Machine, Data and Learning

POMDP Basics

MDP vs. POMDPs

- **MDP:** Agent’s percept in any given state identify the state that it is in, e.g., state (4,3) vs (3,3)
  - Given observations, uniquely determine the state
  - Hence, we will not explicitly consider observations, only states
- **POMDP:** Agent’s percepts in any given state **DO NOT** identify the state that it is in, e.g., may be (4,3) or (3,3)
  - Given observations, not uniquely determine the state
  - POMDP: Partially observable MDP for inaccessible environments

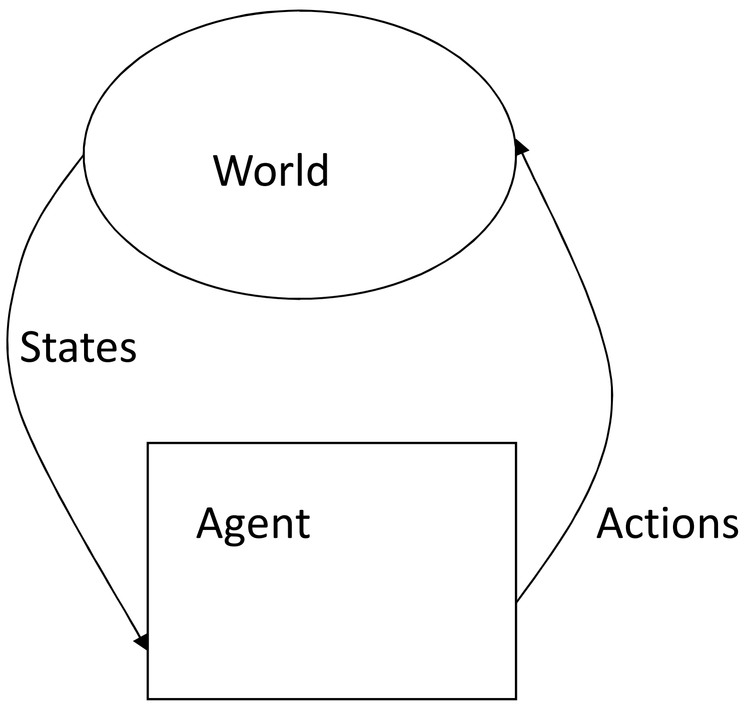
POMDP: Partially Observable Markov Decision Process

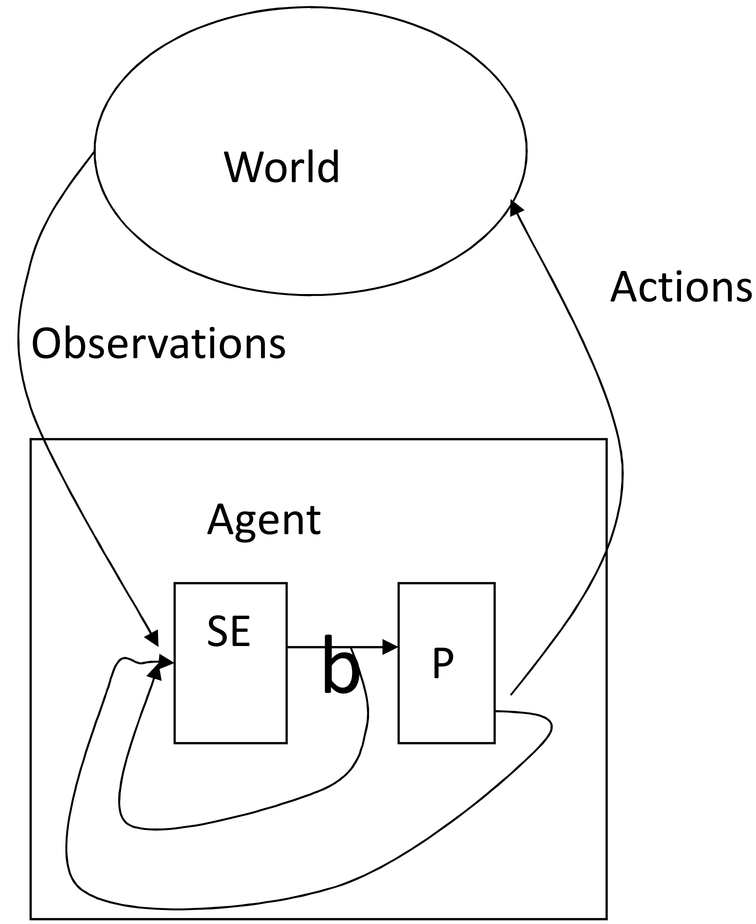
- Set of states, **S**
- Set of actions, **A**
- **P** is the table of transition probabilities
- **R(s,a)** reward received for taking action "a" in state "s"
- **Policy** π maps a **state "s"** to an **action "a"**

- **PLUS**
  - Finite set Ω of observations
  - Table **O** of observation probabilities where **O(o|a,s')** is the probability that "o" is observed given that action "a" taken leads to state s'
  - **Policy** maps **histories of observations** to **actions**

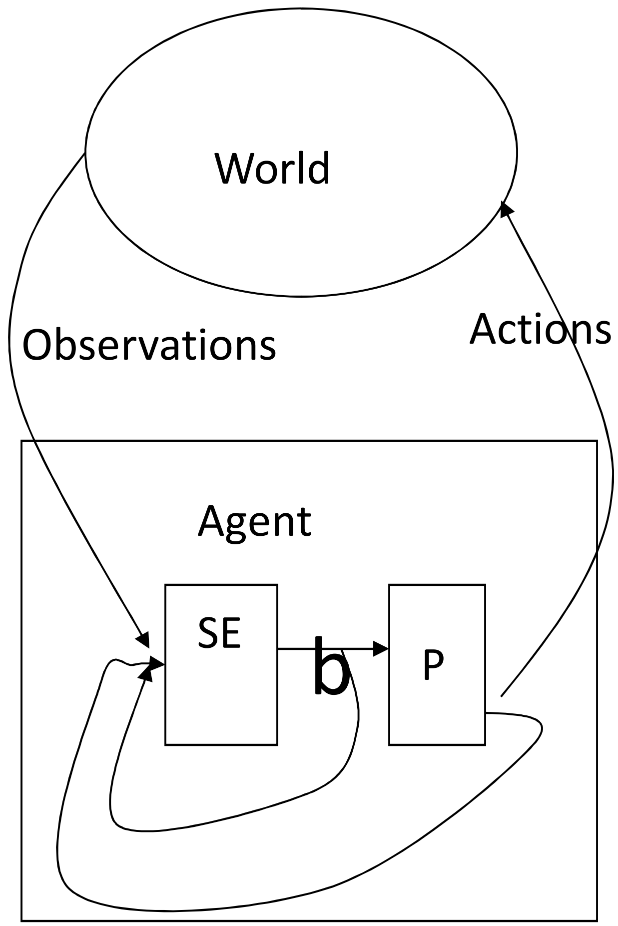
MDP vs POMDP

MDP





POMDP



SE: State estimator

b: Belief state

SE updates the beliefs
based on last observation,
previous belief state
and previous action

P: Policy is no longer a function of the state,
But of the agent’s belief state

POMDP:**<S, A, P, R, Ω, O>**

- **S,** Set of states
- **A,** finite set of actions
- **P** is the table of transition probabilities
- **R(s,a)** reward received for taking action “a” in state “s”
- Finite set Ω of observations, e.g., *{red, green} in example below*
  - Observations hint at state, e.g., *Observe Red room, but not S3*
- Table **O** of observation probabilities
  - ***O(o|a,s’)*** *prob “o” observed given action “a” leads to state s’*
  - *P(red | LEFT, S3) = 0.4*

POMDP: Partially Observable Markov Decision Process

- Agent has initial beliefs
- Agent takes an action
- Gets an observation
- Interprets the observation
- Updates beliefs
- Decides on an action
- Repeats

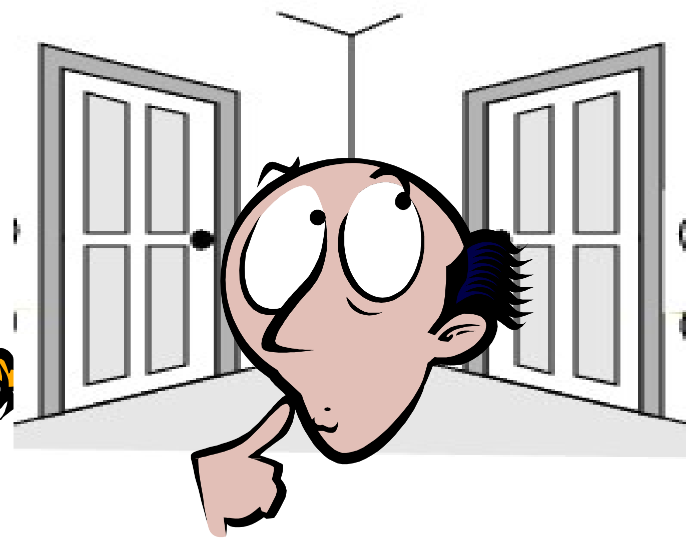
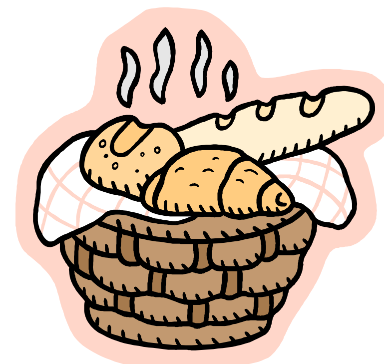
Agent takes optimal action considering world/other agents

Elements: {States, Actions, Transitions, Rewards, Observations }

POMDP: Partially Observable Markov Decision Process

- Underlying dynamics are still **Markovian**: World has **NOT** changed its characteristics, agents sensors have changed
- Observations only hint at what state we are in, but not exactly identify state
- So, somehow agent may need to remember what it observed in the past and what action it took:
  - *If I observed feature "green" in the past, then took action "left" and then observed "red", it must mean that I am either in state S3 (probability of 0.9) or S2 (Prob 0.1) now*
- *Need to maintain beliefs*

Tiger Problem

- Standing in front of two closed doors
- World is in one of two states: tiger is behind left door or right door
- Three actions: Open left door, open right door, listen
    - *Listening is not free, and not accurate (may get wrong info)*

- Reward: Open the wrong door and get eaten by the tiger (large –ve)
  Open the right door and get a prize (small +ve)

Tiger Problem: POMDP Formulation

- Two states: SL and SR
- Three actions: LEFT, RIGHT, LISTEN
- Transition probabilities:

Listen	SL	SR
SL	1.0	0.0
SR	0.0	1.0

Left	SL	SR
SL	0.5	0.5
SR	0.5	0.5

Right	SL	SR
SL	0.5	0.5
SR	0.5	0.5

Tiger Problem: POMDP formulation

- Observations: TL (tiger left) or TR (tiger right)
- Observation probabilities:

Listen	TL	TR
SL	0.85	0.15
SR	0.15	0.85

Left	TL	TR
SL	0.5	0.5
SR	0.5	0.5

Right	TL	TR
SL	0.5	0.5
SR	0.5	0.5

- Rewards:
  - *R(SL, Listen) = R(SR, Listen) = -1*
  - *R(SL, Left) = R(SR, Right) = -100*
  - *R(SL, Right) = R(SR, Left) = +10*

How to Find the Optimal Policy?

- Now lets find an optimal policy for this problem

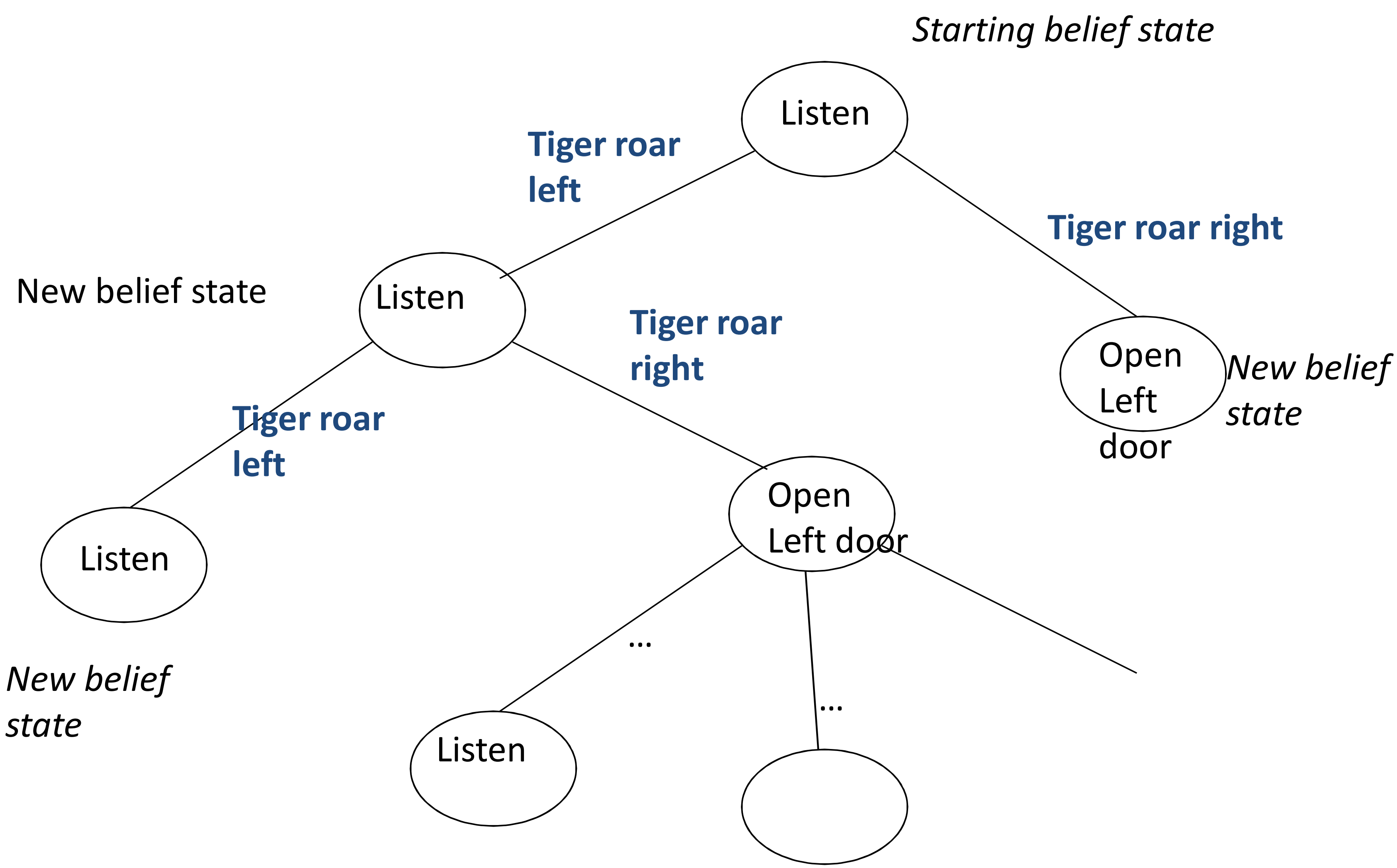
- Why not use value iteration directly?
  - $U_{t+1}(I) = R(I) + \max_{A} \sum_{J} P(J|I,A) * U_t(J)$

- Could we compute the utilities in this manner?

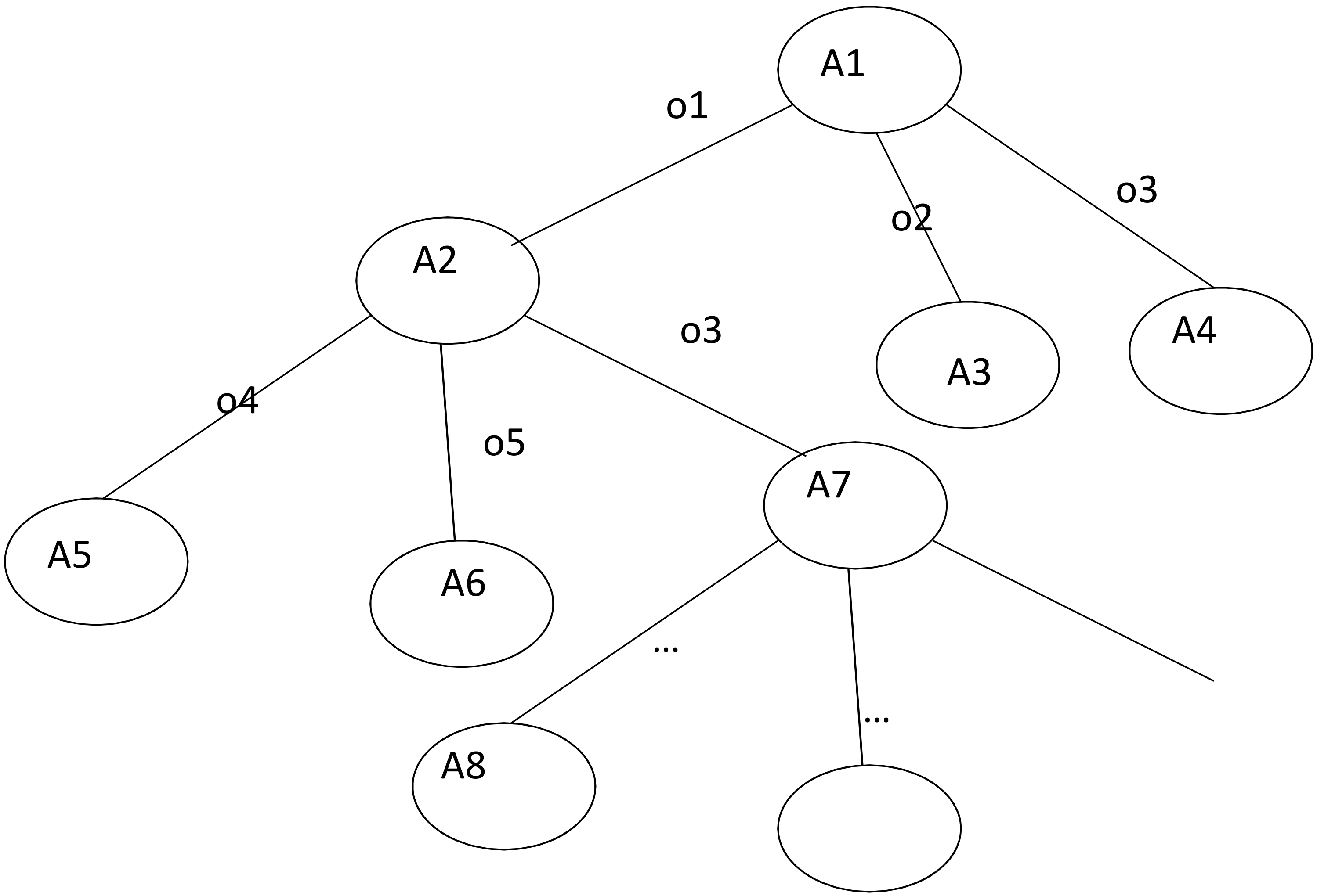
- Could such utilities be actually used?

- Need mapping from belief states to actions!

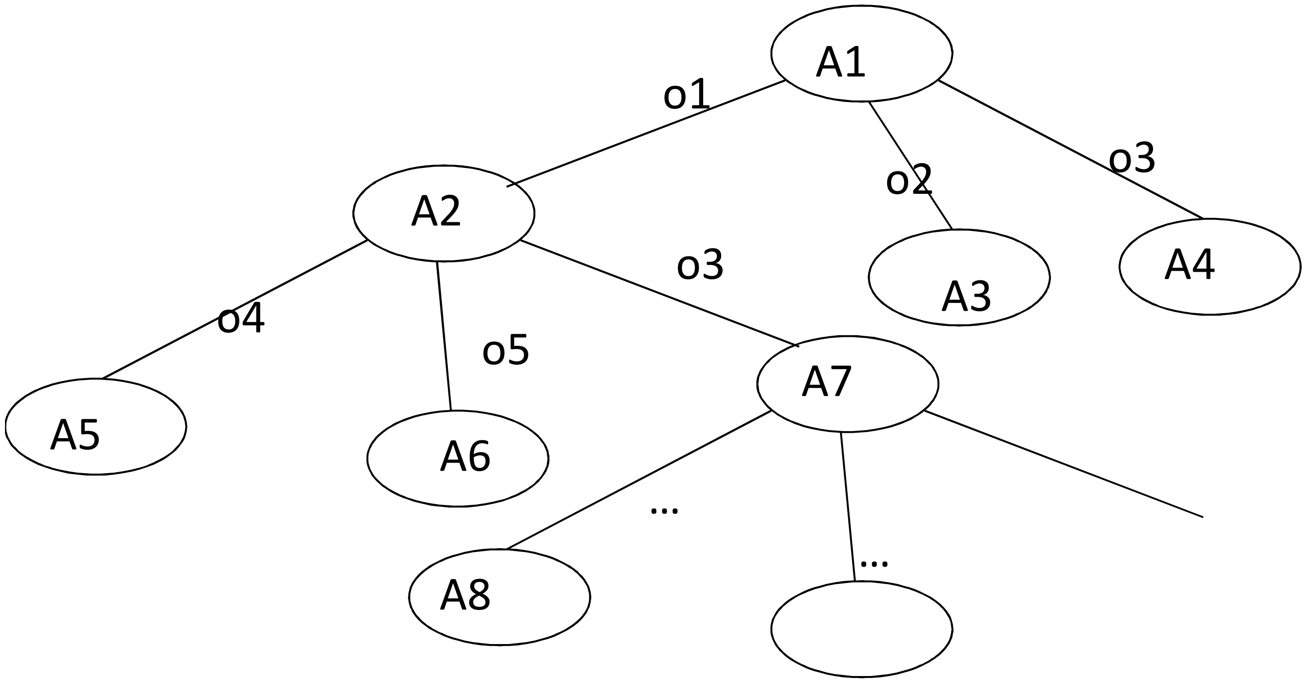
Sample POMDP Policy Tree



POMDP Policy Tree



How Many POMDP policies possible



1

$|O|$

$|O|^2$

...

How many policy trees, if $|A|$ actions, $|O|$ observations, T horizon:

- How many nodes in a tree:

$$N = \sum_{i=0}^{T-1} |O|^i = (|O|^T - 1) / (|O| - 1)$$

How many trees:

$$|A|^N$$

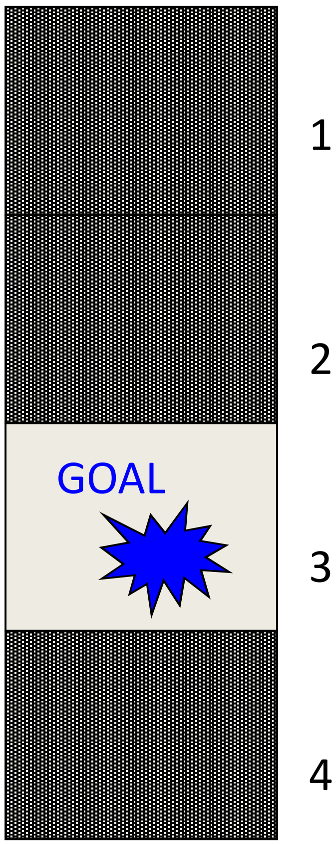
POMDP Belief State

- Computing belief state important, since policy maps belief state to action
  - *Not just the most probable state of the world*

- Probability distributions over the states of the world
  - *Sufficient statistic* for the past history and initial belief state:
    - *No additional data about past actions & observations supplies any further information*

  - *That is, process over belief states is a markov process (why?)*
  - *As if maintained a complete history of actions & observations*

Evolution of Belief State: 1



- Set of states, S1, S2 S3, S4
- For each $s \in S$, A_s set of actions: Down or Up
- Transition Prob T: 0.9 (direction of move), 0.1 opp
- R(s,a) reward received

– Finite set Ω of observations

– O observation probabilities:

  – $Pr(o1|s1) = Pr(o1|s2) = Pr(o1|s4) = 1$

  – $Pr(o2|s3) = 1$

Initial belief state: [0.333, 0.333, 0, 0.333]

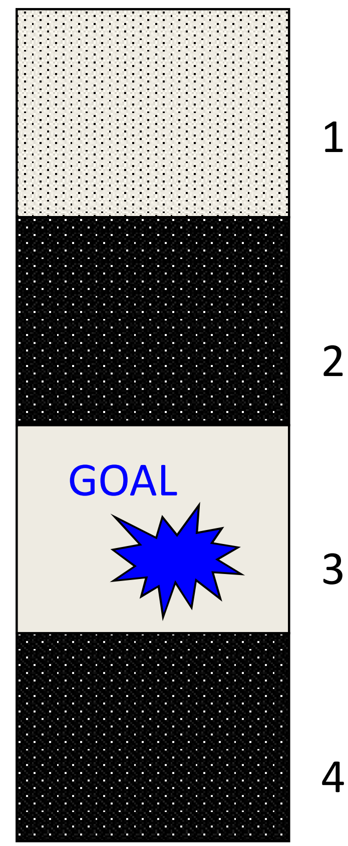
S1 S2 S3 S4

Evolution of Belief State: 2

Suppose agent moves down and observes o1:

• What should the agent believe about its state? Does it now know more about where it is more likely to be?

• [0.100, 0.450, 0, 0.450]



Belief State

- b → probability distribution over our set of states, e.g., over s1, s2, s3, s4
- b(s) denotes the probability assigned to world state s by belief state b
- *In [0.333, 0.333, 0, 0.333], what is b(s1)?*
- 0 <= b(s) <= 1

$$\sum_{s \in S} b(s) = 1$$

Computing Belief States

- s = old state
- b = old belief state, and b(s) probability of s given belief state b
- a = action
- b’ = new belief state
- b’(s’) = probability of s’ given b’
- o = observation

Computing Belief States

$$b'(s') = \Pr(s' \mid o, a, b) = \Pr(s' \wedge o \wedge a \wedge b) / \Pr(o \wedge a \wedge b)$$

$$= \frac{\Pr(o \mid s', a, b) \Pr(s' \mid a, b) * \Pr(a \wedge b)}{\Pr(o \mid a, b) * \Pr(a \wedge b)}$$

$$= \frac{\Pr(o \mid s', a) \Pr(s' \mid a, b)}{\Pr(o \mid a, b)}$$

Will not repeat $\Pr(o \mid a, b)$ in the next slide, but it is there!

– *Treated as a normalizing factor, so that b' sums to 1*

Computing Belief States: Numerator

$= Pr(o \mid s' \, a) \, Pr\,(s' \mid a, b) \;\; = O(s', a, o) \, Pr\,(s' \mid a, b)$

$= O(s', a, o) \sum Pr\,(s' \mid a, b, s) \, Pr\,(s \mid a, b)$

$= O(s', a, o) \sum Pr\,(s' \mid a, b, s) \, b(s) \qquad ; Pr\,(s \mid a, b) = Pr\,(s \mid b) = b(s)$

$= O(s', a, o) \sum T(s, a, s') \, b(s)$

(Please work out some of the details later)

Belief State

Overall formula

$$= \frac{O(s', a, o) \sum T(s, a, s') \, b(s)}{Pr\,(o \mid a, b)}$$

Example

Moves down and does not observe s3

- b → b'
- i.e., [0.333, 0.333, 0, 0.333] → [0.1, 0.45, 0, 0.45]

b'(s1) = probability of s1 in our new belief state b'

Numerator = Pr (o1 | s1, down)) *

[Pr(s1 | s1, down) * b(s1) + Pr (s1 | s2, down) * b(s2) +

Pr (s1 | s3, down) * b(s3) + Pr (s1 | s4, down) * b(s4)]

= 1 * [0.1 * 0.333 + 0.1 * 0.333 + 0 + 0] = 0.0666

Why is this not 0.1?

Example

Moves down and observes o1 (I.e., not observe s3)

- b → b'
- i.e., [0.333, 0.333, 0, 0.333] → [0.1, 0.45, 0, 0.45]

In b'(s1), *Numerator* = 0.0666

The above is unnormalized probability, hence not 0.1!

Denote unnormalized b'(s1) as Ub'(s1)

- Similarly calculate unnormalized Ub'(s2), Ub'(s3), Ub'(s4)
- [Ub'(s1) + Ub'(s2) + Ub'(s3) + Ub'(s4)]/denominator = 1
- Denominator = 0.666 (please check at home)
- b(s1) = 0.0666/0.666 = 0.1

Policy: Map Belief State to Action

Convert POMDP → "Belief MDP"

- *Recall, process over belief states is markov*
- B, the set of belief states, is the set of MDP states
- A, the set of actions, is the same
- R'(b,a) is the reward function on the belief states:

$$R'(b, a) = \sum_{s \in S} b(s)\, R(s, a)$$

- Transition function:

$$T(b, a, b') = Pr(b' \mid a, b) = \sum_{o \in \Omega} Pr(b' \mid a, b, o) * Pr(o \mid a, b)$$

Where $Pr(b' \mid b, a, o) = 1$ *if* $SE(b, a, o) = b'$

$= 0$ *otherwise*

Transition Function

<u>Note: observe-s3 = o2, not(observe-s3) = o1</u>

E.g., T([0.330, 0.330, 0, 0.330], down, [0.1, 0.45, 0, 0.45])

= *Pr (b' | down, b, observe-s3) * pr (observe-s3 | down, b)*

*+ Pr (b' | down, b, Not(observe-s3)) * pr (Not (observe-s3)| down, b)*

= 0 * pr (observe-s3 | down, b) + 1 * pr (Not(observe-s3) | down, b)

= Pr (*Not(observe-s3)* | down, b) = 0.666

T(b, a, b') = Pr (o | a, b) where when action "a" taken in belief-state "b" and we observed "o", we ended up in belief-state b'

Try Value Iteration

- Given belief MDP, if we can generate an optimal policy, it will give rise to optimal behavior for the original POMDP
- How about trying value iteration in this belief MDP?

$$V'(b) = \max [\, r(b,a) + \Sigma P(b' \mid b,a)\, V(b')]$$
$$= \max [\, r(b,a) + \Sigma P(o \mid b,a)\, V(b^a{}_o)]$$

- *Where $r(b, a) = \Sigma r(s,a)b(s)$ is the expected immediate reward for taking action a in belief state b*
- *o is the observation, $P(o|b, a)$ implies the probability of observing o given action a in belief state b*
- *$V(b^a{}_o)$ denotes the value for belief state at the next point in time given that action a was taken in belief state b, with observation o*

Problem in Value Iteration

- Infinite possible belief states:
  - Assume: we don't have a fixed start belief state
  - *MDP has a continuous state space*
  - *No longer a table of states where we can maintain a value per state*
- Also, how to back up values of future belief states --- there are too many (infinite) future belief states as well