Dynamic Programming

Dynamic Programming

The term dynamic programming (DP) refers to a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment as a Markov decision process (MDP).

That is, we assume that its state, action, and reward sets, $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{R}$, are finite, and that its dynamics are given by a set of probabilities $p(s', r \mid s, a)$, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$, $r \in \mathcal{R}$, and $s' \in \mathcal{S}^+$ ($\mathcal{S}^+$ is $\mathcal{S}$ plus a terminal state if the problem is episodic).

Dynamic Programming

강화학습과 DP의 아이디어
Value function을 사용해서 좋은 정책(*Optimal policy*)을 찾고 체계화하는 것

Chapter3에서 정의한 value function을 DP를 이용해서 어떻게 계산할 수 있을까?
*Bellman optimality Equation*을 만족하는 *Optimal value function*을 찾으면
*Optimal Policy*를 찾을 수 있다.

$$\begin{aligned} v_*(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_*(s')\Big], \end{aligned} \tag{4.1}$$

or

$$\begin{aligned} q_*(s,a) &= \mathbb{E}\Big[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1},a') \Bigm| S_t = s, A_t = a\Big] \\ &= \sum_{s',r} p(s',r|s,a)\Big[r + \gamma \max_{a'} q_*(s',a')\Big], \end{aligned} \tag{4.2}$$

Dynamic Programming

DP

1. Bellman Equation
2. Update rules

Policy Evaluation(Prediction)

- 임의의 policy, pi에 대한 State-value function v_{pi}를 어떻게 계산할 것인가?
  → *Policy evaluation(prediction problem)*
- Recall from *Chapter 3*

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] && \text{(from (3.9))} \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] && (4.3) \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma v_\pi(s')\Big], && (4.4)
\end{aligned}
$$

- gamma(discount rate) < 1, eventual termination이 있기 때문에 존재성, 유일성은 보장된다.

Policy Evaluation(Prediction)

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] && \text{(from (3.9))} \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s] && (4.3) \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a) \Big[r + \gamma v_\pi(s') \Big], && (4.4)
\end{aligned}
$$

- 환경의 dynamics을 완전히 알 수 있다면, (4.4)는 |S|개의 미지수를 가진 |S|개의 linear equations이다.
  (v_{pi}(s)가 unknowns이고, s는 S에 속하기 때문에)

  → *Tedious solution: Computation*

  → *Iterative solution methods*

  → ***Iterative policy evaluation***

Iterative Policy Evaluation

- Sequence of approximate value functions

$$v_0, v_1, v_2, \ldots$$

Chosen arbitrarily

Iterative Policy Evaluation

- Sequence of approximate value functions

$$v_0, v_1, v_2, \ldots$$

Chosen arbitrarily

Update rule (Bellman Eqtn)

Iterative Policy Evaluation

$$v_0, v_1, v_2, \ldots$$

Chosen arbitrarily

Update rule (Bellman Eqtn)

$$\begin{aligned} v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r | s, a)\Big[r + \gamma v_k(s')\Big], \end{aligned}$$

Iterative Policy Evaluation

$$\begin{aligned} v_{k+1}(s) &\doteq \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_k(s')\Big], \end{aligned}$$

v_k = v_pi : Fixed point (Convergence)
v_pi의 존재성을 만족하는 가정 → v_pi로의 수렴도 보장된다.

Expected update: 가능한 모든 state들에 대해서 평균값을 구하기 때문에

* Implementation details: *One-array(Inplace)*

Iterative Policy Evaluation

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 $V(\mathit{terminal}) = 0$

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 $\Delta < \theta$

Policy Improvement

- Value function을 계산하는 이유?
  → *Find better policies*

- 더 좋은 policy로 바꾸는 게 좋을까, 나쁠까?

$$
\begin{aligned}
q_\pi(s, a) &\doteq \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \sum_{s', r} p(s', r \mid s, a)\Big[r + \gamma v_\pi(s')\Big].
\end{aligned}
$$

- pi라는 policy에 따라, s라는 state에서 a라는 action을 했을 때 state-action value q_pi를 통해 알아보는 방법

Policy Improvement

$$
\begin{aligned}
q_\pi(s,a) &\doteq \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big].
\end{aligned}
$$

- Key Criterion: v_pi(s)보다 클 것인가?
- **True**: *Policy improvement theorem*

Policy Improvement

$$
\begin{aligned}
q_\pi(s,a) &\doteq \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \sum_{s',r} p(s',r \mid s,a)\Big[r + \gamma v_\pi(s')\Big].
\end{aligned}
$$

That this is true is a special case of a general result called the *policy improvement theorem*. Let π and π' be any pair of deterministic policies such that, for all $s \in \mathcal{S}$,

$$
q_\pi(s, \pi'(s)) \geq v_\pi(s). \tag{4.7}
$$

Then the policy π' must be as good as, or better than, π. That is, it must obtain greater or equal expected return from all states $s \in \mathcal{S}$:

$$
v_{\pi'}(s) \geq v_\pi(s). \tag{4.8}
$$

Policy Improvement

$$
\begin{aligned}
v_\pi(s) &\le q_\pi(s, \pi'(s)) \\
&= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = \pi'(s)] && \text{(by (4.6))} \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \boxed{\gamma v_\pi(S_{t+1})} \mid S_t = s] \\
&\le \mathbb{E}_{\pi'}[R_{t+1} + \boxed{\gamma q_\pi(S_{t+1}, \pi'(S_{t+1}))} \mid S_t = s] && \text{(by (4.7))} \\
&= \mathbb{E}_{\pi'}[R_{t+1} + \boxed{\gamma \mathbb{E}[R_{t+2} + \gamma v_\pi(S_{t+2}) | S_{t+1}, A_{t+1} = \pi'(S_{t+1})]} \mid S_t = s] \\
&= \mathbb{E}_{\pi'}\big[R_{t+1} + \boxed{\gamma R_{t+2} + \gamma^2 v_\pi(S_{t+2})} \bigm| S_t = s\big] \\
&\le \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_\pi(S_{t+3}) \bigm| S_t = s\big] \\
&\;\;\vdots \\
&\le \mathbb{E}_{\pi'}\big[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \bigm| S_t = s\big] \\
&= v_{\pi'}(s).
\end{aligned}
$$

Policy Improvement

- Change at all states (Extension)

$$
\begin{aligned}
\pi'(s) &\doteq \underset{a}{\arg\max}\, q_\pi(s,a) \\
&= \underset{a}{\arg\max}\, \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \underset{a}{\arg\max} \sum_{s',r} p(s', r \mid s, a)\Big[r + \gamma v_\pi(s')\Big],
\end{aligned}
$$

Policy Improvement

- Greedy Policy: 그 순간에 가장 이득이 되는 action을 선택하는 policy → Policy improvement thm의 조건을 만족한다.

- value function을 greedy하게 만드는 방법으로 원래 policy보다 나은 policy를 찾는 과정: *Policy improvement*

  → 새로운 policy가 이전의 policy만큼만 좋다면(not better)

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

- Bellman optimality eqtn에 의해 optimal policy가 된다.

✱ Stochastic의 경우로도 쉽게 확장될 수 있다.

Policy Iteration

EvaluationImprovement

$$\pi_0 \xrightarrow{\text{E}} v_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} v_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} v_*,$$

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 $s \in \mathcal{S}$:
     - $v \leftarrow V(s)$
     - $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s)) \big[r + \gamma V(s')\big]$
     - $\Delta \leftarrow \max(\Delta, |v - V(s)|)$

   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* $\leftarrow \pi(s)$
   - $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a) \big[r + \gamma V(s')\big]$
   - 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

Value Iteration

- Policy iteration에서는 iteration의 단계에서 evaluation을 포함하고 있기 때문에, 계산량 문제가 있다.

- 정확히 수렴할 때 까지 기다려야 할까, 그 전에 멈춰도 될까?

- Policy evaluation을 policy iteration의 수렴을 보장하면서도 줄여질 수 있는 방법이 몇가지 있다.

  → 각 state를 한번만 update 하도록 줄이는 방법이 있는데, 이를 *value iteration* 이라고 한다.

Value Iteration

$$
\begin{aligned}
v_{k+1}(s) &\doteq \max_a \mathbb{E}[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s', r} p(s', r \mid s, a) \Big[r + \gamma v_k(s') \Big], \qquad (4.10)
\end{aligned}
$$

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 $V(terminal) = 0$

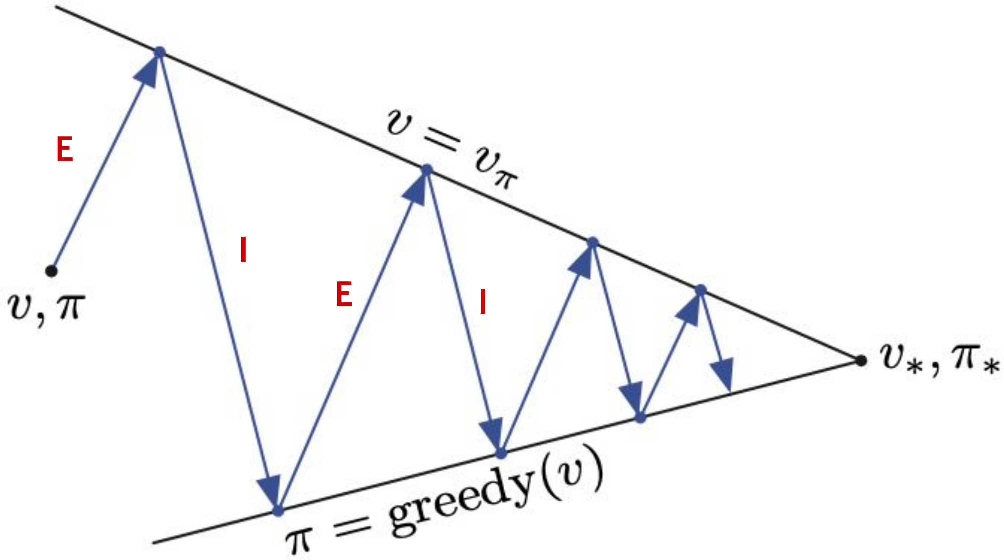
Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $\quad v \leftarrow V(s)$
| $\quad V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a) \big[r + \gamma V(s') \big]$
| $\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

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

Asynchronous DP

- Full-sweep: 모든 state를 참조한다.
  → Computational cost ↑
- In-place iterative DP
  → 중복될 수도 있지만, 수렴을 제대로 하기 위해서는 모든 state를 update 해야한다.
- In-place DP
- Prioritised sweeping
- Real-time DP

Generalized Policy Iteration



Efficiency of DP

- DP는 optimal policy를 polynomial time안에 찾을 수 있다.
- 하지만 모든 state를 참조해야하기 때문에, Curse of dimensionality 문제가 있다.

- Asynchronous DP를 사용한다.(Practical)

Summary

Problem	Bellman Equation	Algorithm
Prediction	Bellman Expectation Equation	Iterative Policy Evaluation
Control	Bellman Expectation Equation + Greedy Policy Improvement	Policy Iteration
Control	Bellman Optimality Equation	Value Iteration

https://www.davidsilver.uk/wp-content/uploads/2020/03/DP.pdf