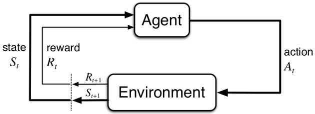
Introduction to Reinforcement learning

Tri Nguyen

Summer Reading 2021 - Oregon State University

August 19, 2021

RL Framework



Input of the framework

Environment - Agent interation:

- At time step t, *agent* at state S_t performs an action $A_t \in \mathcal{A}_t$
- Environment's dynamics. The *environment* acts accordingly change to state S_{t+1}, emits a reward R_{t+1}
- Episodic/Continuing task. *Return*: $G_t = \sum_{k=1}^{\infty} \gamma^k R_{t+k+1}$

Output of the framework

How. An agent that acts on the environment so that it maximizes the expectation of *return*, i.e., $\mathbb{E}[G_t]$.

Roadmap

Given an MDP, find an optimal policy.

- Policy Iteration
- Value Iteration
- Connection between 2 and variants

Environment's dynamics

Markov decision process (MDP) descibes environment's dynamics

- Finite sets of states, action, rewards $\mathcal{S}, \mathcal{A}, \mathcal{R}$
- Random variables $S_t \in \mathcal{S}, R_t \in \mathcal{R}$ are only dependent on preceding state and action, i.e.,
  $p(s', r | s, a) := \Pr(S_t = s', R_t = r | S_{t-1} = s, A_{t-1} = a)$
- **Markov property.** State must include all information of the past that makes a difference for the future

Policy and Value Function

Definition

- Policy: $\pi(a|s) = \Pr(A_t = a|S_t = s)$
- Value function: $v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi\left[R_{t+1} + \gamma G_{t+1}|S_t = s\right]$
- Action-value function: $q_\pi(s, a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a\right]$
- Optimal policy: π_* such that $v_{\pi_*}(s) \geq v_\pi(s)$ and for all $s \in \mathcal{S}$
- Optimal value function $v_{\pi_*} s =$

Remark

- (Bellman equation) v_π is the unique solution of

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|S_t = s, A_t = a)\left(r + \gamma v_\pi(s')\right)$$

- (Bellman optimality equation) v_* is the unique solution of

$$\Rightarrow v_*(s) = \max_{a \in \mathcal{A}(s)} \sum_{s',r} p(s', r|S_t = s, A_t = a)\left(r + \gamma v_*(s')\right) \qquad (1)$$

Policy Evaluation

Let $\boldsymbol{v} = \begin{bmatrix} v_\pi(s_1) & \dots & v_\pi(s_n) \end{bmatrix}^T$,

$$\boldsymbol{R} = \begin{bmatrix} \mathbb{E}[R_{t+1}|S_t = s_1] \\ \dots \\ \mathbb{E}[R_{t+1}|S_t = s_n] \end{bmatrix}, \quad \boldsymbol{P} = \begin{bmatrix} p(s_1|s_1) & p(s_2|s_1) & \dots & p(s_n|s_1) \\ \dots & & & \\ p(s_1|s_n) & p(s_2|s_n) & \dots & p(s_n|s_n) \end{bmatrix}$$

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|S_t = s, A_t = a)\,(r + \gamma v_\pi(s'))$$

$$\Rightarrow \boldsymbol{v}_s = \mathbb{E}_\pi[R_{t+1}|S_t = s] + \sum_{i=1}^{n} \gamma p(s_i|S_t = s)\boldsymbol{v}_{s_i}$$

Then Bellman equation in matrix form is

$$\boldsymbol{v} = \boldsymbol{R} + \gamma \boldsymbol{P}\boldsymbol{v}$$

Any method to solve a linear system can be used to find $\boldsymbol{v}$*?*

Iterative Policy Evaluation

- Let $T(\boldsymbol{v}) = \boldsymbol{R} + \gamma \boldsymbol{P}\boldsymbol{v}$. By fixed-point property, a sequence $\boldsymbol{v}_k$ where $\boldsymbol{v}_k = T(\boldsymbol{v}_{k-1})$ will converge to $\boldsymbol{v}_*$.

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r\,|\,s, a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

- *Expected updates*: In order to produce new $\boldsymbol{v}$, the algorithm updates value of every state
- *In-place update*: Directly used new value of $v_\pi(s)$ during updating $v_\pi(s')$. . It is valid update since

$$|T_\pi(\boldsymbol{v})[s] - T_\pi(\boldsymbol{v}')[s]| \leq \gamma \left\| \boldsymbol{v} - \boldsymbol{v}' \right\|_\infty$$

Policy Improvement

Theorem

For 2 deterministic policy π, π', if for all $s \in \mathcal{S}$,

$$q_\pi(s, \pi'(s)) \geq v_\pi(s) \Rightarrow \pi' \geq \pi$$

Proof.

$$
\begin{aligned}
v_\pi(s) &\leq q_\pi(s, \pi'(s)) \\
&= \mathbb{E}_\pi \left[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s, A_t = \pi'(s)\right] \\
&= \mathbb{E}_{\pi'} \left[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s\right] \\
&\leq \mathbb{E}_{\pi'} \left[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) | S_t = s\right] \\
&= \mathbb{E}_{\pi'} \left[R_{t+1} + \gamma \mathbb{E}\left[R_{t+2} + \gamma v_\pi(S_{t+2}) | S_{t+1}, A_{t+1} = \pi'(S_{t+1})\right] | S_t = s\right] \\
&= \mathbb{E}_{\pi'} \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2}) | S_t = s\right] \\
&\leq \dots \\
&\leq \mathbb{E}_{\pi'} \left[R_{t+1} + \gamma R_{t+2} + \gamma^3 R_{t+3} + \dots | S_t = s\right] \\
&= v_{\pi'}(s)
\end{aligned}
$$

- Key to improvement is to find π'

$$q_\pi(s, \pi'(s)) \geq v_\pi(s) \tag{2}$$

- Seek for an action to improve in short term (1 step)

$$\pi'(s) := \arg\max_a q_\pi(s, a)$$

  then by Policy improvement theorem, $\pi' \geq \pi$

- If no improvement is available, i.e., $v_\pi = v_{\pi'}$ then $v_\pi = v_*$ because

$$\begin{aligned} v_{\pi'}(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma v_\pi(s) | S_t = s, A_t = a] \\ &= \max_a \mathbb{E}[R_{t+1} + \gamma v_{\pi'}(s) | S_t = s, A_t = a] \end{aligned}$$

- Note that $v_\pi = v_{\pi'}$ but not $\pi = \pi'$ gives a hint about reaching optimal value.

Policy Iteration

$$\pi_0 \xrightarrow{\text{Eval}} v_{\pi_0} \xrightarrow{\text{Improve}} \pi_1 \xrightarrow{\text{Eval}} v_{\pi_1} \xrightarrow{\text{Improve}} \dots \xrightarrow{\text{Improve}} \pi_* \xrightarrow{\text{Eval}} v_{\pi_*}$$

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   - $\Delta \leftarrow 0$
   - Loop for each $s \in \mathcal{S}$:
     - $v \leftarrow V(s)$
     - $V(s) \leftarrow \sum_{s',r} p(s', r \,|\, s, \pi(s)) \big[r + \gamma V(s')\big]$
     - $\Delta \leftarrow \max(\Delta, |v - V(s)|)$

   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   - *old-action* $\leftarrow \pi(s)$
   - $\pi(s) \leftarrow \operatorname{argmax}_a \sum_{s',r} p(s', r \,|\, s, a) \big[r + \gamma V(s')\big]$
   - If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*

   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Figure: There's a subtle bug

- Truncated variant: run a small K number of iterations for step 2.
- Converge very fast in few iterations, but computationally expensive because of policy evaluation

Value Iteration

Combine policy improvement and truncated policy evalution in one step:

$$v_{k+1}(s) := \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) | S_t = s, A_t = a] \tag{3}$$

- Truncated policy evaluation: Vanilla policy evalution with only 1 iteration
- (3) is Bellman optimality equation if substituting v_k, v_{k+1} by v_*

Value Iteration, for estimating $\pi \approx \pi_*$

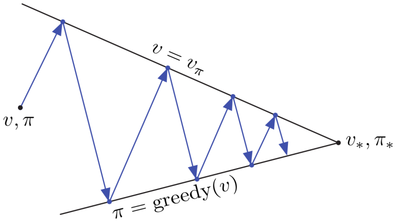
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r | s, a)\big[r + \gamma V(s')\big]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
    $\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r | s, a)\big[r + \gamma V(s')\big]$

- The inner loop does not neccessarily need to run over all $s \in \mathcal{S}$ (*Asynchronous DP*)

Generalized Policy Iteration



There's is a notion of alternating between policy evaluation and policy improvement

- In PI, a policy improvement is performed after a completing policy evaluation and vice versa.
- In VI, only single iteration of policy evaluation is performed in between each policy improvement.
- We can even interleaved at finer grain: mixed asynchronous DP in VI with PI
- Policy evaluation and policy improvement are both competing and coorperating.

Algorithm 1 `Value iteration`

Input: An MDP, an initial value v_0

Output: An (approximately) optimal policy

$k \leftarrow 0$

repeat

$v_{k+1} \leftarrow T v_k$ // Update the value

$k \leftarrow k+1$

until some stopping criterion

Return greedy(v_k)

Algorithm 2 `Policy iteration`

Input: An MDP, an initial policy π_0

Output: An (approximately) optimal policy

$k \leftarrow 0$

repeat

$v_k \leftarrow (I - \gamma P^{\pi_k})^{-1} r^{\pi_k}$ // Estimate the value of π_k

$\pi_{k+1} \leftarrow$ greedy(v_k) // Update the policy

$k \leftarrow k+1$

until some stopping criterion

Return π_k

Figure: $v = R + \gamma P v \Rightarrow v = (I - \gamma P)^{-1} R$

Define

$$T_\pi(v)[s] := \sum_a \pi(a|s) \sum_{s',r} p(s', r | S_t = s, A_t = a)\left(r + \gamma v_\pi(s')\right)$$

$$T(v)[s] := \max_{a \in \mathcal{A}(s)} \sum_{s',r} p(s', r | S_t = s, A_t = a)\left(r + \gamma v_*(s')\right)$$

- Value iteration

$$\begin{cases} \pi_{k+1} \leftarrow \text{greedy}(v_{\pi_k}) \\ v_{k+1} \leftarrow T_{\pi_{k+1}}(v_k) \end{cases} \Leftrightarrow \begin{cases} \pi_{k+1} \leftarrow \text{greedy}(v_{\pi_k}) \\ v_{k+1} \leftarrow R + \gamma P v_k \end{cases}$$

- Policy iteration

$$\begin{cases} \pi_{k+1} \leftarrow \text{greedy}(v_{\pi_k}) \\ v_{k+1} \leftarrow T^\infty_{\pi_{k+1}}(v_k) \end{cases} \Leftrightarrow \begin{cases} \pi_{k+1} \leftarrow \text{greedy}(v_{\pi_k}) \\ v_{k+1} \leftarrow (I - \gamma P)^{-1} R \end{cases}$$

- Bertsekas and Ioffe (1996) introduced an operator which is proved to be a $\lambda\gamma$-contraction respect to l_∞ norm

$$\begin{aligned} M_k(v) &:= (1-\lambda)T_{\pi_{k+1}}(v_k) + \lambda T_{\pi_{k+1}}(v) \\ &= (1-\lambda)(R + \gamma P v_k) + \lambda(R + \gamma P v) \\ &= R + (1-\lambda)\gamma P v_k + \lambda\gamma P v \end{aligned}$$

- Since $M_k(v)$ is a contraction, it has a unique fixed point, $v_M = M_k^\infty(v)$ exists

$$\begin{aligned} v_M &= R + (1-\lambda)\gamma P v_k + \lambda\gamma P v_M \\ \Leftrightarrow v_M &= (I - \lambda\gamma P)^{-1}(R + (1-\lambda)\gamma P v_k) \end{aligned}$$

- Use $M_k(v)$, λ policy iteration's update is given by.

$$\begin{cases} \pi_{k+1} \leftarrow \text{greedy}(v_{\pi_k}) \\ v_{k+1} \leftarrow (I - \lambda\gamma P)^{-1}(R + (1-\lambda)\gamma P v_k) \end{cases}$$

- Let $T_\lambda(v) := v_M = (I - \lambda\gamma P)^{-1}(R + (1-\lambda)\gamma P v_k)$ be operator.

$$\begin{cases} \pi_{k+1} \leftarrow \text{greedy}(v_{\pi_k}) \\ v_{k+1} \leftarrow T_\lambda(v_k) \end{cases}$$

- In the updating step, λ policy iteration trying to find fixed point of operator $M_k(v)$.
- Define a sequence of $v_1, v_2, \ldots$ such as $v_{j+1} = M_k(v_j)$ We have:

$$\begin{aligned}
v_{j+1} = M_k(v_j) &= (1-\lambda) T_{\pi_{k+1}}(v_k) + \lambda T_{\pi_{k+1}}(v_j) \\
&= (1-\lambda) T_{\pi_{k+1}}(v_k) + \lambda T_{\pi_{k+1}}(M_k(v_{j-1})) \\
&= (1-\lambda) T_{\pi_{k+1}}(v_k) + \lambda T_{\pi_{k+1}} \left((1-\lambda) T_{\pi_{k+1}}(v_k) + \lambda T_{\pi_{k+1}}(v_{j-1})\right) \\
&= (1-\lambda)(T_{\pi_{k+1}}(v_k) + \lambda T^2_{\pi_{k+1}}(v_k)) + \lambda^2 T^2_{\pi_{k+1}} T_{\pi_{k+1}}(v_{j-1})) \\
&= \ldots \\
&= (1-\lambda) \sum_{j=0}^{\infty} \lambda^j T^{j+1}_{\pi_{k+1}}(v_k)
\end{aligned}$$

so that the update rule becomes $\begin{cases} \pi_{k+1} \leftarrow \text{greedy}(v_{\pi_k}) \\ v_{k+1} \leftarrow (1-\lambda) \sum_{j=0}^{\infty} \lambda^j T^{j+1}_{\pi_{k+1}} v_k \end{cases}$

Summary

- Policy evaluation is based on Bellman equation
- Value iteration is based on Bellman optimality equation
- GPI views a different levels of interleaving between policy evaluation and policy improvement