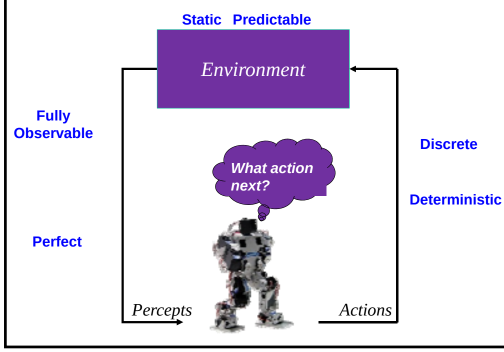
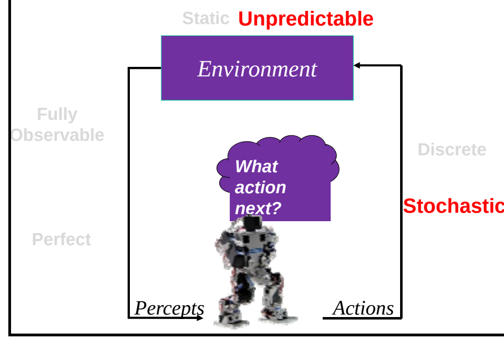
CSE-473 Artificial Intelligence

Partially-Observable MDPS (POMDPs)

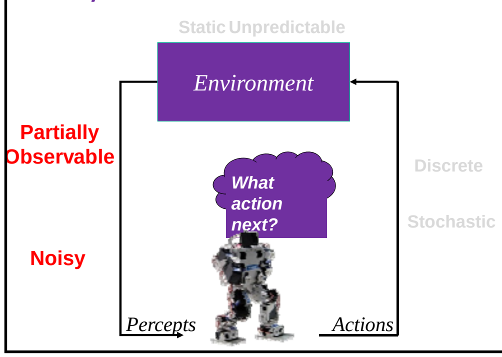
Classical Planning



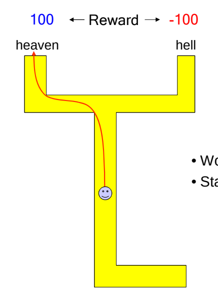
Stochastic Planning (MDPs, Reinforcement Learning)



Partially-Observable MDPs



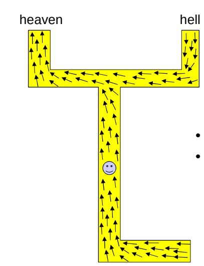
Classical Planning



- Sequential Plan
- World deterministic
- State observable

5

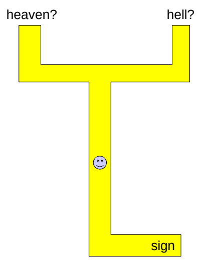
MDP-Style Planning



- Policy
- World stochastic
- State observable

6

Stochastic, *Partially* Observable



Markov Decision Process (MDP)

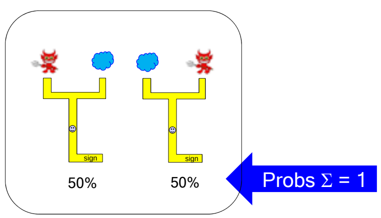
- **S**: set of states
- **A**: set of actions
- **Pr**(s'|s,a): transition model
- **R**(s,a,s'): reward model
- γ: discount factor
- s_0: start state

Partially-Observable MDP

- **S**: set of states
- **A**: set of actions
- **Pr**(s'|s,a): transition model
- **R**(s,a,s'): reward model
- γ: discount factor
- s_0: start state
- **E** set of possible evidence (observations)
- **Pr**(e|s)

Belief State

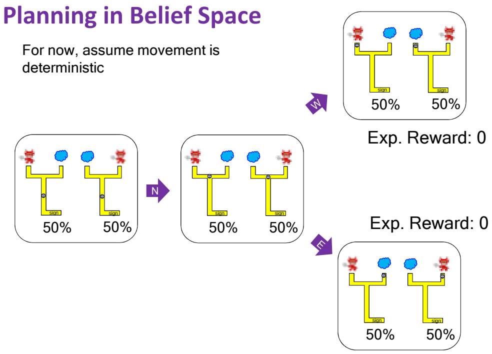
- State of agent's mind
- Not just of world



Note: POMDP

Planning in Belief Space

For now, assume movement is deterministic



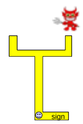
Partially-Observable MDP

- **S**: set of states
- **A**: set of actions
- **Pr**(s'|s,a): transition model
- **R**(s,a,s'): reward model
- γ: discount factor
- s_0: start state
- **E** set of possible evidence (observations)
- **Pr**(e|s)

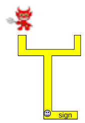
Evidence Model

- **S** = {s_{wb}, s_{eb}, s_{wm}, s_{em} s_{wul}, s_{eul} s_{wur}, s_{eur}}
- **E** = {heat}
- **Pr(e|s):**
  Pr(heat | s_{eb}) = 1.0
  Pr(heat | s_{wb}) = 0.2
  Pr(heat | s_{other}) = 0.0

s_{eb}

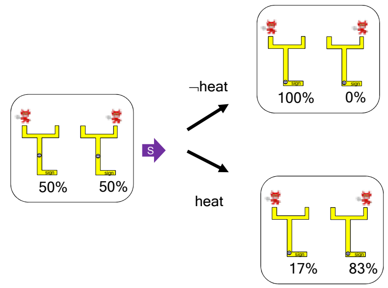


s_{wb}



Planning in Belief Space

Pr(heat | s_{eb}) = 1.0
Pr(heat | s_{wb}) = 0.2



Objective of a Fully Observable MDP

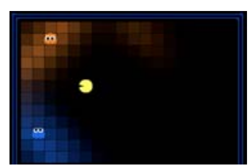
- Find a policy π: **S → A**
- which maximizes expected discounted reward
  - given an infinite horizon
  - assuming full observability

Objective of a POMDP

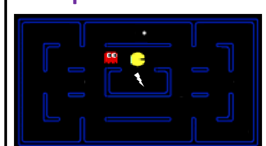
- Find a policy
  π: BeliefStates(**S**) → **A**
  A belief state is a ***probability distribution*** over states
- which maximizes expected discounted reward
  - given an infinite horizon
  - assuming partial & noisy observability

Planning in HW 4

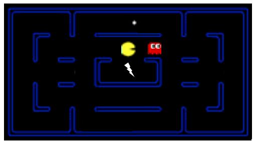
- Map Estimate
- Now “know” state
- Solve MDP



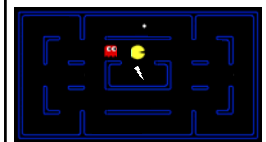
Best plan to eat final food?

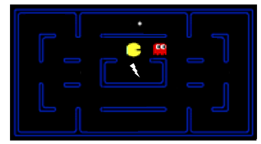


Best plan to eat final food?



Problem with Planning from MAP Estimate





49% 51%

10% 90%

- Best action for belief state over k worlds may not be the best action in any one of those worlds

POMDPs

- In POMDPs we apply the very same idea as in MDPs.
- Since the state is not observable, the agent has to make its decisions based on the belief state which is a posterior distribution over states.
- Let b be the belief of the agent about the state under consideration.
- POMDPs compute a value function over belief space:

$$V_T(b) \;=\; \max_u \left[r(b,u) + \gamma \int V_{T-1}(b')\, p(b' \mid u, b)\; db' \right]$$

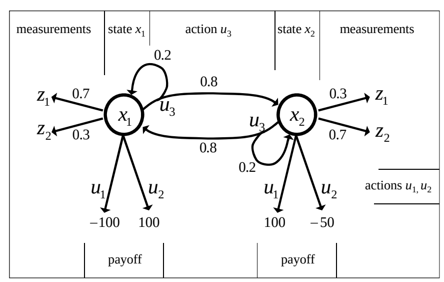
21

Problems

- Each belief is a probability distribution, thus, each value in a **POMDP is a function of an entire probability distribution**.
- **This is problematic, since probability distributions are continuous**.
- How many belief states are there?
- For **finite worlds** with finite state, action, and measurement spaces and finite horizons, however, we can **effectively represent the value functions by piecewise linear functions**.

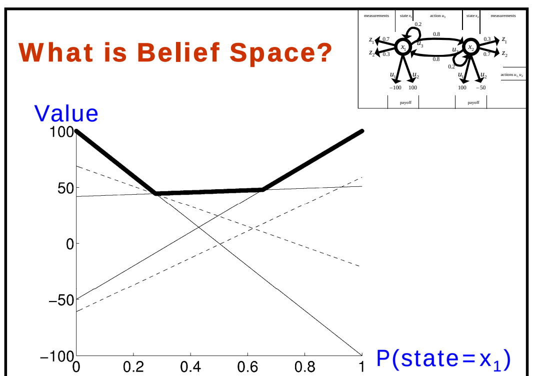
22

An Illustrative Example



23

What is Belief Space?



24

The Parameters of the Example

- The actions u_1 and u_2 are terminal actions.
- The action u_3 is a sensing action that potentially leads to a state transition.
- The horizon is finite and γ=1.

$$r(x_1,u_1) = -100 \qquad r(x_2,u_1) = +100$$
$$r(x_1,u_2) = +100 \qquad r(x_2,u_2) = -50$$
$$r(x_1,u_3) = -1 \qquad r(x_2,u_3) = -1$$

$$p(x'_1|x_1,u_3) = 0.2 \qquad p(x'_2|x_1,u_3) = 0.8$$
$$p(x'_1|x_2,u_3) = 0.8 \qquad p(z'_2|x_2,u_3) = 0.2$$

$$p(z_1|x_1) = 0.7 \qquad p(z_2|x_1) = 0.3$$
$$p(z_1|x_2) = 0.3 \qquad p(z_2|x_2) = 0.7$$

25

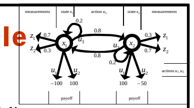
Payoff in POMDPs

- In MDPs, the payoff (or return) depended on the state of the system.
- In POMDPs, however, the true state is not exactly known.
- Therefore, we compute the **expected payoff** by **integrating over all states**:

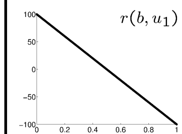
$$\begin{aligned} r(b,u) &= E_x[r(x,u)] \\ &= \int r(x,u)p(x)\,dx \\ &= p_1\, r(x_1,u) + p_2\, r(x_2,u) \end{aligned}$$

26

Payoffs in Our Example



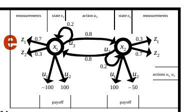
- If we are totally certain that we are in state x_1 and execute action u_1, we receive a reward of -100
- If, on the other hand, we definitely know that we are in x_2 and execute u_1, the reward is +100.
- In between it is the linear combination of the extreme values weighted by the probabilities



$$\begin{aligned} r(b,u_1) &= -100\, p_1 + 100\, p_2 \\ &= -100\, p_1 + 100\,(1-p_1) \\ &= 100 - 200\, p_1 \end{aligned}$$

27

Payoffs in Our Example



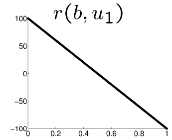
- If we are totally certain that we are in state x_1 and execute action u_1, we receive a reward of -100
- If, on the other hand, we definitely know that we are in x_2 and execute u_1, the reward is +100.
- In between it is the linear combination of the extreme values weighted by the probabilities

$$\begin{aligned} r(b,u_1) &= -100\, p_1 + 100\, p_2 \\ &= -100\, p_1 + 100\,(1-p_1) \\ &= 100 - 200\, p_1 \\ r(b,u_2) &= 100\, p_1 - 50\,(1-p_1) \\ &= 150\, p_1 - 50 \\ r(b,u_3) &= -1 \end{aligned}$$

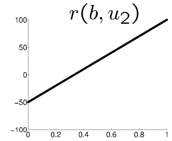
28

Payoffs in Our Example (2)

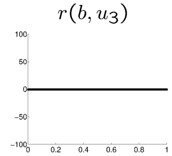
$r(b,u_1)$



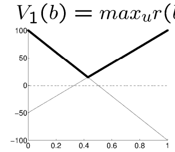
$r(b,u_2)$



$r(b,u_3)$



$V_1(b) = max_u r(b,u)$



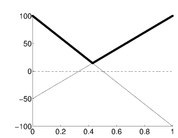
29

The Resulting Policy for T=1

- Given a finite POMDP with time horizon = 1
- Use $V_1(b)$ to determine the optimal policy.

$$\pi_1(b) = \begin{cases} u_1 & \text{if } p_1 \le \frac{3}{7} \\ u_2 & \text{if } p_1 > \frac{3}{7} \end{cases}$$

- Corresponding value:



30

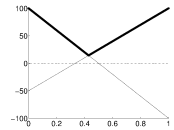
Piecewise Linearity, Convexity

- The resulting value function $V_1(b)$ is the maximum of the three functions at each point

$$V_1(b) = \max_u r(b,u) = \max \left\{ \begin{array}{ll} -100\ p_1 & +100\ (1-p_1) \\ 100\ p_1 & -50\ (1-p_1) \\ -1 & \end{array} \right\}$$

- It is piecewise linear and convex.

Pruning

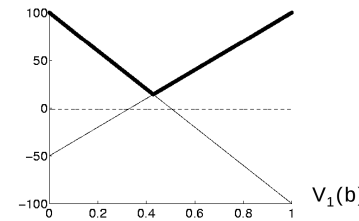


- With $V_1(b)$, note that only the first two components contribute.
- The third component can be safely pruned

$$V_1(b) = \max \left\{ \begin{array}{ll} -100\ p_1 & +100\ (1-p_1) \\ 100\ p_1 & -50\ (1-p_1) \end{array} \right\}$$

Increasing the Time Horizon

- Assume the robot can make an observation before deciding on an action.



Increasing the Time Horizon

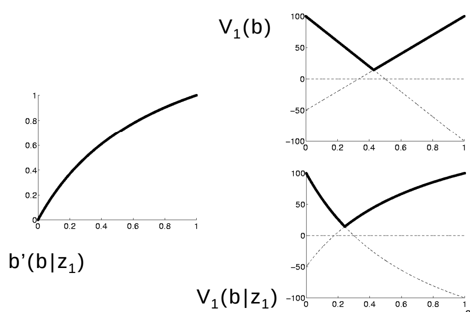
- What if the robot can observe before acting?
- Suppose it perceives z_1: $p(z_1 \mid x_1)=0.7$ and $p(z_1 \mid x_2)=0.3$.
- Given the obs z_1 we update the belief using Bayes rule.

$$p'_1 = \frac{0.7\, p_1}{p(z_1)} \quad \text{where} \quad p(z_1) = 0.7\, p_1 + 0.3(1-p_1) = 0.4\, p_1 + 0.3$$

- Now, $V_1(b \mid z_1)$ is given by

$$V_1(b \mid z_1) = \max \left\{ \begin{array}{ll} -100 \cdot \frac{0.7\, p_1}{p(z_1)} & +100 \cdot \frac{0.3\,(1-p_1)}{p(z_1)} \\ 100 \cdot \frac{0.7\, p_1}{p(z_1)} & -50 \cdot \frac{0.3\,(1-p_1)}{p(z_1)} \end{array} \right\} = \frac{1}{p(z_1)} \max \left\{ \begin{array}{ll} -70\ p_1 & +30\ (1-p_1) \\ 70\ p_1 & -15\ (1-p_1) \end{array} \right\}$$

Value Function



Expected Value after Measuring

- But, we do not know ***in advance*** what the next measurement will be,
- So we must compute the expected belief

$$\bar{V}_1(b) = E_z[V_1(b \mid z)] = \sum_{i=1}^{2} p(z_i) V_1(b \mid z_i)$$

$$= \sum_{i=1}^{2} p(z_i) V_1\left(\frac{p(z_i \mid x_1)\, p_1}{p(z_i)}\right)$$

$$= \sum_{i=1}^{2} V_1\left(p(z_i \mid x_1)\, p_1\right)$$

Expected Value after Measuring

- But, we do not know ***in advance*** what the next measurement will be,
- So we must compute the expected belief

$$\begin{aligned}\bar{V}_1(b) &= E_z[V_1(b \mid z)] \\ &= \sum_{i=1}^{2} p(z_i)\, V_1(b \mid z_i) \\ &= \max\left\{\begin{array}{ll} -70\, p_1 & +30\,(1-p_1) \\ 70\, p_1 & -15\,(1-p_1) \end{array}\right\} \\ &\quad + \max\left\{\begin{array}{ll} -30\, p_1 & +70\,(1-p_1) \\ 30\, p_1 & -35\,(1-p_1) \end{array}\right\}\end{aligned}$$

37

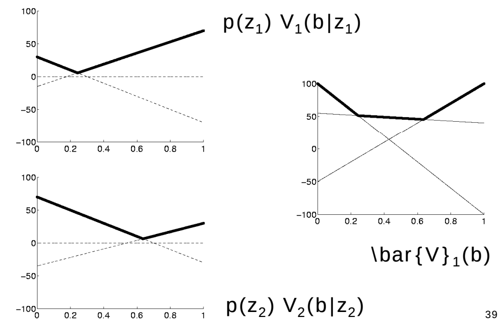
Resulting Value Function

- The four possible combinations yield the following function which then can be simplified and pruned.

$$\begin{aligned}\bar{V}_1(b) &= \max\left\{\begin{array}{llll} -70\, p_1 & +30\,(1-p_1) & -30\, p_1 & +70\,(1-p_1) \\ -70\, p_1 & +30\,(1-p_1) & +30\, p_1 & -35\,(1-p_1) \\ +70\, p_1 & -15\,(1-p_1) & -30\, p_1 & +70\,(1-p_1) \\ +70\, p_1 & -15\,(1-p_1) & +30\, p_1 & -35\,(1-p_1) \end{array}\right\} \\ &= \max\left\{\begin{array}{ll} -100\, p_1 & +100\,(1-p_1) \\ +40\, p_1 & +55\,(1-p_1) \\ +100\, p_1 & -50\,(1-p_1) \end{array}\right\}\end{aligned}$$

38

Value Function

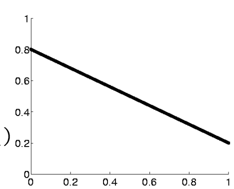


39

State Transitions (Prediction)

- When the agent selects u_3 its state may change.
- When computing the value function, we have to take these potential state changes into account.

$$\begin{aligned}p_1' &= E_x[p(x_1 \mid x, u_3)] \\ &= \sum_{i=1}^{2} p(x_1 \mid x_i, u_3) p_i \\ &= 0.2 p_1 + 0.8(1-p_1) \\ &= 0.8 - 0.6 p_1\end{aligned}$$



40

Resulting Value Function after executing u_3

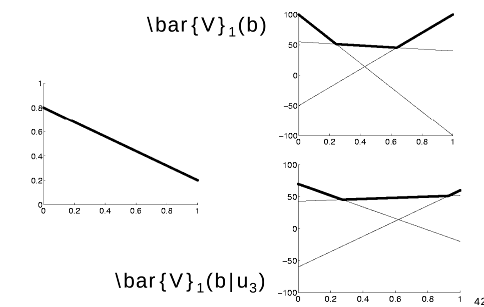
Taking the state transitions into account, we finally obtain.

$$\begin{aligned}\bar{V}_1(b) &= \max\left\{\begin{array}{llll} -70\, p_1 & +30\,(1-p_1) & -30\, p_1 & +70\,(1-p_1) \\ -70\, p_1 & +30\,(1-p_1) & +30\, p_1 & -35\,(1-p_1) \\ +70\, p_1 & -15\,(1-p_1) & -30\, p_1 & +70\,(1-p_1) \\ +70\, p_1 & -15\,(1-p_1) & +30\, p_1 & -35\,(1-p_1) \end{array}\right\} \\ &= \max\left\{\begin{array}{ll} -100\, p_1 & +100\,(1-p_1) \\ +40\, p_1 & +55\,(1-p_1) \\ +100\, p_1 & -50\,(1-p_1) \end{array}\right\}\end{aligned}$$

$$\bar{V}_1(b \mid u_3) = \max\left\{\begin{array}{ll} 60\, p_1 & -60\,(1-p_1) \\ 52\, p_1 & +43\,(1-p_1) \\ -20\, p_1 & +70\,(1-p_1) \end{array}\right\}$$

41

Value Function after executing u_3



42

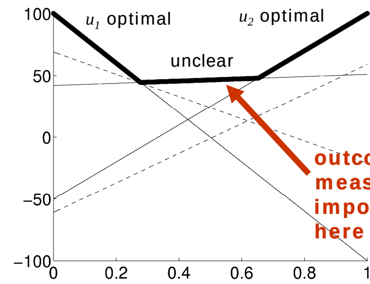
Value Function for T=2

- Taking into account that the agent can either directly perform u_1 or u_2 or first u_3 and then u_1 or u_2, we obtain (after pruning)

$$\bar{V}_2(b) = \max \left\{ \begin{array}{rr} -100\, p_1 & +100\,(1-p_1) \\ 100\, p_1 & -50\,(1-p_1) \\ 51\, p_1 & +42\,(1-p_1) \end{array} \right\}$$

43

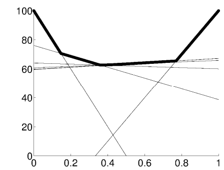
Graphical Representation of $V_2(b)$

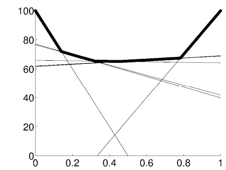


44

Deep Horizons

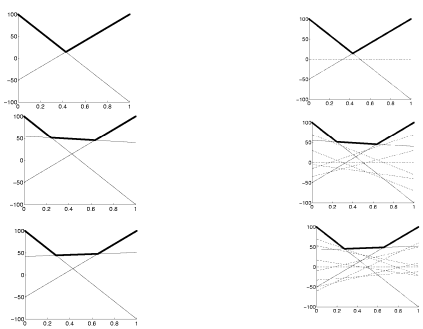
- We have now completed a full backup in belief space.
- This process can be applied recursively.
- The value functions for T=10 and T=20 are





45

Deep Horizons and Pruning



46

Why Pruning is Essential

- Each **update introduces additional linear components** to V.
- Each **measurement squares the number of linear components**.
- Thus, an unpruned value function for T=20 includes more than $10^{547,864}$ linear functions.
- At T=30 we have $10^{561,012,337}$ linear functions.
- The pruned value functions at T=20, in comparison, contains only 12 linear components.
- The combinatorial explosion of linear components in the value function are the major reason why **exact solution of POMDPs is usually impractical**

47

POMDP Approximations

- Point-based value iteration
- QMDPs
- AMDPs

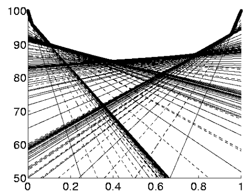
49

Point-based Value Iteration

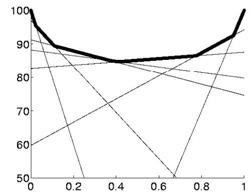
- Maintains a set of example beliefs
- Only considers constraints that maximize value function for at least one of the examples

Point-based Value Iteration

Value functions for T=30



Exact value function



PBVI

QMDPs

- QMDPs only consider state uncertainty in the first step
- After that, the world becomes fully observable.

POMDP Summary

- POMDPs compute the optimal action in partially observable, stochastic domains.
- For finite horizon problems, the resulting value functions are piecewise linear and convex.
- In each iteration the number of linear constraints grows exponentially.
- Until recently, POMDPs only applied to very small state spaces with small numbers of possible observations and actions.
  - But with PBVI, |S| = millions