Markov decision processes (MDP)

• Initial State
  - \( S_0 \)
• Transition Model
  - \( T(s, a, s') \)
    - How does Markov apply here?
    - Uncertainty is possible
• Reward Function
  - \( R(s) \)
    - For each state

Building an optimal policy

• Value Iteration
  - Calculate the utility of each state
  - Use the state utilities to select an optimal action in each state
  - Your policy is simple – go to the state with the best utility
  - Your state utilities must be accurate
    - Through an iterative process you assign correct values to the state utility values

Iterative solution of Bellman equations

- Start with arbitrary initial values for state utilities
- Update the utility of each state as a function of its neighbors

\[
U(s) = R(s) + \gamma \max \sum T(s, a, s') U(s')
\]

Example

- Let \( \gamma = 1 \) and \( R(s) = -0.04 \)

Notice:
- Utilities higher near goal reflecting fewer -0.04 steps in sum

Building a policy

- How might we acquire and store a solution?
  - Is this a search problem?
    - Isn’t everything?
  - Avoid local mins
  - Avoid dead ends
  - Avoid needless repetition

Key observation: if the number of states is small, consider evaluating states rather than evaluating action sequences
Policy Iteration

• Imagine someone gave you a policy
  – How good is it?
    • Assume we know $\gamma$ and $R$
    • Eyeball it?
      • Try a few paths and see how it works?
      • Let’s be more precise…

Policy iteration

• Checking a policy
  – Just for kicks, let’s compute a utility (at this particular iteration of the policy, $i$) for each state according to Bellman’s equation

\[ U_i(s) = R(s) + \gamma \sum_{s'} T(s, \pi_i(s), s')U_i(s') \]

Policy iteration

• Checking a policy
  – But we don’t know $U_i(s')$
  – No problem
    • $n$ Bellman equations
    • $n$ unknowns
    • equations are linear (in value iteration, the equations had the non-linear “max” term)
  – We can solve for the $n$ unknowns in $O(n^3)$ time using standard linear algebra methods

Policy Iteration

• Often the most efficient approach
  – Requires small state spaces to be tractable: $O(n^2)$
  – Approximations are possible
    • Rather than solve for $U$ exactly, approximate with a speedy iterative technique
    • Explore (update the policy of) only a subset of total state space
      – Don’t bother updating parts you think are bad

Can MDPs be used in real situations?

• Remember our assumptions
  – We know what state we are in, $s$
  – We know the reward at $s$
  – We know the available actions, $a$
  – We know the transition function, $t(s, a, s')$
Is life fully observable?

- We don’t always know what state we are in
  - Frequently, the environment is partially observable
    - agent cannot look up action, $\pi(s)$
    - agent cannot calculate utilities

We can build a model of the state uncertainty and we call them
Partially Observable MDPs (POMDPs)

Our robot problem as a POMDP

- No knowledge of state
  - Robot has no idea of what state it is in
  - What’s a good policy?

- The “Drunken Hoo” strategy

Observation Model

- To help model uncertainty
  - Observation Model, $O(s, o)$
    - specifies the probability of perceiving the observation $o$ when in state $s$
      - In our example, $O()$ returns nothing with prob. 1

Belief state

- To help model uncertainty
  - A belief state, $b$
    - the probability distribution over being in each state, $b(s)$
      - initial $b = (1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0)$
    - $b(s)$ will be updated with each new observation / action

\[ b(s) \] normalizes equation so $b$ sums to 1.0
\[ b(s) = \alpha O(s, a) \sum_{s'} F(s,a,s') b(s') \]

Insight about POMDPs

- Beliefs are more important than reality
  - Optimal action will depend on agent’s belief state
    - not its actual state
    - $\pi'(b)$ maps belief state to actions
  - Think about “The Matrix”

POMDP agent

- A POMDP agent iterates through following steps
  - Given current belief state, $b$, execute the action $a = \pi'(b)$
  - Receive the observation $o$
  - Use $a$ and $o$ to update the belief state
  - Repeat
Mapping MDPs to POMDPs

- What is the probability an agent transitions from one belief state to another after action \( a \)?
  - We would have to execute the action to obtain the new observation if we were to use this equation
    \[
    b'(x) = O(x, a) \sum_s T(s, a, s') b(s)
    \]
  - Instead, use conditional probabilities to construct \( b' \) by summing over all states agent might reach

Predicting future observation

- Prob of perceiving \( o \) given
  - starting in belief state \( b \)
  - action \( a \) was executed
  - \( s' \) is the set of potentially reached states

\[
\begin{align*}
P(o|a, b) &= \sum_{s'} P(o|a, s') P(s'|a, b) \\
&= \sum_{s'} O(s', o) P(s'|a, b) \\
&= \sum_{s'} O(s', o) \sum_s T(s, a, s') b(s)
\end{align*}
\]

Predicting new belief state

- Previously we predicted observation…
  Now predict new belief state
  - \( \tau(b, a, b') \)
    - prob of reaching \( b' \) from \( b \) given action \( a \)

\[
\begin{align*}
\tau(b, a, b') &= P(b'|a, b) = \sum_{s'} P(b'|s, a, b) P(s'|a, b) \\
&= \sum_{s'} P(b'|s, a, b) \sum_s O(s', o) \sum_s T(s, a, s') b(s)
\end{align*}
\]
  - This is a transition model for belief states

Computing rewards for belief states

- We saw that \( R(s) \) was required…
  - How about \( R(b) \)?
    - call it \( \rho(b) \)

\[
\rho(b) = \sum_s b(s) R(s)
\]

Pulling it together

- We’ve defined an observable MDP to model this POMDP
  - \( \tau(b, a, b') \) and \( \rho(b) \) replace \( t(s, a, s') \) and \( R(s) \)
  - The optimal policy, \( \pi^*(b) \) is also an optimal policy for the original POMDP

An important distinction

- The “state” is continuous in this representation
  - The belief state of the 4x3 puzzle consists of a vector of 11 numbers (one cell is an obstacle) between 0 and 1
    - The state in our older problems was a discrete cell ID
  - We cannot reuse the exact value/policy iteration algorithms
    - “Summing” over states is now impossible
    - There are ways to make them work, though
Truth in advertising

• Finding optimal strategies is slow
  – It is intractable for problems with a few dozen states