Partially Observable Markov Decision Processes

3/3/17

(Dis)Advantages of Online MCTS

+ Just like in game playing, MCTS handles high branching factors very well.
+ No training phase is required.
– Each move takes a long time.
– We're back to an un-factored MDP, so we can't directly do approximate Q-learning.
  + Online MCTS and function approximation can be combined.
  – That combination is beyond scope for this class.

Discussion: compare online MCTS and approximate Q-learning. When should we prefer each?

Observability

- The MDP model allows for noisy transitions.
- It still assumes the agent always knows everything relevant about the world.
  - The agent can always tell exactly what sate it’s in.

What if there are features of the environment that are definitely relevant to decision making, but aren't directly observable to the agent?

- Name some environments where this can happen.

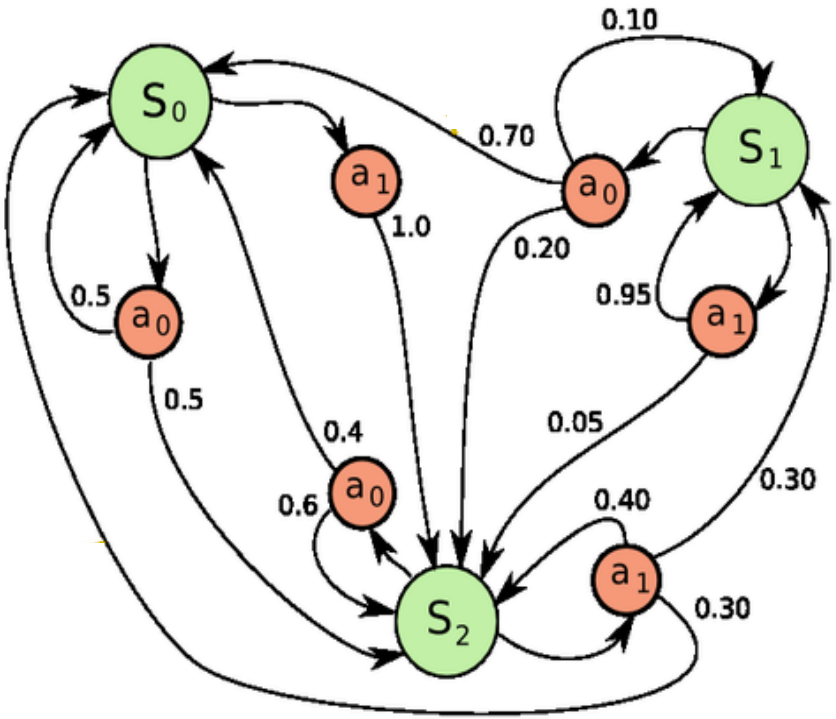
MDPs vs POMDPs

In an MDP, the agent always knows its state.

In a **PO**MDP, the state is **<u>partially observable</u>**.

The agent believes some probability distribution over what state it's in. eg:

$P(S_0, S_1, S_2) = \langle 0.45, 0.55, 0.0 \rangle$



Optimal Policy in a POMDP

In an MDP, if we know the value of every state, the optimal policy picks the best action in expectation:

$$\arg\max_a \left[R(s,a) + \sum_{s'} P(s'|s,a)V(s') \right]$$

In a POMDP, we need to extend the EV calculation to our uncertainty over states:

$$\arg\max_a \left[\sum_s P(s) \left(R(s,a) + \sum_{s'} P(s'|s,a)V(s') \right) \right]$$

belief

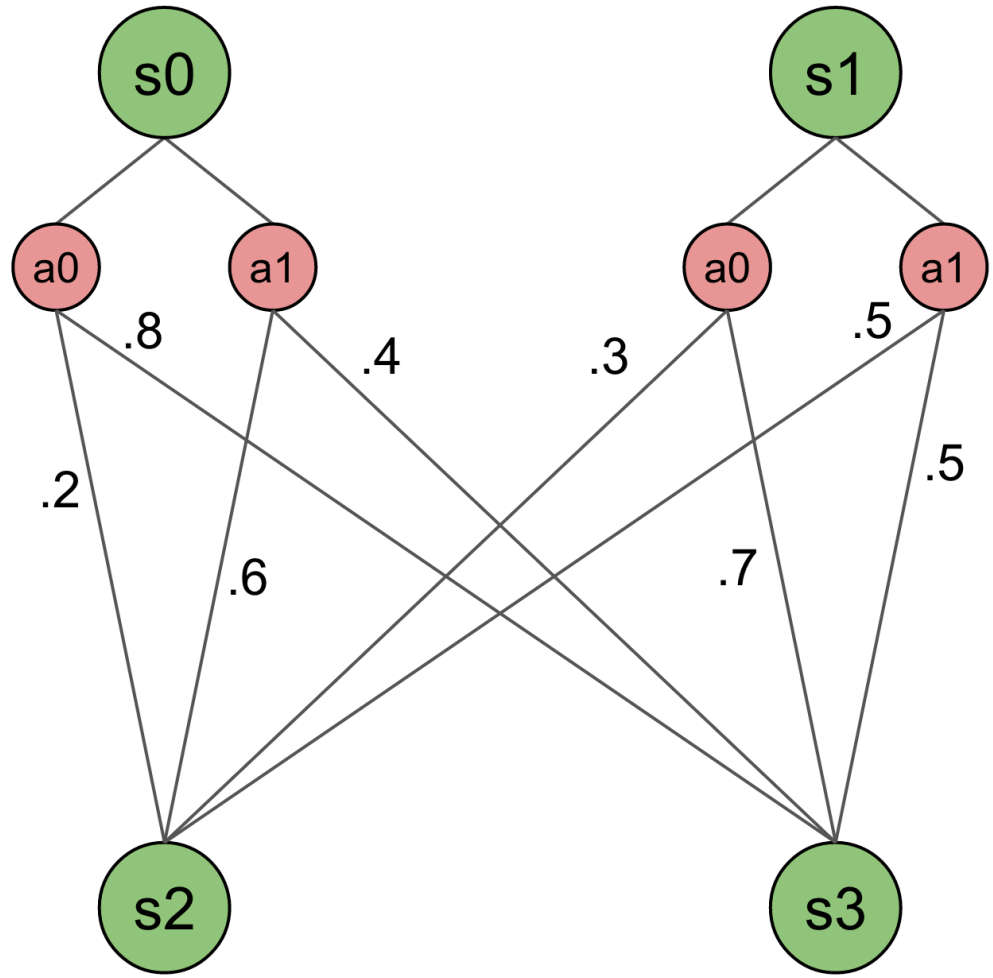
transition probability

Exercise: compute action EVs

```
R(s0, a0) = 0
R(s0, a1) = 1
R(s1, a0) = 2
R(s1, a1) = -1

P(s0) = .25
P(s1) = .75

V(s2) = 3
V(s3) = 4
```



$$\arg\max_a \left[\sum_s P(s) \left(R(s,a) + \sum_{s'} P(s'|s,a) V(s') \right) \right]$$

Updating Beliefs

- The agent may get observations that change its beliefs about the probability of each state.

- For example, if we see a the blue ghost down a corridor, all states where the blue ghost is elsewhere now have probability 0.

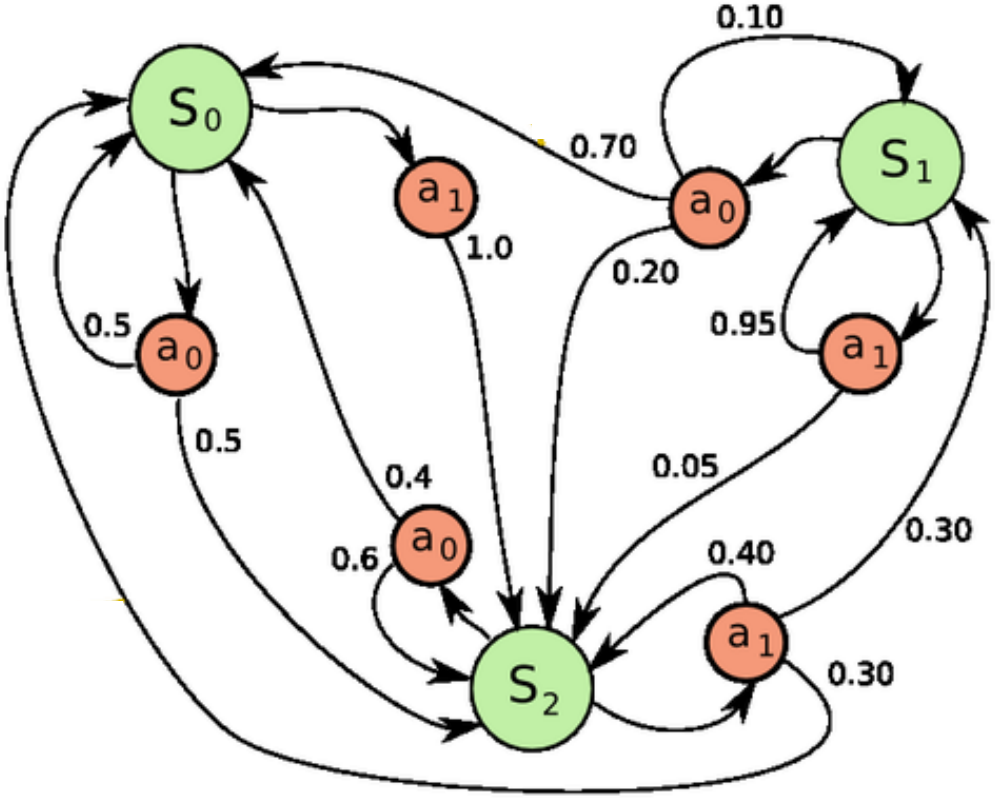
- Each step, the agent gets an observation and updates its beliefs.

Exercise: what is the belief distribution?

Initial distribution: ⟨0.4, 0.3, 0.3⟩

Action: a0

Observation: not in S_1



Converting POMDPs to MDPs

In a POMDP:

- Action + observation updates beliefs
- Value is a function of beliefs.

Instead we can view this as an MDP where:

- There is a state for every possible belief.
- Beliefs are probabilities, so we have a continuum.
  - There are infinitely many belief-states.
- Taking an action transitions to another belief-state.
  - Observations are random, so this transition is random.

Value Iteration in POMDPs

Value iteration in a finite MDP:

1. Initialize each state's value to 0.
2. Compute the greedy policy for each state.
3. Update the value of each state based on this policy.
4. Goto step 2; repeat until converged.

In a POMDP, there are infinitely many states.

- We can't loop through them.
- Value is a piecewise-linear function of belief.
  - We can do value iteration over a finite set of linear functions.
    - This algorithm is described in the optional reading.

Connect Four Tournament

Rank	names	wins	draws
1	**dboshko1-tfeldma1**	**114**	**7**
2	**slim1-tchen2**	**102**	**2**
3	**jye1**	**98**	**7**
4	**swallac3-nhoang1**	**100**	**0**
5	mparker3-mbaer1	90	5
6	apowell1-hyan1	86	5
7	tkyaw1-lbrumga1	81	14
8	jhan2-schen3	81	3
9	azhao2-sfischm1	80	4
10	smalawi1	75	12
11	rhiggin1-nfeldba1	72	7
12	swang5-zzhao1	70	7
13	dmin1-mriley1	67	7
14	yhigash1-msong2	64	6
15	amansar1-cpillsb1	58	9
16	jnovak1-twarner2	56	4
17	kyee1-bchen6	52	7
18	eliu2-itang1	52	0
19	aabitin1-lceball1	20	0
20	asiegel1-jshah1	15	0
21	gbarret1-zliu1	10	0
22	jlee5	10	0
23	dholmgr1-cllop1	8	0

Semifinal with w=7, h=6, c=4, t=5:

```
jye1/dboshko1-tfeldma1:                0-2-0
slim1-tchen2/swallac3-nhoang1:         1-1-0
jye1/swallac3-nhoang1:                 1-1-0
jye1/slim1-tchen2:                     1-1-0
dboshko1-tfeldma1/slim1-tchen2:        1-1-0
dboshko1-tfeldma1/swallac3-nhoang1:    1-1-0
```

Semifinal with w=8, h=8, c=4, t=10:

```
jye1/dboshko1-tfeldma1:                1-1-0
dboshko1-tfeldma1/swallac3-nhoang1:    1-1-0
jye1/slim1-tchen2:                     1-1-0
jye1/swallac3-nhoang1:                 1-1-0
slim1-tchen2/swallac3-nhoang1:         1-1-0
dboshko1-tfeldma1/slim1-tchen2:        1-1-0
```

Semifinal with w=11, h=11, c=5, t=90:

```
jye1/dboshko1-tfeldma1:                1-1-0
dboshko1-tfeldma1/swallac3-nhoang1:    0-2-0
jye1/swallac3-nhoang1:                 0-2-0
(slim1-tchen2: betterEval requires c=4)
```

Semifinalists vs. Bryce:

```
jye1/bryce:                 0-2-0
dboshko1-tfeldma1/bryce:    1-1-0
swallac3-nhoang1/bryce:     1-1-0
slim1-tchen2/bryce:         0-2-0
```