CONTENTS

1 Reinforcement Learning 1

1.1 Definitions 1

1.2 Bellman Equations 2

1.3 Dynamic Programming 5

1.4 TD Learning 5

1 REINFORCEMENT LEARNING

1.1 DEFINITIONS

A set of agents $\{A_i\}_{i=1}^n$ receive observations and rewards from a shared environment, they all decide on actions, these actions are combined into one joint action and fed back into the environment, and the process repeats.

An **episode** is a complete environment run from intitial to terminal state (can end via reaching a terminal state or hitting a max number of timesteps).

> **Definition 1** (MDP)**.** A **finite MDP** consists of:
>
> - a finite set of states $\mathcal{S}$ with terminal states $\hat{\mathcal{S}} \subset \mathcal{S}$.
> - a finite set of actions $\mathcal{A}$
> - a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$
> - a state transition probability function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$
> - an initial state distribution $\mu : \mathcal{S} \to [0, 1]$
>
> In a **partially observable MDP (POMDP)**, agents receive observations o^t of the state s^t, where $s^t \mapsto o_t$ is some (potentially stochastic) map.

Common adaptations:

1. continuous states and/or actions (or mixtures of continuous and discrete)
2. $\mathcal{R}$ can be made probabilistic

> **Definition 2** (Markov property)**.** The **Markov property** asserts that future states and rewards are only dependent on the latest state-action pair:
>
> $$\mathbb{P}\left(s^{t+1}, r^t \mid s^0, a^0, \ldots, s^t, a^t\right) = \mathbb{P}\left(s^{t+1}, r^t \mid s^t, a^t\right).$$

Agents typically try to maximize the **discounted return**

$$\mathbb{E}_\pi\left(\sum_{t=0}^{\infty} \gamma^t r^t\right),$$

where $\gamma < 1$ is a fixed **discount factor**. If $|r| < r_{max}$, then

$$\mathbb{E}_\pi \left(\sum_{t=0}^{\infty} \gamma^t r^t \right) \leq r_{max} \sum_{t=0}^{\infty} \gamma^t = \frac{r_{max}}{1-\gamma} < \infty.$$

In words, bounded returns and $\gamma < 1$ means the discounted return is finite.

The notation u^t refers to the realized (not expected) discounted return starting at time t:

$$\begin{aligned} u^t &:= r^t + \gamma r^{t+1} + \gamma^2 r^{t+2} + \dots \\ &= r^t + \gamma u^{t+1}. \end{aligned}$$

1.2 BELLMAN EQUATIONS

Definition 3 (Value function)**.** A **value function** maps states to a policy's expected return from that point. A **state-value function** (Q function) does the same, but with state-action pairs.

$$\begin{aligned} V^\pi(s) &:= \mathbb{E}_\pi \left(u^t \mid s^t = s \right) \\ Q^\pi(s,a) &:= \mathbb{E}_\pi \left(u^t \mid s^t = a, a^t = a \right) \end{aligned}$$

Note $V^\pi(s) = \mathbb{E}_{a \sim \pi} \left(Q^\pi(s, \cdot) \right) = \sum_a \pi(a|s) Q^\pi(s,a)$.

A policy π^* is optimal if its value functions are optimal:

$$\begin{aligned} V^*(s) &= \max_\pi V^\pi(s) \text{ for all } s \\ Q^*(s,a) &= \max_\pi Q^\pi(s,a) \text{ for all , } s, a. \end{aligned}$$

Both V^π and Q^π can be rewritten recursively, and in that form are called the **Bellman equations**. Expanding $u^t = r^t + \gamma u^{t+1}$ and applying the law of total expectation over states s^{t+1} yields the following.

Proposition 1. Value and state-value functions satisfy the following Bellman equations:

$$\begin{aligned} V^\pi(s) &= \sum_{s',a} \pi(a|s) \mathcal{T}(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma V^\pi(s') \right), \\ Q^\pi(s,a) &= \sum_{s'} \mathcal{T}(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma V^\pi(s') \right) \\ &= \sum_{s'} \mathcal{T}(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma \sum_{a'} \pi(a'|s') Q^\pi(s',a') \right). \end{aligned}$$

Consider the system of equations given by the Bellman equation for $V^\pi(s_i)$ for each $s_i \in \mathcal{S}$. This system of $|\mathcal{S}|$ equations can be solved by any usual linear system solver, e.g. Gaussian elimination.

We can make the Bellman equations express how optimal value functions behave by introducting a max term (which also makes them nonlinear).

Proposition 2. The optimal value functions satisfy the Bellman optimality equations:

$$V^*(s) = \max_a \sum_{s'} \mathcal{T}(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma V^*(s')\right),$$

$$Q^*(s,a) = \sum_{s'} \mathcal{T}(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma \max_{a'} Q^*(s',a')\right).$$

The Bellman equations are useful for bootstrapping value function estimators, and as such can be thought of as operators acting on value functions. This viewpoint also allows us to prove that the Bellman equations 1) converge under boostrapping, and 2) have unique solutions.

Definition 4 (Bellman operators)**.** For a policy π, the **Bellman operators** are

$$\mathcal{B}^\pi : V(s) \mapsto \sum_{s',a} \pi(a|s)\mathcal{T}(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma V(s')\right)$$

$$\mathcal{B}^\pi : Q(s,a) \mapsto \sum_{s'} \mathcal{T}(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma \sum_{a'} \pi(a'|s') Q(s',a')\right).$$

The **Bellman optimality operators** are

$$\mathcal{B}^* : V(s) \mapsto \max_a \sum_{s'} \mathcal{T}(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma V(s')\right)$$

$$\mathcal{B}^* : Q(s,a) \mapsto \sum_{s'} \mathcal{T}(s'|s,a) \left(\mathcal{R}(s,a,s') + \gamma \max_{a'} Q^*(s',a')\right).$$

I've overloaded the notation to have $\mathcal{B}$ apply to both value functions and state-value functions, since the proper operator should be clear from context.

Both convergence and uniquness of solution follow from the fact that these operators are contraction maps in the max norm.

Definition 5. A (γ)-**contraction map** on a complete metric space (X,d) is a map $\phi : X \to X$ such that

$$d(\phi(x), \phi(y)) \leq \gamma \, d(x,y)$$

for all $x, y \in X$, where $\gamma \in [0,1)$ is constant.

Theorem 1 (Contraction Mapping Principle)**.** A contraction map ϕ has a unique fixed point x^* that can be computed as the limit of successive applications of ϕ to an arbitrary starting point x^0:

$$x^0 \to \phi(x^0) \to \phi(\phi(x^0)) \to \cdots \to x^*.$$

Proof. See Marsden & Hoffman pg. 301. □

Proposition 3. The Bellman operators are γ-contraction maps in the max norm.

Proof. Fix a policy π. For any value functions V and W,

$$\begin{aligned}
\|\mathcal{B}^\pi V - \mathcal{B}^\pi W\|_\infty &= \max_s |(\mathcal{B}^\pi V)(s) - (\mathcal{B}^\pi W)(s)| \\
&= \gamma \max_s \left| \sum_{s',a} \mathcal{T}(s'|s,a)\pi(a|s)\left(V(s') - W(s')\right) \right| \\
&\leq \gamma \max_s \sum_{s',a} \mathcal{T}(s'|s,a)\pi(a|s)\left|V(s') - W(s')\right| \\
&\leq \gamma \|V - W\|_\infty \max_s \sum_{s',a} \mathcal{T}(s'|s,a)\pi(a|s) \\
&= \gamma \|V - W\|_\infty .
\end{aligned}$$

The last line is because, by the law of total expectation,

$$\sum_{s',a} \mathcal{T}(s'|s,a)\pi(a|s) = \sum_{s'} \mathbb{P}\left(s'|s\right) = 1$$

for all s. For any state-value functions P and Q, **Computation** The last line is because

$$\sum_{s',a'} \mathcal{T}(s'|s,a)\pi(a'|s') = \sum_{s',a'} \mathcal{T}(s'|s,a)\pi(a'|s',s,a) = \sum_{a'} \mathbb{P}\left(a'|s,a\right) = 1$$

for all s, a. The computations for $\mathcal{B}^*$ are almost identical, except they require the additional property

$$\left| \max_x f(x) - \max_x g(x) \right| \leq \max_x |f(x) - g(x)| .$$

This is true because of the reverse triangle inequality:

$$\left| \max_x f(x) - \max_x g(x) \right| = \left| \|f\|_\infty - \|g\|_\infty \right| \leq \|f - g\|_\infty = \max_x |f(x) - g(x)| .$$

□

Corollary 1. The Bellman equations have unique solutions. Furthermore, successive application of the Bellman operators converges to a unique value function:

$$\begin{gathered}
\mathcal{B}^\pi \to V^\pi \text{ or } Q^\pi \\
\mathcal{B}^* \to V^* \text{ or } Q^*
\end{gathered}$$

Proof. Each of the Bellman operators is a contraction map, so by the contraction mapping principle, successive application of each operator to an arbitrary starting function converges to a unique fixed point.

Since the provided functions V^π, V^*, Q^π, and Q^* are clearly fixed points of their respective operators, they must be the aforementioned unique fixed points.

Note that since we're working in the finite MDP setting, the vector spaces of value funtions and state-value functions are finite dimensional. All norms on finite dimensional vector spaces are equivalent, so convergence is independent of any norm. □

1.3 DYNAMIC PROGRAMMING

If we have full knowledge of the MDP (i.e. $\mathcal{T}$ and $\mathcal{R}$), then we can use a DP approach to compute an optimal policy.

There are two main types of DP approaches to this:

- **Policy iteration:** switch between policy evaluation & policy improvement, producing a sequence

$$\pi^0 \to V^0 \to \pi^1 \to V^1 \to \cdots \to \pi^* \to V^*$$

- **Value iteration:** directly measure a value function and back out an optimal policy at the end, producing a sequence

$$V^0 \to V^1 \to \cdots \to V^*$$

In policy iteration, given a policy π, we could use Gaussian elimination on the system of Bellman equations to calculate $V^\pi(s)$ for all $s \in \mathcal{S}$, but this is $\mathcal{O}(|\mathcal{S}|^3)$. And in the case of value iteration, the system of Bellman optimality equations is nonlinear anyway.

Instead, we use bootstrapping to recursively back out a value function. For policy iteration, fix π, then we can compute V^π as the limit of the sequence generated by

$$V^{i+1} \leftarrow \mathcal{B}^\pi V^i.$$

This gives us a method for policy evaluation (the step $\pi^i \to V^i$), but we still need to produce a better policy π^{i+1}.

policy improvement

Alternatively, we can do value iteration instead of policy iteration, where we directly find V^* and then back out a policy π^* at the end. We can do this by generating the sequence

$$V^{i+1} \leftarrow \mathcal{B}^* V^i,$$

which converges to V^*. We then back out an optimal policy via

$$\pi^*(s) = \arg\max_a \sum_{s'} \mathcal{T}(s'|s,a)\left(\mathcal{R}(s,a,s') + \gamma V^*(s')\right).$$

Note that this is different than "go to the state with the highest value", as it requires a one-step lookahead. Consider an environment with a target (absorbing) state, where an agent receives a reward from moving into the target state and never in any other scenario. In this case, the learned value function is maximized in the closest state to the target state, although clearly the desired agent behavior isn't to get one state away from the target and then stop.

value iteration algo

1.4 TD LEARNING

Intro

Theorem 2 (Jaakkola et al.)**.** Consider the stochastic process

$$\Delta_{t+1}(x) := (1 - \alpha_t(x))\Delta_t(x) + \alpha_t(x)F_t(x)$$

defined on $(\Omega, \mathcal{F}, \mathbb{P})$, where Δ_t and α_t are $\mathcal{F}_t$-measurable and F_t is $\mathcal{F}_{t+1}$-measurable. This process converges to 0 with probability 1 if for all x and t:

1. the coefficients α satisfy the **standard stochastic approximation conditions**
   - $\sum_t \alpha_t(x) = 0$
   - $\sum_t \alpha_t^2(x) < \infty$
   - $0 \leq \alpha_t(x) \leq 1$
2. $\|\mathbb{E}\left(F_t(x) \mid \mathcal{F}_t\right)\|_\infty \leq \gamma\|\Delta_t\|_\infty$ for fixed $\gamma \in [0, 1)$
3. $\mathbb{V}\left(F_t(x) \mid \mathcal{F}_t\right) \leq C\left(1 + \|\Delta_t(x)\|_\infty\right)$ for fixed $C \geq 0$

At time t, the entry Δ_t can be thought of as an old model that is being updated via interpolation to become closer to a new model F_t that draws from a new timestep of information (thus why F_t is $\mathcal{F}_{t+1}$-measurable instead of $\mathcal{F}_t$). In the context of RL, this looks like having access to s^{t+1} when bootstrapping.

Lemma 1. For random variables $0 \leq X \leq x_{\text{max}}$ and $0 \leq Y \leq y_{\text{max}}$,

$$\text{Cov}(X, Y) \leq x_{\text{max}}y_{\text{max}}.$$

This lemma implies that for bounded non-negative random variables $0 \leq X \leq x_{\text{max}}$, variance is bounded as $\mathbb{V}(X) \leq x_{\text{max}}$; although Popociviu's variance bound already asserts this without the non-negativity requirement, namely $|X| \leq x_{\text{max}}$.

Proof. $\text{Cov}(X, Y) = \mathbb{E}\left(XY\right) - \mathbb{E}\left(X\right)\mathbb{E}\left(Y\right) \leq \mathbb{E}\left(XY\right) \leq x_{\text{max}}y_{\text{max}}$. □

Proposition 4. Suppose the learning rate $\alpha_t(s, a)$ satisfies the standard stochastic approximation conditions, $0 \leq \mathcal{R} \leq r_{\text{max}}$, and $Q^0 \geq 0$. Then Q-learning converges to Q^*.

Proof. This is just checking the conditions from the previous theorem. Define the process $\{\Delta_t\}$ by

$$\Delta_t(s, a) := Q^t(s, a) - Q^*(s, a)$$

and

$$F_t(s^t, a^t) := r^t + \gamma \max_{a'} Q^t(s^{t+1}, a') - Q^*(s^t, a^t).$$

To show (2), we use Corollary 1's result that the Bellman operator $\mathcal{B}^*$ is a γ-contraction in the max norm with unique fixed point Q^*:

$$\begin{aligned}
\|\mathbb{E}\left(F_t(s^t, a^t) \mid \mathcal{F}_t\right)\|_\infty &= \|\mathcal{B}^*Q^t - Q^*\|_\infty \\
&= \|\mathcal{B}^*Q^t - \mathcal{B}^*Q^*\| \\
&\leq \gamma\|Q^t - Q^*\|_\infty \\
&= \gamma\|\Delta_t\|_\infty.
\end{aligned}$$

Showing (3) requires the preceding lemma; note that if $\mathcal{R} \geq 0$ and $Q^0 \geq 0$, then the Q-learning update rule ensures that $Q^t \geq 0$ for all t. Noting that $Q^*(s^t, a^t)$ has no randomness when conditioning on $\mathcal{F}_t$,

$$\begin{aligned}
\mathbb{V}(F_t \mid \mathcal{F}_t) &= \mathbb{V}(r^t + \gamma \max_{a'} Q^t(s^{t+1}, a') - Q^*(s^t, a^t) \mid \mathcal{F}_t) \\
&= \mathbb{V}(r^t + \gamma \max_{a'} Q^t(s^{t+1}, a') \mid \mathcal{F}_t) \\
&= \mathbb{V}(r^t \mid \mathcal{F}_t) + \gamma^2 \mathbb{V}(\max_{a'} Q^t(s^{t_1}, a')) + 2\gamma \mathrm{Cov}(r^t, \max_{a'} Q^t(s^{t+1}, a')) \\
&\leq r_{\text{max}}^2 + \gamma^2 \|Q^t\|_\infty^2 + 2\gamma r_{\text{max}} \|Q^t\|_\infty \\
&\leq r_{\text{max}}^2 + 4\gamma^2 \|Q^t\|_\infty^2 + 2\gamma r_{\text{max}} \|Q^t\|_\infty \\
&= (r_{\text{max}} + 2\gamma \|Q^t\|_\infty)^2 \\
&= (r_{\text{max}} + 2\gamma \|Q^* + \Delta_t\|_\infty)^2 \\
&\leq (r_{\text{max}} + 2\gamma \|Q^*\|_\infty + 2\gamma \|\Delta_t\|_\infty)^2 \\
&\leq C(1 + \|\Delta_t\|_\infty)^2
\end{aligned}$$

where $C = \max\{r_{\text{max}} + 2\gamma\|Q^*\|_\infty, 2\gamma\}$. By the preceding theorem, the process $\Delta_t = Q^t - Q^*$ converges to 0 with probability 1, i.e. $Q^t \to Q^*$. □

Note 1. If we assume s^{t+1} and r^t are independent (the covariance term vanishes), then we can use Popoviciu's variance bound to satisfy (3) under the relaxed assumption of $|\mathcal{R}| \leq r_{\text{max}}$, i.e. no need for $\mathcal{R}, Q^0 \geq 0$.

I don't mind the non-negativity assumption, though, since given a bounded reward function, you can always make it non-negative by shifting all rewards by a fixed amount. And initializing a Q function is arbitrary anyway, so might as well make Q^0 non-negative.