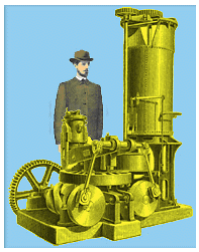
Chapter 6
Robot Learning

6.1 Introduction
6.2 MDP
6.3 Q-Learning

Introduction



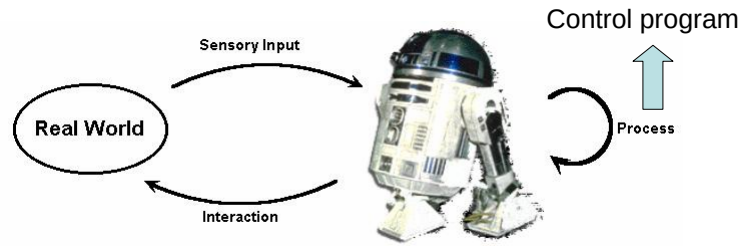
+

=

หุ่นยนต์ = ผู้ช่วยทุ่นแรงกายและแรงสมองของมนุษย์

Control Program

โปรแกรมควบคุมคือข้อกำหนดว่าจะให้หุ่นยนต์ทำอะไรเมื่อใด เพื่อให้งานที่ต้องการสำเร็จผล



ผู้ใช้ต้องตั้งโปรแกรมควบคุมเอง หรือว่ามีวิธีที่หุ่นยนต์จะสร้างโปรแกรมควบคุมได้เอง

ผู้ใช้ต้องตั้งโปรแกรมควบคุมเอง หรือว่ามีวิธีที่หุ่นยนต์จะสร้างโปรแกรมควบคุมได้เอง

นิยาม

- เสตท (state) คือสถานะของหุ่นยนต์ที่ต้องการพิจารณา
- แอคชัน (action) คือการทำงานของหุ่นยนต์ที่เป็นไปได้
- รางวัล (reward) คือค่าสกาลาร์ที่หุ่นยนต์ได้รับหลังจากทำแอคชัน
- โปรแกรมควบคุม (policy หรือ control program) คือข้อกำหนดว่าจะให้หุ่นยนต์ทำแอคชันใดเมื่ออยู่ในเสตทต่างๆ

วงจรชีวิตของหุ่นยนต์

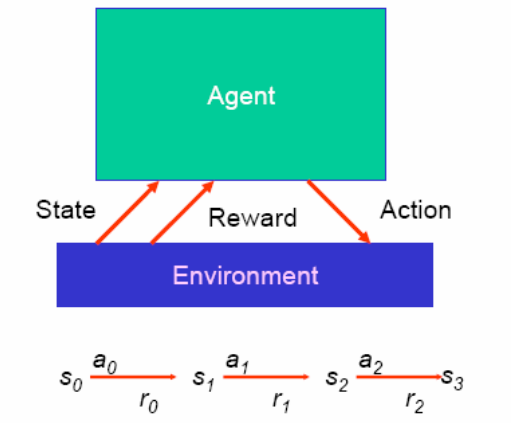
ณ เวลา t สิ่งที่เกิดกับหุ่นยนต์เป็นตามลำดับดังนี้

- หุ่นรับรู้ว่าตัวเองอยู่ที่เสตท $s_t \in S$
- หุ่นเลือกทำแอคชัน $a_t \in A$
- สิ่งแวดล้อมสนองตอบการกระทำ a_t ด้วยรางวัล $r_t = r(s_t, a_t)$
- และหลังจากได้รับรางวัล หุ่นจะไปอยู่ที่เสตท $s_{t+1} = \delta(s_t, a_t)$

MDP

- สิ่งแวดล้อมเป็น MDP (Markov Decision Process) ถ้า ฟังก์ชัน r และ δ ขึ้นกับเสตทปัจจุบัน ไม่ขึ้นกับเสตทในอดีต
- ในบทนี้เราจะพิจารณาเฉพาะปัญหาที่สิ่งแวดล้อมมีคุณสมบัติเป็น MDP เท่านั้น ตัวอย่างปัญหาเช่น การเล่นหมากรุก
- เป้าหมายคือต้องการหาโปรแกรมควบคุม $\pi : S \mapsto A$ ที่ทำให้หุ่นยนต์สะสมรางวัลได้มากสุด

MDP



Problem Definition

- กำหนดให้ $V^{\pi}(s_t)$ คือรางวัลสะสมที่จะได้จากการอยู่ที่เสตท และทำตามโปรแกรมควบคุม π
- ขอเรียก $V^{\pi}(s_t)$ ว่ามูลค่า (value) ของเสตท s_t
- โปรแกรมควบคุมที่ต้องการคือ π^* ซึ่งเป็นโปรแกรมที่ทำให้ได้มูลค่าของแต่ละเสตทสูงสุด หรือเขียนได้ว่า $\pi^* \equiv \arg\max_{\pi}(V^{\pi}(s)), \forall s$

วิธีคิดมูลค่า

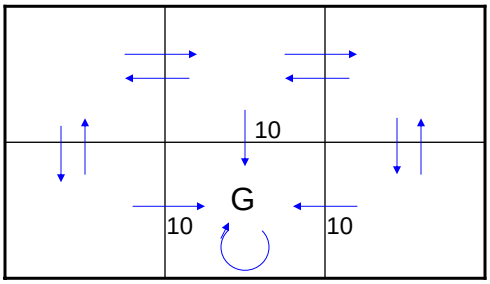
การคิดมูลค่าแบบลดทอน (discounted accumulative reward) กำหนดให้

$$V^{\pi}(s_t) = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

โดย γ เป็นค่าคงที่บวกไม่เกิน 1

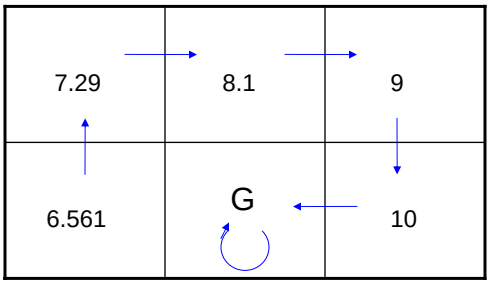
Example

กำหนดให้สิ่งแวดล้อมมี 6 เสตท แต่ละเสตทมีแอคชันดังรูป โดยมีเพียง 3 แอคชันเท่านั้นที่มีรางวัล กำหนดให้ $\gamma = 0.9$



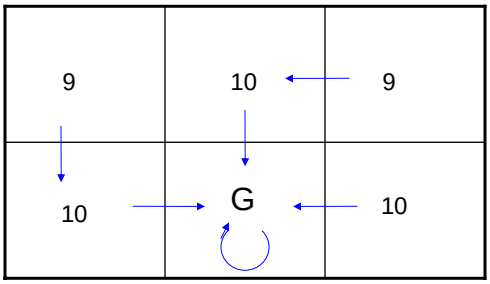
Example

สำหรับโปรแกรมควบคุม π ดังรูป มูลค่าของแต่ละเสตทจะเป็นไปตามที่ระบุในแต่ละช่อง



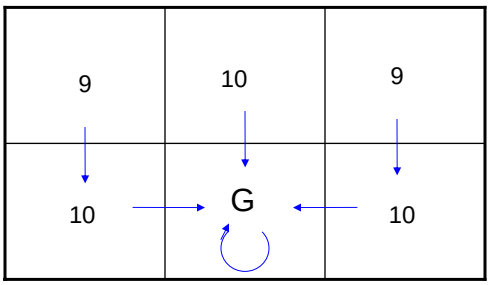
Example

โปรแกรมควบคุม π^* ที่ดีสุด



Example

อีกโปรแกรมควบคุม π^* ที่ดีสุด



Q Learning: Background

แล้วให้หุ่นยนต์เรียนรู้อะไร ?

Idea: คำนวณ V^{π^*} (เขียนย่อว่า V^*) แล้วจะหา π^* ได้ดังนี้

$$\pi^*(s) = \arg\max_a [r(s,a) + \gamma V^*(\delta(s,a))]$$

แต่นั่นคือเราต้องรู้ δ และ r

Q Learning: Derivation

หากเขียนใหม่ในรูปของมูลค่าของแอคชันที่แต่ละเสตท จะได้

$$Q(s,a) = r(s,a) + \gamma V^*(\delta(s,a))$$

ดังนั้นหากคำนวณฟังชัน Q ได้แล้ว เราก็จะสามารถหา โปรแกรมควบคุมดีสุดได้ด้วย

$$\pi^*(s) = \arg\max_a Q(s,a)$$

โดยไม่ต้องรู้ r และ δ

Q Learning: Concept

แล้วหา Q ได้อย่างไร ?

เขียน V^* ในรูปของ Q ได้คือ

$$V^*(s) = \max_{a'} Q(s, a')$$

จึงเขียน

$$Q(s,a) = r(s,a) + \gamma V^*(\delta(s,a))$$

ได้เป็น

$$Q(s,a) = r(s,a) + \gamma \max_{a'} Q(\delta(s,a), a')$$

Q Learning: Algorithm

For each s,a initialize the table entry $\hat{Q}(s,a)$ to zero

Observe the current state s

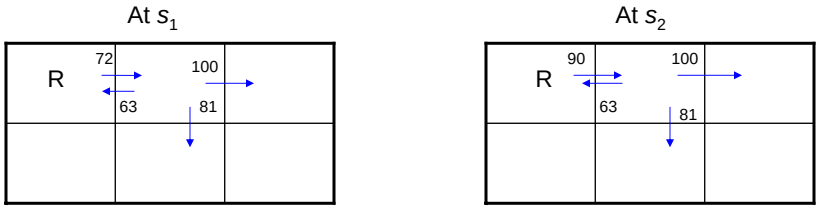
Do forever

- Select an action a and execute it
- Receive immediate reward r
- Observe the new state s'
- Update the table entry for $\hat{Q}(s,a)$ as follows:

$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

- $s \leftarrow s'$

Example



ประมาณค่าของ Q ที่เปลี่ยนไปจาก 72 เป็น 90 หลังจากการเดินไปทางขวาของ R และการ update ตาม Q Learning algorithm (ค่าที่ลูกศรเป็นค่าประมาณของแต่ละแอคชัน)

$$\begin{aligned}\hat{Q}(s_1, a_{right}) &\leftarrow r + \gamma \max_{a'} \hat{Q}(s_2, a') \\ &\leftarrow 0 + 0.9 \max\{63, 81, 100\} \\ &\leftarrow 90\end{aligned}$$

Q Learning: Convergence

ให้ $\Delta_n \equiv \max_{s,a} |\hat{Q}_n(s,a) - Q(s,a)|$ คือ maximum error ที่ update ครั้งที่ n

จะได้ว่า

$$\begin{aligned}|\hat{Q}_{n+1}(s,a) - Q(s,a)| &= |(r + \gamma \max_{a'} \hat{Q}_n(s',a')) - (r + \gamma \max_{a'} Q(s',a'))| \\ &= \gamma |\max_{a'} \hat{Q}_n(s',a') - \max_{a'} Q(s',a')| \\ &\le \gamma \max_{a'} |\hat{Q}_n(s',a') - Q(s',a')| \\ &\le \gamma \max_{s'',a'} |\hat{Q}_n(s'',a') - Q(s'',a')| \\ &\le \gamma \Delta_n\end{aligned}$$

นั่นคือ เมื่อ $n \to \infty$ แล้ว $\Delta_n \to 0$

Q Learning: Action Selection

$$P(a_i \mid s) = \frac{k^{\hat{Q}(s,a_i)}}{\sum_j k^{\hat{Q}(s,a_j)}}$$