Partially-Observable MDPs
Decision Theoretic Agents

- Introduction to Probability [Ch13]
- Belief networks [Ch14]
- Dynamic Belief Networks [Ch15]
- Single Decision [Ch16]

- Sequential Decisions [Ch17]
  - MDPs [Ch17.1 - 17.3]
    - (Value Iteration, Policy Iteration, TD(\(\lambda\)))
  - POMDPs [Ch17.4 - 17.5]
    - Dynamic Decision Networks
  - Game Theory [Ch17.6 - 17.7]
Partially Accessible Environment

- In inaccessible environment
  percept NOT enough to determine state
  Partially Observable
  Markov Decision Problem “POMDP”

⇒ Need to base decision on
  DISTRIBUTION over possible states,
  based all previous percepts, . . . (E)

Eg: Given only distance to walls in 4 directions,
  “[ 2, 1 ] ≡ [ 2, 3 ]”
  but DIFFERENT actions for each!
If  \( P( \text{Loc}[2,1] | E) = 0.8 \),  \( P( \text{Loc}[2,3] | E) = 0.2 \)
then utility of action \( a \) is
\[ 0.8 \times U( a | \text{Loc}[2,1]) + 0.2 \times U( a | \text{Loc}[2,3]) \]
Dealing with POMDPs

- Why not view “percept == state”... and just apply MDP alg to “percept”??

1. Markov property does NOT hold for percepts (percept ≠ states)
   - MDP means
     - *next state depends only on current state*
   - But in POMDP:
     - next percept does NOT depend only on current percept

2. May need to take action to *reduce uncertainty*
   - ... not needed in MDP, as always KNOW state
     - ⇒ utility should include ValueOfInformation...
Extreme Case: Senseless Agent

- What if **NO** observations?
- Perhaps
  - act to reduce uncertainty
  - then go to goal
    - (a) Initially: could be ANYWHERE
    - (b) After “Left” 5 times
    - (c) ... then “Up” 5 times
    - (d) ... then “Right” 5 times
- Prob of reaching [4,3]: 77.5%
  - but slow: Utility $\approx 0.08$
“Senseless” Multi-step Agents

- Want **sequence of actions** \([a_1, \ldots, a_n]\)

that maximizes the expected utility:

\[
\arg\max_{[a_1, \ldots, a_n]} \sum_{[s_0, \ldots, s_n]} P(s_0, \ldots, s_n | a_1, \ldots, a_n) \times U(s_0, a_1, \ldots, a_n, s_n)
\]

- If deterministic, use problem solving techniques to “solve”
  - (finding optimal sequence)

- Stochastic \(\Rightarrow\) don’t know state. . .

  but deal w/ DISTRIBUTION OVER STATES
Unobservable Environments

- **View Action-Sequence as Big Action**
  
  For each possible actions-sequence \([a_1, \ldots, a_n]\)
  
  compute \(P(S_0, S_1, \ldots, S_n | a_1, \ldots, a_n)\)
  
  compute \(U([s_0, a_1, \ldots, a_n, s_n])\)
  
  compute \(score = \sum_{[s_0, \ldots, s_n]} P(s_0, \ldots, s_n | a_1, \ldots, a_n) \cdot U([s_0, a_1, \ldots, a_n, s_n])\)
  
  Return action-sequence that gave maximum \(score\)

- **As Markovian:**
  
  \[ P( S_0, S_1, \ldots, S_n | a_1, \ldots, a_n ) = \]
  \[ P( S_0 ) \cdot P( S_1 | S_0, a_1 ) \cdot P( S_2 | S_1, a_2 ) \cdot \cdots \cdot P( S_n | S_{n-1}, a_n ) \]

  \[ U( [s_0, a_1, \ldots, a_n, s_n] ) = \sum_t R( s_t ) \]

  \[ \Rightarrow \] For each action sequence, requires searching over all possible sequences of resulting states.

  - If \( P( S_{t+1} | S_t, A_{t+1} ) \) deterministic, can be solved using search...
Next action must depend on Complete Sequence of Percepts, \( \mathbf{O} \)

- (That is all available to agent!)

Compress \( \mathbf{O} \) into “distribution over states”

- \( \mathbf{p} = [p_1, \ldots, p_n] \) where \( p_i = P(\text{state} = i \mid \mathbf{O}) \)

Given new percept \( \mathbf{O}_t \),

\( \mathbf{p}' = [P(\text{state} = i \mid \mathbf{O}, \mathbf{O}_t)] \)
POMDPs

- Partially Observable Markov Decision Problem
  - $M^a_{s,s'} \equiv P( s' \mid s, a )$: transition
  - $R(s)$: reward function
  - $O(s, o) \equiv P( o \mid s )$: observation model
    - [If senseless: $O(s, \{\}) = 1.0$]
  - Belief state $b(.) \equiv$ distribution over states
    - $b(s) \equiv P( s \mid ... )$ is prob $b$ assigns to $s$
    - Eg: $b_{\text{init}} = \langle 1/9, 1/9, ... 1/9, 0,0 \rangle$
  - Given $b(.)$, after action $a$, observation $o$
    - $b'(s') = O(s', o) \sum_s P( s \mid a, s' ) b(s)$
    - $b' = \text{Forward}( b, a, o )$

Filtering!

- Optimal action depends only on current belief state!
  - . . . not on actual state
What to do, in POMDP?

Policy $\pi$ maps BELIEF STATE $b$ to ACTION $a$

$\pi(b) = a \quad \pi: [0, 1]^n \mapsto \{ \text{North, East, South, West} \}$

Given optimal policy $\pi^*$

1. Given $b_i$ compute/execute action $a_i = \pi(b_i)$
2. Receive observation $o_i$
3. Compute $b_{i+1} = \text{Forward}(b_i, a_i, o_i)$

With MDPs, can just "reach" new state ... no observations...
With POMDPs, need to know observation $o_i$ to determine $b'$

Some POMDP actions may be

- to reduce uncertainty
- to gather information

How to compute optimal $\pi^*$ ?
... perhaps make POMDP look like MDP?
Transform POMDP into MDP?

- Every MDP needs
  - Transition $M$: State $\rightarrow$ Action $\rightarrow$ Distribution over State
  - Reward $R$: State $\rightarrow$ $\mathbb{R}$

$\Rightarrow$ Given “belief state” $b$, need

- $\rho(b) =$ (expected) reward for being in $b$
  $$= \sum_s b(s) R(s)$$

- $\mu(b, a, b') = P( b' | b, a )$
  $$\ldots \text{prob of reaching } b' \text{ if take action } a \text{ in } b. \ldots$$

Depends on observation $o$:

- $P( b' | a, b ) = \sum_o P( b' | o, a, b ) P( o | a, b )$
  $$= \sum_o \delta[ b' = \text{Forward}(b, a, o) ] P( o | a, b )$$

- where $\delta[ b' = \text{Forward}(b, a, o) ] = 1$ \text{ iff } b' = \text{Forward}(b, a, o)$

- Need DISTRIBUTION over observations . . .
Distribution over Observations

\[ P(o|a,b) = \sum_{s'} P(o,a,s',b) P(s'|a,b) \]
\[ = \sum_{s'} O(s',o) P(s'|a,b) \]
\[ = \sum_{s'} O(s',o) \sum_{s} P(s'|a,s) b(s) \]

So...

\[ \mu^a_{b,b'} = P(b'|a,b) \]
\[ = \sum_{o} P(b'|o,a,b) P(o|a,b) \]
\[ = \sum_{o} \sum_{s'} \sum_{s} O(s',o) \sum_{s} P(s'|a,s) b(s) \]
POMDP $\Rightarrow$? MDP ??

- $\mu_{b,b'}^a = P(b' | b, a)$
- $\rho(b) = \text{(expected) reward}$
- ... define OBSERVABLE MDP!
  (Agent can always observe its beliefs!)
- Optimal policy for this MDP $\pi^*(b)$
  is optimal for POMDP

  **Solving POMDP on physical state space**

  $\equiv$

  **solving MDP on corresponding BELIEF STATE SPACE!**

- But. . . this MDP has continuous
  (and usually HIGH-Dimension)
  state space!
- Fortunately . . .
Transform POMDP into MDP

- Fortunately, ∃ versions of
  - value iteration
  - policy iteration

that apply to such continuous-space MDPs
(Represent $\pi(b)$ as set of REGIONS of belief space
  each with specific optimal action)
$U \equiv$ LINEAR FUNCTION of $b$ w/in each region
Each iteration refines boundaries of regions . . .

- Solution:
  
  \[
  \text{[Left, Up, Up, Right, Up, Up, Right, Up, Up, ...]}
  \]
  
  (Left ONCE to ensure NOT at [4,1],
  then go Right and Up until reaching [4, 3].)

  Succeeds 86.6%, quickly. . .

  Utility = 0.38

- In general: finding optimal policies is PSPACE-Hard!
Solving POMDP, in General

**function** DECISION-THEORETIC-AGENT( percept) **returns** action

- calculate updated probabilities for current state based on available evidence including current percept and previous action
- calculate outcome probabilities for actions given action descriptions and probabilities of current states
- select *action* with highest expected utility given probabilities of outcomes and utility information

**return** action

- To determine current state $S_t$:  
  - Deterministic: previous action $a_{t-1}$ from $S_{t-1}$ determines $S_t$  
  - Accessible: current percepts identify $S_t$  
  - Partially accessible: use BOTH action and percepts

- Computing outcome probabilities:  
  . . . as above

- Computing *expected utilities*:  
  At time $t$, need to think about making decision $D_{t+i}$  
  At that time $t+i$, agent will THEN have percepts $E_{t+1}, \ldots, E_{t+i}$  
  But not known now (at time $t$). . .
Challenges

- To decide about \( A_t \) (action at time \( t \)), need distribution of current state based on:
  - all evidence (\( E_i \) is evidence at time \( i \))
  - all actions (\( A_i \) is action at time \( i \))

\[
Bel(S_t) \equiv P(S_t | E_1, \ldots, E_t, A_1, \ldots, A_{t-1})
\]

\[\Rightarrow \text{very hard to compute, in general}\]

- But. . . some simplifications:
  - \( P(S_t | S_1, \ldots, S_{t-1}, A_1, \ldots, A_{t-1}) = P(S_t | S_{t-1}, A_{t-1}) \)
    Markov
  - \( P(E_t | S_1, \ldots, S_t, E_1, \ldots, E_t, A_1, \ldots, A_{t-1}) = P(E_t | S_t) \)
    Evidence depends only on current world
  - \( P(A_{t-1} | A_1, \ldots, A_{t-2}, E_1, \ldots, E_{t-1}) = P(A_{t-1} | E_1, \ldots, E_{t-1}) \)
    Agent acts based only input. . . and knows what it did

RECURSIVE form of \( Bel() \) updated with each evidence:

- **Prediction Phase:**
  Predict distribution over state, before evidence

\[
Bel(S_t) = \sum_{s_{t-1}} P(S_t | S_{t-1} = s_{t-1}, A_{t-1}) \, Bel(S_{t-1} = s_{t-1})
\]

- **Estimation Phase:** ... Incorporate \( E_t \)

\[
Bel(S_t) = \alpha \, P(E_t | S_t) \, Bel(S_t)
\]
function DECISION-THEORETIC-AGENT(E_t) returns an action
inputs: E_t, the percept at time t
static: BN, a belief network with nodes X
      Bel(X), a vector of probabilities, updated over time

\[
\begin{aligned}
\tilde{Bel}(X_t) &\leftarrow \sum_{X_{t-1}} P(X_t \mid X_{t-1}=x_{t-1}, A_{t-1}) \ Bel(X_{t-1}=x_{t-1}) \\
Bel(X_t) &\leftarrow \alpha \ P(E_t \mid P X_t) \tilde{Bel}(X_t) \\
action &\leftarrow \arg \max_{A_t} \ \sum_{X_t} \left[ \tilde{Bel}(X_t=x_t) \sum_{X_{t+1}} P(X_{t+1}=x_{t+1} \mid X_t=x_t, A_t) \ U(x_{t+1}) \right] \\
\end{aligned}
\]
return action

- Dependencies are reasonable:
  - action mode: \( P( S_t \mid S_{t-1}, A_{t-1} ) \)
  - sensor model: \( P( E_t \mid S_t ) \)
Partially Observable MDPs
Dynamic Decision Networks
Approximate Method for Solving POMDP's

Two Key Ideas:
- Compute optimal value function \( U(S) \) assuming complete observability
  (Whatever will be needed later, will be available)
- Maintain \( \text{Bel}(S_t) = P(S_t | E_t, A_t, S_{t-1}, ..., S_0, E_0) \)

At each time \( t \):
- Observe current percept \( E_t \)
- Update \( \text{Bel}(S_t) \)
- Choose next \( k \) optimal actions \([a_{t+1}, ..., a_{t+k}]\)
to maximize
\[
\sum_{S_{t+1}, ..., S_{t+k}} \sum_{E_{t+1}, ..., E_{t+k}} P(S_{t+1} | S_t, a_{t+1}) \cdot P(E_{t+1} | S_{t+1}) \cdot \cdots \cdot P(S_{t+k} | S_{t+k-1}, a_{t+k}) \\
[\sum_{i=1}^{k} R(S_{t+i} | S_{t+i-1}, a_{t+i}) + U(S_{t+k})]
\]
- Perform action \( a_{t+1} \)
Look-ahead Search

ExpectiMiniMax

\[ D_t \text{ in } P(X_t | E_{1:t}) \]
\[ E_{t+1} \]
\[ D_{t+1} \text{ in } P(X_{t+1} | E_{1:t+1}) \]
\[ E_{t+2} \]
\[ D_{t+2} \text{ in } P(X_{t+2} | E_{1:t+2}) \]
\[ E_{t+3} \]
\[ U(X_{t+3}) \]
Wrt Dynamic Decision Networks

- Handle uncertainty correctly… sometimes efficiently…
- Deal with streams of sensor input
- Handle unexpected events (as have no fixed “plan”)
- Handle noisy sensors, sensor failure
- Act in order to obtain information as well as to receive rewards
- Handle relatively large state spaces as they decompose state into set of state var's with sparse connections
- Exhibit graceful degradation under time pressure and in complex environments using various approximation techniques
Open Problems wrt Probabilistic Agents

- First-order probabilistic representations
  
  *If any car hits lamp post going over 30mph, occupants of car injured with probability 0.60.*

- Methods for scaling up MDP's

- More efficient algorithms for POMDP's

- Learning environment
  
  - $M^a_{ij}$, $P(E | S)$, ...
Probabilistic Agents Summary

- **Three key components:**
  - $P( S' \mid S,A)$ (action model)
  - $P( E \mid S )$ (sensor model)
  - $R( S' \mid S,A)$ (reward function)

- In accessible environments,
  - \{ Value iteration, Policy iteration \} work well.
  - Each computes local (state) utility function, optimal policy.

- In \{ unobservable, partially-observable \} environments,
  - lookahead search gives approx solutions
    - Updating current beliefs in a DDN is easy.
    - Look-ahead search is hard.