20 Scaling up: Partial observability

What if the current state is uncertain?

- Represent knowledge of current state by a *probability distribution* over possible current states
  - called a *belief state* $B(s)$
- That is, we encounter three different states at each moment
  - world state (true state of world)
  - perceptual state (state immediately perceived by senses)
  - belief state (state of information about true state of world — combines perception and memory)
- Perception of current state can alter belief state
- Memory can be used to further disambiguate current state
- Optimal actions now depend on belief state, not true state

  $$\pi : B(s) \mapsto a$$

  In this case, the policy is not defined as a function that maps a state to action, but a function that maps a belief state to action

  Equivalently, optimal policy can keep a memory of past states to disambiguate current state
- Can improve expected reward in two ways:
  - Act to achieve rewards
  - Act to reduce uncertainty about current state
    (can attain better performance if have a better idea about current world state)

20.1 Value of Information

Consider the expected future one step reward from a current state s

$$U(s,a) \;=\; \sum_{s'} \mathrm{P}(s'|s,a)\; R(s')$$

The expected future one step reward given a *belief* state B is given by

$$\begin{aligned} U(B,a) &= \sum_{s} B(s)\; U(s,a) \\ &= \sum_{s} B(s)\; \sum_{s'} \mathrm{P}(s'|s,a)\; R(s') \end{aligned}$$

To decide whether to take an action that maximizes immediate reward, or instead execute a "sensing" action (which will provide information about the current state of the world s), one first needs to know how to calculate the expected value of executing the sensing action. This is called the *value of information.*

For example, a sensing action might tell you whether the true world state s is in a restricted set I or not; that is, whether $s \in I$ or $s \in S \setminus I$ for some subset I of the set of possible states S. What is the value of knowing whether $s \in I$ or $s \in S \setminus I$?

Value of information

$$\begin{aligned} &= \text{expected reward of optimal actions given information} \\ &- \text{expected reward of optimal action not given information} \\[1em] &= B(I)\;\max_a\; U(\,B(s\,|\,I),\, a\,) \;\;+\;\; B(S \backslash I)\;\max_a\; U(\,B(s\,|\,S \backslash I),\, a\,) \\ &- \max_a\; U(B(s),a) \end{aligned}$$

Note It is obvious that the value of information is always nonnegative because one gets to choose a different action in each different states subsets (I versus $S \backslash I$) in the informed case, but can only choose a single action in the uninformed case.

20.2 Example: Let's Make a Deal!

This was a popular game show in the 1970's hosted by (an expatriate Canadian) Monte Hall.

- Monte shows you three doors, numbered 1, 2, 3.
- One door contains a prize of $9,999
  (Gee, it was only the 70's after all. That was a lot of money back then.)
- The other two doors contain $∅
- You pick a door, say $a \in \{1, 2, 3\}$
- Monte then reveals one of the other doors $c \in \{1, 2, 3\}$, $c \neq a$, *that does not contain the prize.*
  (Sneaky Monte knows which door contains the prize)
- Monte then asks you: Do you want to switch your choice a for the remaining door b, or do you just want to stay with a?
- Do you switch?

We can use the value of information to determine what you should do.

Mathematical model

- State = location of prize (1,2, or 3)

  To you, this is a random variable S
- Belief state $\quad B(S = 1) \ = \ B(S = 2) \ = \ B(S = 3) \ = \ \frac{1}{3}$
- You pick a door $\quad a =$ 1,2, or 3
- Reward (depends on action a as well as true underlying state S)

$$R(S, a) \quad = \quad \begin{cases} 0 & \text{if } a \neq S \\ 9999 & \text{if } a = S \end{cases}$$

The expected reward of any action given no additional information is

$$\begin{aligned} U(B(s), a) &= \sum_s B(S = s)\ R(s, a) \\ &= \frac{1}{3} 9999 \ = \ 3333 \end{aligned}$$

But now note that Monte gives you extra information! That is, after you pick a, Monte reveals one of the remaining doors, c, that does not contain the prize. Both this door c and the door b that remains are affected by your choice a and by the true location of the prize, S.

$$\begin{aligned} c(a, S) &= s' \quad \text{for some } s' \neq S, s' \neq a \\ b(a, S) &= \begin{cases} S & \text{if } S \neq a \\ s' \text{ for some } s' \neq S, s' \neq a & \text{if } S = a \end{cases} \end{aligned}$$

What is the new belief state given the information that $c \neq S$?

$$\begin{aligned} B(S = b(a, S)) &= B(S = b(a, S) | S = a)\ B(S = a) \\ &+ B(S = b(a, S) | S \neq a)\ B(S \neq a) \\ &= 0 + \frac{2}{3} \\[1em] B(S = a) &= \frac{1}{3} \end{aligned}$$

What is the expected reward of picking the other door $b(a, S)$?

$$\frac{2}{3} 9999 \ = \ 6666$$

What is the expected reward of staying with your first choice a?

$$\frac{1}{3} 9999 \ = \ 3333$$

What is the value of information of knowing that $c(a, S)$ does not contain the prize?

$$6666 - 3333 \ = \ 3333$$

You should definitely switch!
By switching to door b your expected winnings go from \$3,333 to \$6,666. In fact, the value of the information that Monte is giving you is \$3,333, so you should be willing to *pay* him \$3,332 to reveal door c for you (this way you would still make \$1 — on average!)

20.3 Optimal sequential behavior

Calculating optimal behavior strategies in partially observable environments is *hard*

$$\pi : \underbrace{\text{belief state}}_{\text{uncountably many}} \mapsto \text{action}$$

Can disambiguate current state by keeping a history

Optimal policy combines history (memory) with current observation (perception) to choose optimal action

20.4 Scaling up: Robotics, control, games

Have to cope with

- Partial observability
  - use perception and memory
- Complex state spaces
  - use selective attention
- Large state spaces
  - Cannot pre-compute optimal action for every state
  - Could try to find compact representation for approximately optimal policy
  - Or, calculate optimal actions on-line for states that are reached. This interleaves thinking and acting (combine behavior and search). E.g. Computer chess: combine heuristic evaluation $\hat{V}^*(s)$ with depth limited search to choose actions while you play by looking ahead

Section 25.4 in the text gives some general tools and ideas for building agents that might behave successfully in complex, uncertain domains.

Readings

Russell and Norvig: Sections 6.3, 16.6, 25.4
Dean, Allen, Aloimonos: Section 8.4