CS 416
Artificial Intelligence

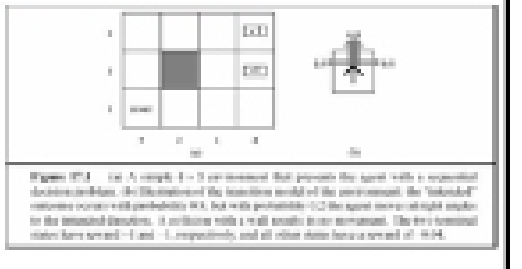
Lecture 19

Making Complex Decisions

Chapter 17

Robot Example

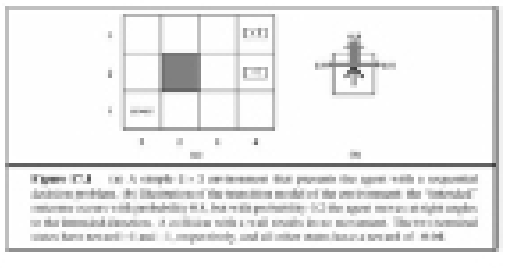
- Imagine a robot with only local sensing
  - Traveling from A to B
  - Actions have uncertain results – might move at right angle to desired
  - We want robot to "learn" how to navigate in this room



- Sequential Decision Problem

Similar to 15-puzzle problem

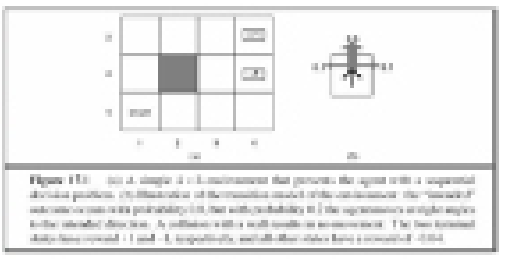
- How is this similar and different from 15-puzzle?
  - Let robot position be the blank tile
  - Keep issuing movement commands
  - Eventually a sequence of commands will cause robot to reach goal



Our model of the world is incomplete

How about other search techniques

- Genetic Algorithms
  - Let each "gene" be a sequence of L, R, U, D
    - Length unknown
    - Poor feedback
- Simulated annealing?

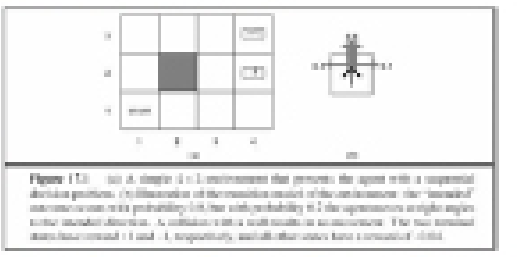


Markov decision processes (MDP)

- 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 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*

Building a policy

- Specify a solution for any initial state
  - Construct a policy that outputs the best action for any state
    - policy = π
    - policy in state s = $\pi(s)$
  - Complete policy covers all potential input states
  - Optimal policy, π^*, yields the highest expected utility
    - Why expected?
      - Transitions are stochastic

Using a policy

- An agent in state s
  - s is the percept available to agent
  - $\pi^*(s)$ outputs an action that maximizes expected utility
- The policy is a description of a simple reflex

Example solutions

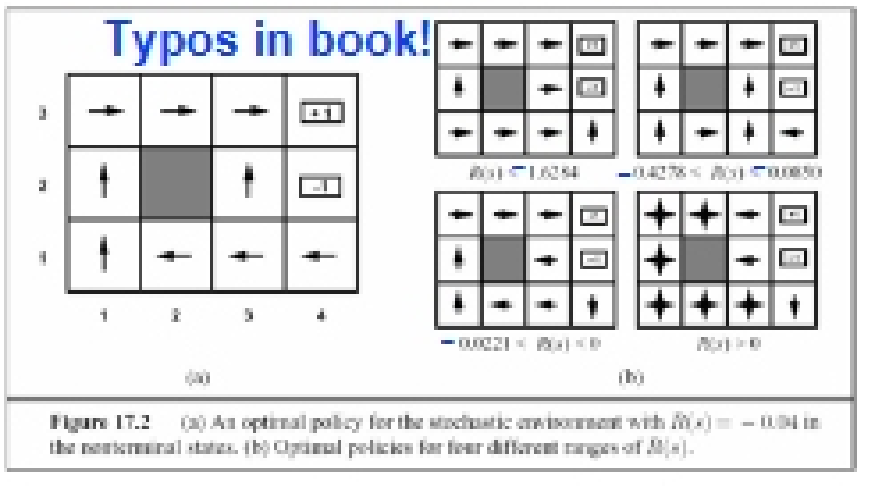


Figure 17.2 (a) An optimal policy for the stochastic environment with $R(s) = -0.04$ in the nonterminal states. (b) Optimal policies for four different ranges of $R(s)$.

Striking a balance

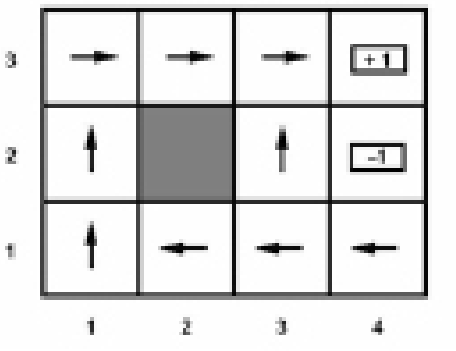
- Different policies demonstrate balance between risk and reward
  - Only interesting in stochastic environments (not deterministic)
  - Characteristic of many real-world problems
- Building the optimal policy is the hard part!

Attributes of optimality

- We wish to find policy that maximizes the utility of agent during lifetime
  - Maximize $U([s_0, s_1, s_2, \ldots, s_n])$
- But is length of lifetime known?
  - Finite horizon – number of state transitions is known
    - After timestep N, nothing matters
  - $U([s_0, s_1, s_2, \ldots, s_n)] = U([s_0, s_1, s_2, \ldots, s_n, s_{n+1}, s_{n+k})]$ for all $k>0$
  - Infinite horizon – always opportunity for more state transitions

Time horizon

- Consider spot (3, 1)
  - Let horizon = 3
  - Let horizon = 8
  - Let horizon = 20
  - Let horizon = inf
  - Does π^* change?



- Nonstationary optimal policy

Evaluating state sequences

- Assumption
  - If I say I will prefer state a to state b tomorrow I must also say I prefer state a to state b today
  - State preferences are stationary
- Additive Rewards
  - U[(a, b, c, ...)] = R(a) + R(b) + R(c) + ...
- Discounted Rewards
  - U[(a, b, c, ...)] = R(a) + γR(b) + γ^2R(c) + ...
  - γ is the discount factor between 0 and 1
    - What does this mean?

Evaluating infinite horizons

- How can we compute the sum of infinite horizon?
  - U[(a, b, c, ...)] = R(a) + R(b) + R(c) + ...
  - If discount factor, γ, is less than 1

$$U_h([s_0, s_1, s_2, \ldots]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq \sum_{t=0}^{\infty} \gamma^t R_{\max} = R_{\max}/(1-\gamma)$$

  - note R_{max} is finite by definition of MDP

Evaluating infinite horizons

- How can we compute the sum of infinite horizon?
  - If the agent is guaranteed to end up in a terminal state eventually
    - We'll never actually have to compare infinite strings of states
    - We can allow γ to be 1

Evaluating a policy

- Each policy, π, generates multiple state sequences
  - Uncertainty in transitions according to T(s, a, s')
- Policy value is an expected sum of discounted rewards observed over all possible state sequences

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi\right]$$

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

Utility of states

- The utility of a state s is...
  - the expected utility of the state sequences that might follow it
    - The subsequent state sequence is a function of $\pi(s)$
- The utility of a state given policy π is...

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi, s_0 = s\right]$$