2} + ... + (1-\alpha)^{t-1}\alpha r_1 + (1-\alpha)^t Q_0 \\ &= (1-\alpha)^t Q_0 + \sum_{k=1}^{t} \alpha(1-\alpha)^{t-k} r_k \end{aligned}$$

The step size parameter may (and often is) kept constant, however if $\alpha_t = \frac{1}{t}$ we simply recover the sample average method, which allows for the estimator to converge to the true value by the law of large numbers. More generally, a result from stochastic approximation theory gives us conditions required to assure convergence with probability 1:

- $\sum_{t=1}^{\infty} \alpha_t = \infty$, and that $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$

These EWMA methods will be useful for us downstream when we explore different RL algorithmic solution methods.

5.3 PAC-MDP

6 Deep RL

6.1 Deep Q-networks (DQN)

6.2 Policy Gradients

6.3 Actor Critic