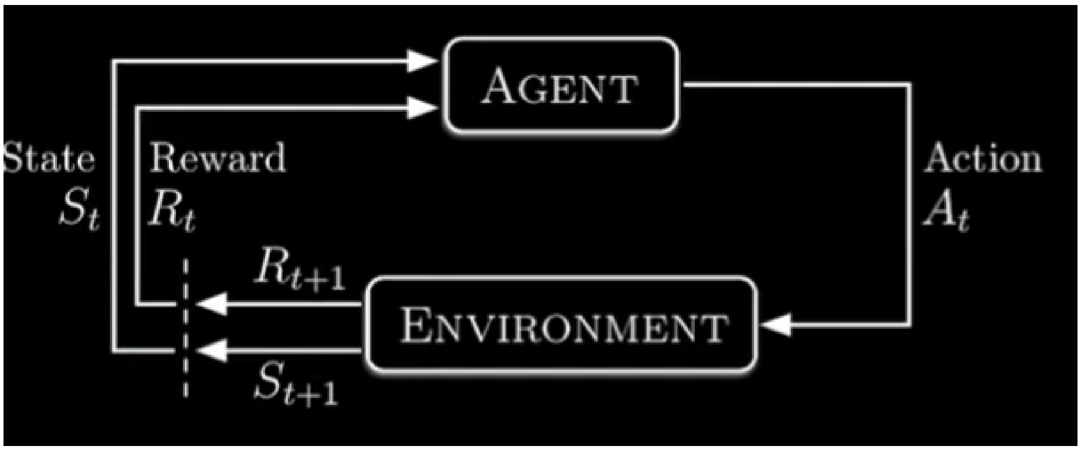
Chapter 3 Finite Markov Decision Tree

Created	@November 2, 2023 1:42 PM
Tags	



Time step = 時間點 $t, t = 0, 1, 2, ...$ 不一定是秒數或是固定長度的時間 但是這樣比較方便解說。

Agent = Decision maker(決策者)

Environment = Agent互動的物件場景,包含(comprising)任何在agent之外的東西.

State = 在每個time step, Agent會收到Environment所產生的狀態 S

Action = Agent 基於 environment 的產生的 State 所做出的行動

Reward = 在Agent與環境互動過後下一個時間點產生的結果。

Trajectory(投射) = 上圖過程若是持續迴圈從這樣開始

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, ...$$

Finite MDP

在有限MDP之下，所有 States, Actions, Rewards($\mathcal{S}, \mathcal{A}, \mathcal{R}$) 的各個子集皆為有限個元素。$R_t, S_t$兩個隨機變數的離散的機率分部只和上一個的Time Step 的State跟Action相關，可以以下式表達。

$$p(s', r|s, a) = \Pr\{S_t = s', R_t = r|St - 1 = s, A_{t-1} = a\} \text{ For all } s', s \in \mathcal{S}, r \in \mathcal{R}, a \in \mathcal{A}$$

以上可以解釋為 在state s 使用 action a 的前提下，下一個 state 在 s' 上且獲得 reward r 的機率

所有狀態行動對機率分布總合為1

$$\sum_{s' \in s} \sum_{r \in \mathcal{R}} p(s', r | s, a) = 1, \text{for all } s \in \mathcal{S}, a \in \mathcal{A}(s)$$

在單個馬可洛夫決策過程(single MDP)，機率分布完全取決於環境，也就是說 S_t, R_t 出自於上次 S_{t-1}, A_{t-1}的結果，然而我們應把它視為state而不是決策過程上的限制。

State必須包含所有在前一個state-environment互動的資訊。若是state有上述說的狀況，我們可以說它是有**Markov Property**.

$$p(s'|s, a) = \Pr\{S_t = s'|S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathcal{R}} p(s', r|s, a)$$

我們也可以計算出(s, a)-pair 的期望值

$$r(s, a) = \mathbb{E}[R_t|S_{t-1} = s, A_{t-1} = a] = \sum_{r \in \mathcal{R}} r \sum_{s \in \mathcal{S}} p(s', r|s, a)$$

或是使用三個變數(s, a, s')的函式

$$r(s, a, s') = \mathbb{E}[R_t|S_{t-1} = s, A_{t-1} = a, S_t = s'] = \sum_{r \in \mathcal{R}} r \frac{p(s', r|s, a)}{p(s'|s, a)}$$

Goal and Reward

在RL中 Reward會從Environment回傳給Agent。Reward為常數 $R_t \in R$

我們需要的布是最大化現有的Reward，而是最大化長期累積的Reward。

我們可以定義此獎勵假說：

Goal(目標)的意義可以解釋為最大化累積獲得的Reward的期望值

Return and Episode

從上面所知 Goal 的目標是最大化累積獲得的Reward的期望值

若一連串的Reward可以寫作 $G_t = R_{t+1} + R_{t+2} + R_{t+3} \cdots + R_T$

T作為最終Time step。

若Final Time step是存在的，這個方法自然會覺得合理(並非無限持續的)

我們便可以把一段一段的agent-environment的互動視為一個**Episode**(迭代)

比如說一盤棋的一代就是一個完整的棋局，直到勝負方出現。

每一代最後都會有一個特殊的 State 稱作為**Terminal State.**

我們可以看做所有的Episode最後會結束在Terminal State.

*Episodic Task*顧名思義是每個Episode所需完成的目標

而每代結束所花費的時間time of termination會因為每一代而有所不同.

然而另一個情況是，很多時候並沒有一個Terminal State我們會稱之為*continuing task* 像是新聞氣象預測之類的。

$$T = \infty$$

再來要談的是**Discounting rate(折現率)** $\gamma, 0 \le \gamma \le 1$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

短期的獎勵會比長期的獎勵還要有價值

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \ldots \\ &= R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \ldots) \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned}$$

若該 Episode 在 $t+1$ 終止，我們定義該 $G_t = 0$。

加入discounting factor 可以讓一個 continuing task 的 G_t 收束，舉個例子，如果

$\gamma < 1$ 且reward 一直為 1

$$G_t = \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$$

Unified Notation For Episodic and Continuing Task

我們需要一個統一的方法表示 Episodic and Continuing Task

$$G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

T可以為無限大(continuing task) γ也可以為1(no discounting)

Policies and Value Functions

Policy : 如果Agent遵守 policy π 在時間 t 之下，那 $\pi(a|s)$為在 $S_t = s$ 之下採取 $A_t = a$ 的機率 $\pi(a|s)$中 π 是一般函數，而 $(a|s)$ 為在個別獨立的 state $s \in \mathcal{S}$ 前提下 action $a \in \mathcal{A}(s)$ 的機率分布。

Value Functions : 一估算在以當前狀態來做評分的函式，可以給定state或是state-action對(pairs)，這個"評分優劣"取決於未來的獎勵多寡或式獎勵的期望值。

Value Functions 在 policy π 底下，以 $v_\pi(s)$表示。在MDP的前提下，我們可以定義 $v_\pi(s)$ 如下式

State-value Function :

$$v_\pi(s) = \mathbb{E}\pi[G_t|S_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s], \text{for all } s \ni \mathcal{S}$$

Where $\mathbb{E}_\pi[\cdot]$表示為給定變數於policy π 回傳的期望值， t 可以是任意時間，記得在 t 在Terminal State的期望值為0

另一種形式是，policy π 在給定的 (s, a) pairs 下回傳的期望值

Action-value Function :

$$q_\pi(s,a) = \mathbb{E}\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a]$$

IMPORTANT EXERCISE : 3.12 3.13

3.12 給定 q_π 和 π 求 v_π

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a)$$

(在state s 下選擇個別 action a 的機率) * (action-value)

3.13 給定 v_π 和 π 求 q_π

$$q_\pi(s,a) = \sum_{s',r} p(s',r|s,a)[r+\gamma v_\pi(s')]$$

(所有在status s, action a 在下一步導向 s', r 機率) * (G_t)

---

v_π 及 q_π可以靠著學習經驗去預測，如果agent遵從policy π 並在多個Sample 取得各代的平均，這個平均最終會收束到一個值 $v_\pi(s)$,如果各自的平均取自於在該state所使用的action，最後則是會趨近於 $q_\pi(s,a)$。我們把這種預測方式叫做**Monte Carlo Method**因為他把許多隨機的資料的實際回傳值作平均。

這些value function其實也遵守遞迴關聯性，在任意的 π 及任意的 s 下，State s 會跟接下來的states產生下列的關係。

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t=s] = \mathbb{E}_\pi[R_{t+1}+\gamma G_{t+1}|S_t=s]$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s',r|s,a)\Big[r+\gamma \mathbb{E}_\pi[G_{t+1}|S_{t+1}=s']\Big]$$

$$= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r+\gamma v_\pi(s')] \leftarrow \text{ this is the Bellman Equation for } v_\pi$$

Optimal Policy and Optimal Value Functions

如果我們定義Policy $\pi \geq$ policy π' 如果 π 的期望值大過 π'。換句話說只有在 $v_\pi(s) \geq v_{\pi'}(s)$成立時才會有 $\pi \geq \pi'$ 發生。

然而在眾多 policy 中一定有一個 policy 是比其他 policy 好的， 我們將其稱之為 **Optimal policy**。

Optimal policy會以星號*標示如 $v_*(s), q_*(s,a)$

他們也都有其state-value function 我們將其定義為

$$v_*(s) = \max_\pi v_\pi(s)$$

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

q_* in terms of v_*

$$q_*(s,a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1})|S_t=s, A_t=a]$$

Bellman Equation for v_* 或是 Bellman Optimality Equation

$$v_*(s) = \max_{a\in\mathcal{A}(s)} q_{\pi_*}(s,a) = \max_a \mathbb{E}_{\pi_*}[G_t|S_t=s, A_t=a]$$

$$= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1}|S_t=s, A_t=a]$$

$$= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma v_*(S_{t+1})|S_t=s, A_t=a]$$

$$= \max_a \sum_{s',r} p(s',r|s,a)[r+\gamma v_*(s')]$$

Bellman Optimality Equation of q_*

$$q_*(s,a) = \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1},a')|S_t=s, A_t=a]$$

$$= \sum_{s',r} p(s',r|s,a)[r + \max_{a'} q_*(s',a')]$$

Figure 3.4 畫得很清楚

v_* 為在 state s 之下選擇最大的 action a 或是說 $\underset{a\in\mathcal{A}(s)}{max}$ 並獲得 r 及 s' ，另一個說法就是這個 a 的選定是 **greedy** 的

q_* 則為在 state s 並已經事先選定 action a的情況下，下一代每個有可能會獲得的 s' 中找出各個 s' 可能的 $a' \in \mathcal{A}(s')$之 $\underset{a'}{max}$並乘上 $p(s',r|s,a)$機率分布就可獲得

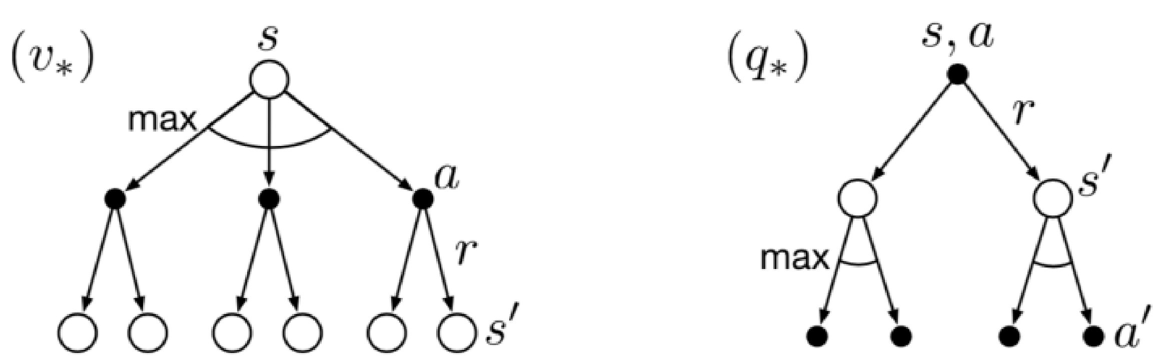


Figure 3.4: Backup diagrams for v_* and q_*