Partial Observability

Objectives of this lecture:

- Introduction to POMDPs
- Solving POMDPs
- RL and POMDPs

Partially Observable MDPs (POMDPs)

Based on Cassandra, Kaelbling, & Littman, 12th AAAI, 1994

Start with an MDP <S, A, T, R>, where
S is finite state set
A is finite action set
T is the state transition function: T(s, a, s') is prob that next state is s', given doing a in state s
R is the reward function: R(s, a) is the immediate reward for doing a in state s

Add partial observability:
O, a finite set of possible observations
O, an observation function: O(a, s, o) is probability of observing o after taking action a in state s

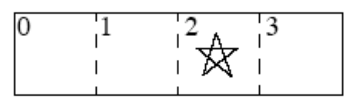
Complexity: finite horizon: PSPACE-complete.
infinite horizon: undecidable

A Little Example

Two actions: left, right; deterministic
If moves into a wall, stays in current state
If reaches the goal state (star), moves randomly to state 0, 1, or 3, and receives reward 1

Agent can only observe whether or not it is in the goal state

Belief State



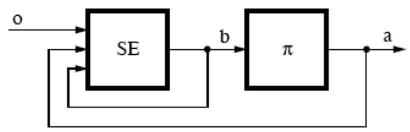
b: **belief state**: a discrete probability distribution over state set S
b(s) = prob agent is in state s

After goal: (1/3, 1/3, 0, 1/3)
After action right and not observing the goal: (0, 1/2, 0, 1/2)
After moving right again and still not observing the goal: (0, 0, 0, 1)

But in general, some actions in some situations can increase uncertainty, while others can decrease it. An optimal policy in general will sometimes take actions only to gain information.

The "Belief MDP"

Belief state estimator



$$
\begin{aligned}
\mathrm{SE}_{s'}(b,a,o) &= \Pr(s' \mid a,o,b) \\
&= \frac{\Pr(o \mid s',a,b)\Pr(s' \mid a,b)}{\Pr(o \mid a,b)} \\
&= \frac{O(a,s',o)\sum_{s\in\mathcal{S}} T(s,a,s')b(s)}{\Pr(o \mid a,b)}
\end{aligned}
$$

where $Pr(o \mid a,b)$ is a normalizing factor defined as

$$\Pr(o \mid a,b) = \sum_{s'\in\mathcal{S}} O(a,s',o) \sum_{s\in\mathcal{S}} T(s,a,s')b(s) \ .$$

Belief MDP cont.

Cassandra et al. say:

The key to finding truly optimal policies in the partially observable case is to cast the problem as a *completely observable* continuous-space MDP. The state set of this "belief MDP" is $\mathcal{B}$ and the action set is $\mathcal{A}$. Given a current belief state b and action a, there are only $|\mathcal{O}|$ possible successor belief states b', so the new state transition function, τ, can be defined as

$$\tau(b,a,b') = \sum_{\{o\in\mathcal{O}|\mathrm{SE}(b,a,o)=b'\}} \Pr(o \mid a,b) \ ,$$

where $\Pr(o \mid a,b)$ is defined above. If the new belief state, b', cannot be generated by the state estimator from b, a, and some observation, then the probability of that transition is 0. The reward function, ρ, is constructed from R by taking expectations according to the belief state; that is,

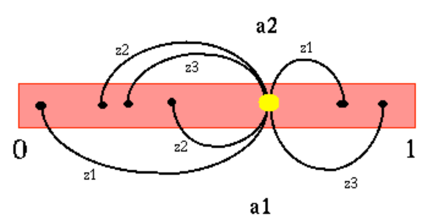
$$\rho(b,a) = \sum_{s\in\mathcal{S}} b(s)R(s,a) \ .$$

At first, this may seem strange; it appears the agent is rewarded simply for *believing* it is in good states. Because of the way the state estimation module is constructed, it is not possible for the agent to purposely delude itself into believing that it is in a good state when it is not.

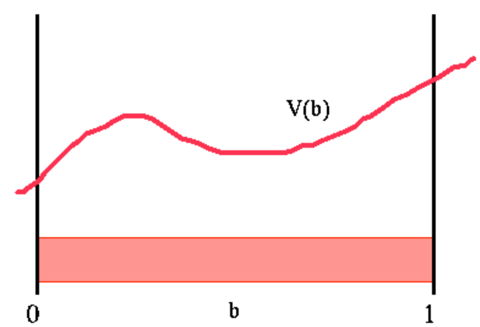
Value Iteration for the Belief MDP

from Tony Cassandra's "POMDPs for Dummies"
http://www.cs.brown.edu/research/ai/pomdp/tutorial

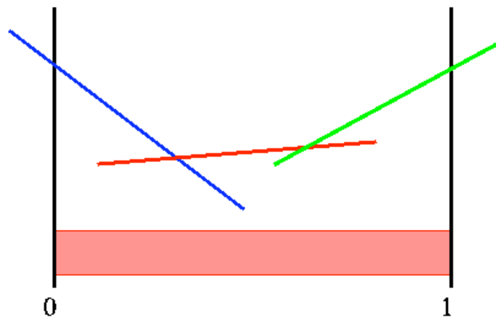
1D belief space for a 2 state POMDP



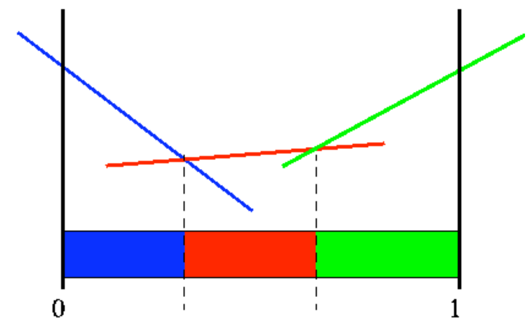
Value function over belief space



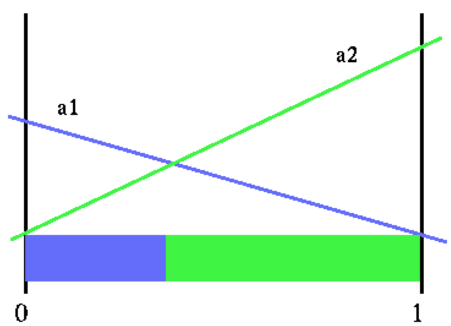
Sample PWLC value function



Sample PWLC function and its partition of belief space



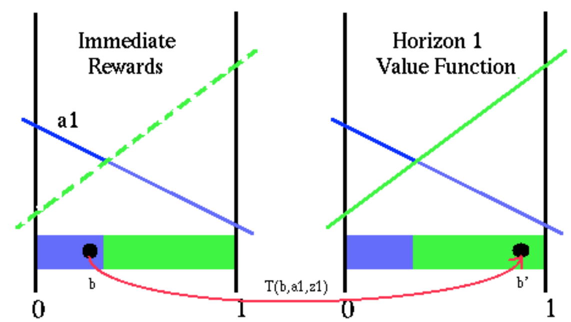
Immediate rewards for belief states



a1 has reward 1 in s1; 0 in s2
a2 has reward 0 in s1; 1.5 in s2

This is, in fact, the Horizon-1 value function

Value of a fixed action and observation

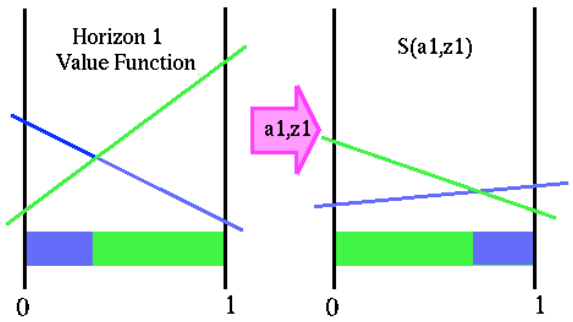


Summing these for the best action from b' gives the optimal horizon-2 value of taking a1 in b and observing z1

Note: here T is the earlier $\mathrm{SE}_{s'}(b,a,o)$

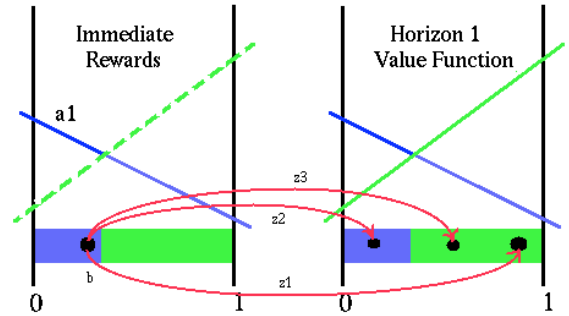
Transformed value function

Doing this for all belief sates:

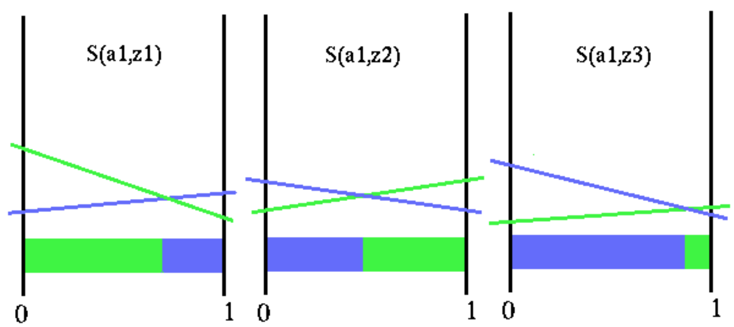


Immed reward + S(a1, z1) is the whole value function for action a1 and observation z1 [times P(z1 | a1, b)]

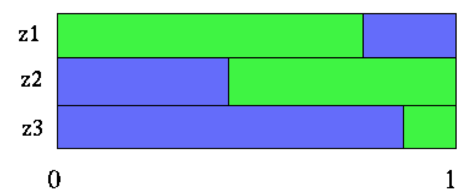
Do this for each observation given a1

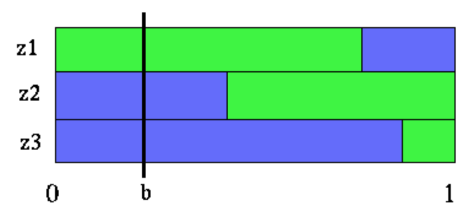


Transformed value function for all observations



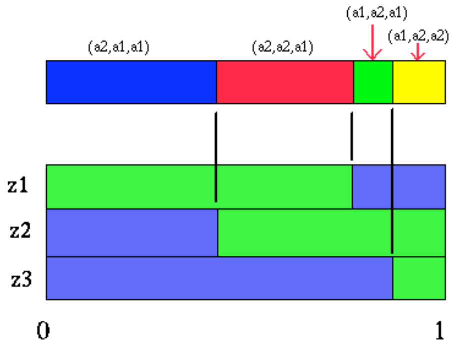
Partitions for all observations



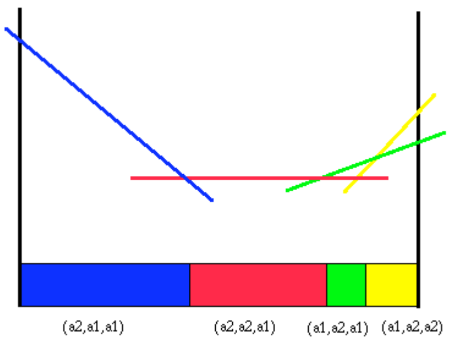


If we start at b and do a1, then next best action is:
a1 if we observe z2 or z3
a2 if we observe z1

Partition for action a1

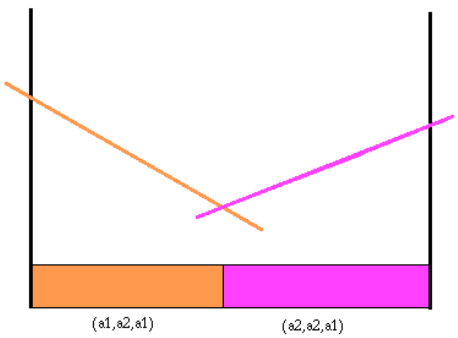


Value function and partition for action a1

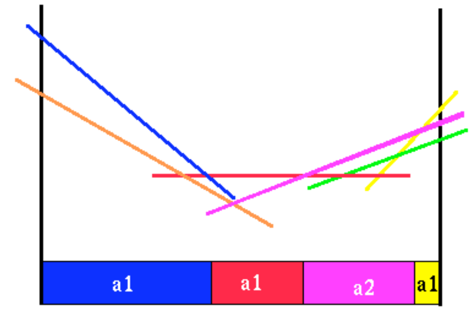


Produced by summing the appropriate S(a1,) lines

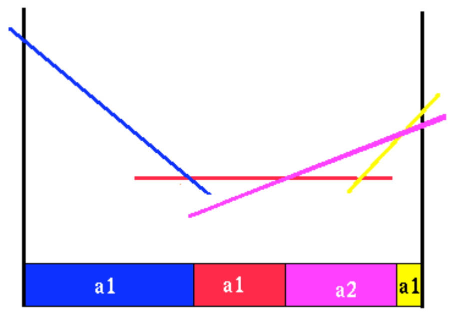
Value function and partition for action a2



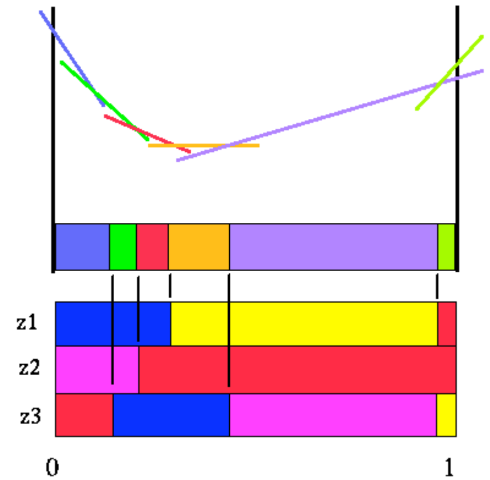
Combined a1 and a2 value functions



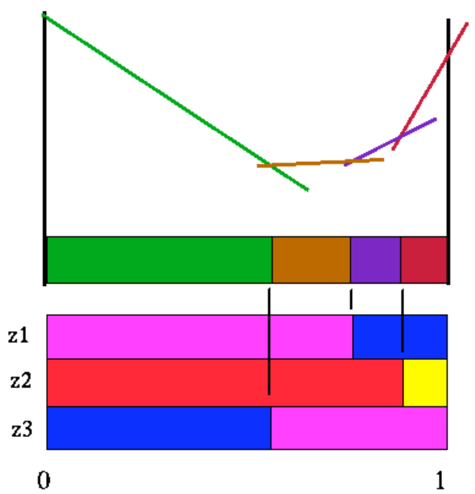
Value function for horizon 2



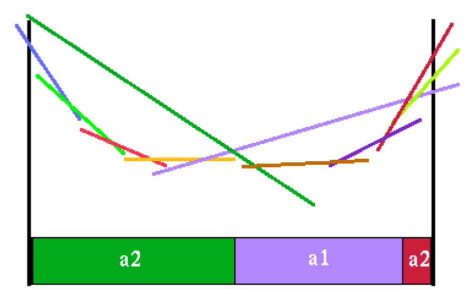
Value function for action a1 and horizon 3



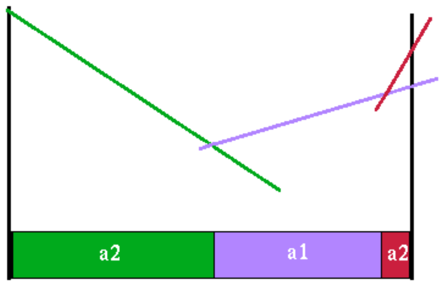
Value function for action a2 and horizon 3



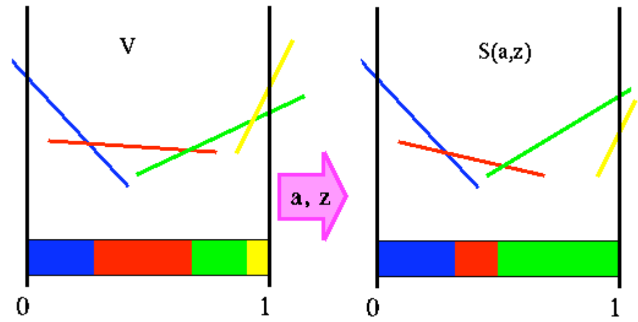
Value functions for both actions a2 and horizon 3



Value function for horizon 3

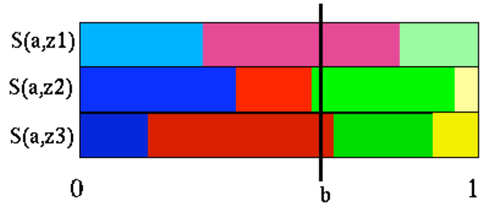


General Form of POMDP Solution

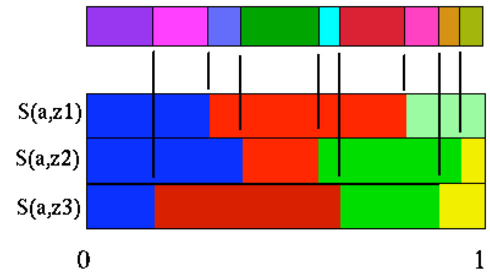


Transformed V for a and z

Adjacent belief partitions for transformed value function



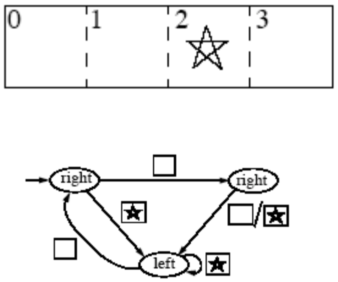
Making a new partition from S(a,z) partitions



How do you do this in general? Not so easy….

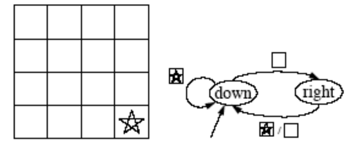
Policy Graphs

When all belief states in one partition are transformed into belief states in the same partition, given an optimal action and resulting observation, can form a finite state machine as policy.

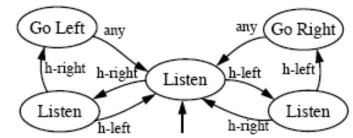


More policy graphs

Only goal state is distinguishable



Tiger Problem:
Two doors: tiger or big reward
You can choose to listen (for a small cost)
If tiger is on left, you will hear it on left with prob 0.85, and on right with prob 0.15, and symmetrically if tiger is on right
Iterated: restarts with tiger and reward randomly repositioned



RL for POMDPs

- Memoryless policies: treat observations as if they were Markov states
  - Use non-bootstrapping algorithm to estimate *Q(o, a)* for observations *o*; do policy improvement
  - Policies can be bad
  - Stochastic policies can be better
- Q_{MDP} method:
  - Ignore the observation model and find optimal Q-values for the underlying MDP
  - Extend to belief states like this: $Q_a(b) = \sum_s b(s)\, Q_{MDP}(s,a)$
  - Assume all uncertainty disappears in one step: cannot produce policies that act to gain information
  - But can work surprisingly well in many cases

RL for POMDPs

- Replicated Q-learning
  - Use a single vector, q_a, to approx Q-function for each action: $Q_a(b) = q_a \cdot b$
  - At each step, for every state s:

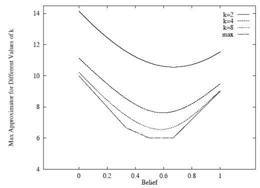
$$\Delta q_a(s) = \alpha\, b(s)\left(r + \gamma \max_{a'} Q_{a'}(b') - q_a(s)\right)$$

  - Reduces to normal Q-learning if belief state collapses to deterministic case
  - Certainly suboptimal, but sometimes works well

RL for POMDPs

- Smooth Partially Observable Value Approximation (SPOVA) Parr and Russell

$$V(b) = \sqrt[k]{\sum_{\gamma \in \Gamma} (b \cdot \gamma)^k}.$$



For each belief state b

$E \leftarrow V(b) - (R(b) + \beta \max_{a \in A} \sum_{b' \in next(b,a)} P(b'|b,a)V(b'))$

For i from 1 to $|\Gamma|$

For j from 1 to n

$\gamma_{ij} \leftarrow \gamma_{ij} + \alpha E b_j (\gamma_i \cdot b)^{k-1} / V(b)^{k-1}$

SPOVA

$a \leftarrow$ best action according to V

$b' \leftarrow$ simulated result of taking a in b.

$E_{RL}(b) \leftarrow V(b) - (R(b) + V(b'))$

For i from 1 to $|\Gamma|$

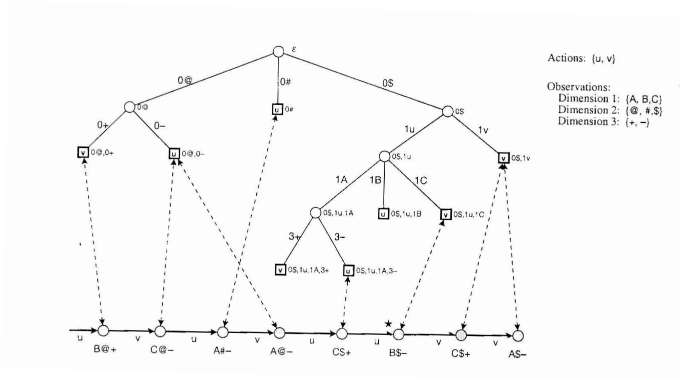
For j from 1 to n

$\gamma_{ij} \leftarrow \gamma_{ij} + \alpha E_{RL}(b) b_j (\gamma_i \cdot b)^{k-1} / V(b)^{k-1}$

SPOVA-RL

RL for POMDPs

- McCallum's U-Tree algorithm, 1996



RL for POMDPs

- Linear Q-Learning
  - Almost the same as replicated Q-learning:

$$\Delta q_a(s) = \alpha\, b(s)\left(r + \gamma \max_{a'} Q_{a'}(b') - q_a(s)\right) \quad \text{replicated}$$

$$\Delta q_a(s) = \alpha\, b(s)\left(r + \gamma \max_{a'} Q_{a'}(b') - q_a \cdot b\right) \quad \text{linear}$$