Decision-Making

Non-recap of last class

- We'll return to planning next week…

Agenda

- Simple and complex decision-making
- Markov Decision Problems
- Concept of Utility
- Value and policy iteration

Expected Utility

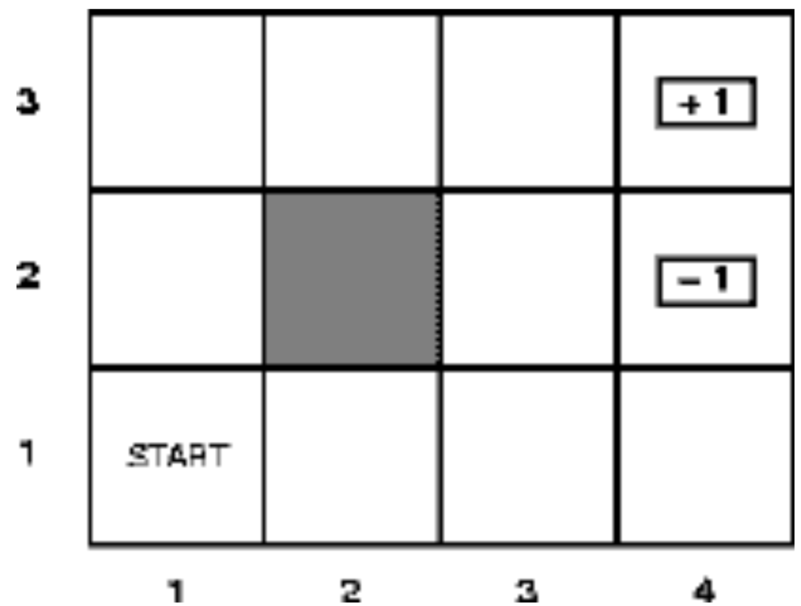
- $EU(a|\mathbf{e}) = \Sigma_{s'} P(\text{RESULT}(a) = s' \mid a, \mathbf{e})\, U(s')$

- Maximum Expected Utility
  - rational agent should choose action that maximizes expected utility:

$$a^* = \underset{a}{\operatorname{argmax}}\, EU(a \mid \mathbf{e})$$

Making Complex Decisions

- from START, agent executes a sequence of actions (north, south, east, west), terminating when it reaches one of the terminal states with a reward of +1 or -1
- all other states have reward of -0.04 (think of this as a path cost)

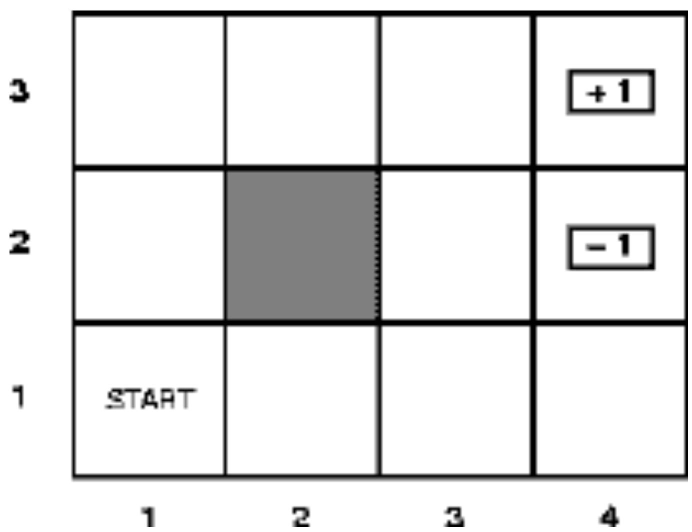


Deterministic case

- if we know where we started and what happens when we move in any direction:
  - can build entire state tree
  - use classical search techniques to find optimal solution

Non-deterministic case

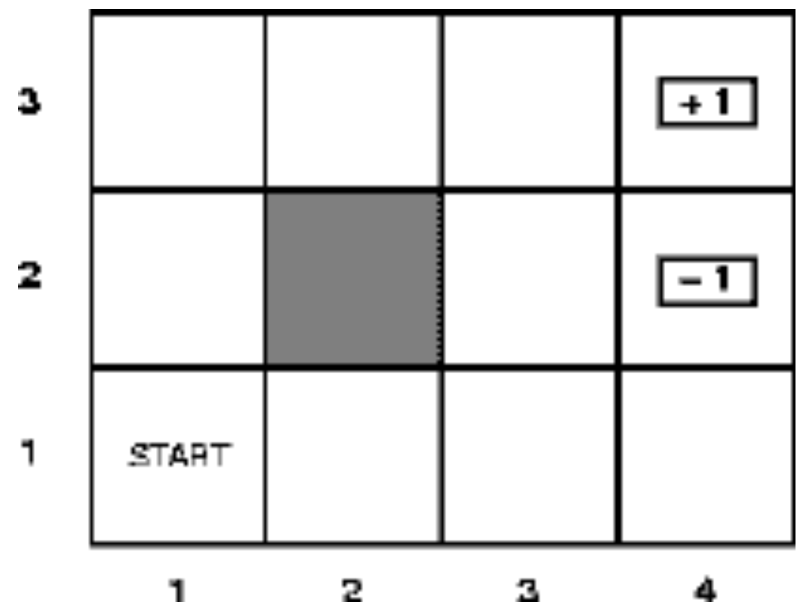
- 0.8 probability that each action achieves intended effect
- transition model : *P(s' | s,a) or equivalently, T(s,a,s')* refers to probability of reaching state *s'* if action *a* performed in state *s*
- can't search!



Conditional Plan

- need a solution more like this:

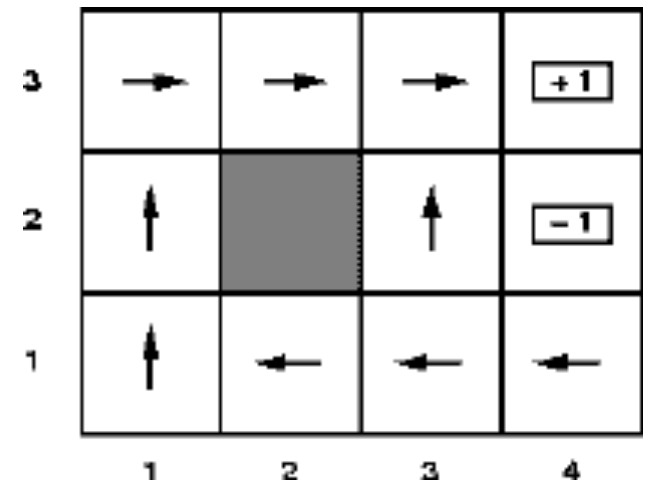
North
if (2,1) then West
else North
...



Markov decision problems (MDP)

- sequential decision problem
- environment is fully observable
- transition probabilities depend only on current state (memoriless)
- defined by:
  - initial state: S_0
  - transition model: $T(s,a,s')$ or $P(s' | s,a)$
  - reward function: $R(s)$

Policy solution to MDP



- *π(s):* what should the agent do for any state *s* that it might reach?

- *π*(s):* optimal policy, yields highest expected utility

Utility Function in an MDP

- how good is a particular state?
- because the decision problem is sequential, the utility function depends on a sequence of states

"the utility of a state is the expected utility of the state sequences that might follow it"

Utility of state sequence U_h

- for additive rewards

  $U_h([s_0, s_1, \ldots, s_n]) = R(s_0) + U_h([s_1, \ldots, s_n]) = \Sigma R(s_i)$

- unbounded world problem: what if there are positive rewards at non-terminal states?

Discounting

- concept of "discounted rewards":
  - rewards are less valuable the longer we wait for them
  - $U_h([s_0,s_1,\ldots,s_n]) = R(s_0) + \gamma U_h([s_1,\ldots,s_n])$
    $= R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots = \Sigma \gamma^i R(s_i)$

    where γ is the discount factor (< 1) for the wait
    (*$\gamma=1$ degenerates to the additive case)*
  - ensures that utility of an infinite sequence is *finite*

Utilities of states

- the utility of a state is the expected utility of the state sequences that might follow it

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

- therefore, the true utility of a state $U(s) = U^{\pi^*}(s)$

Optimal Policy

- choose action that achieves maximum expected utility of subsequent state
- hence, the optimal policy is:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi\right].$$

But how to solve these series?

- observation:
  - direct relationship between utility of a state and its neighbours:
- Bellman equation:
  - utility of a state = immediate reward for that state…

$$U(s) = R(s) +$$

But how to solve these series?

- observation:
  - direct relationship between utility of a state and its neighbours:
- Bellman equation:
  - utility of a state = immediate reward for that state…
  - plus expected discounted utility of the next state…

$$U(s) = R(s) + \gamma \sum_{s'} P(s' \mid s, a) U(s')$$

But how to solve these series?

- observation:
  - direct relationship between utility of a state and its neighbours:
- Bellman equation:
  - utility of a state = immediate reward for that state…
  - plus expected discounted utility of the next state…
  - following the optimal policy

$$U(s) = R(s) + \gamma \max_a \sum_{s'} P(s' | s, a) U(s')$$

Value Iteration

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s' \mid s, a) U_i(s')$$

-.04	-.04	-.04	+1
-.04		-.04	-1
-.04	-.04	-.04	-.04

Example: Value Iteration applied

.812	.868	.912	+1
.762		.650	-1
.705	.655	.611	.388

Policy Iteration

- the policy evaluation step can be solved directly in $O(n^3)$ using linear algebra techniques
- but we can approximate this by a simplified Bellman update (modified policy iteration):

$$U_{i+1}(s) \leftarrow R(s) + \gamma \sum_{s'} P(s' | s, \pi_i(s)) U_i(s')$$

Policy Iteration Algorithm

pick an initial policy π_0 (randomly)

then iterate:

- policy evaluation:
  - calculate utility of each state, given π_i: $U_i = U^{\pi_i}(s)$

    $$U_{i+1}(s) = R(s) + \gamma \sum_{s'} P(s' \mid s, \pi_i(s)) U_i(s')$$

  - simpler than value iteration because actions are fixed!
- policy improvement:
  - calculate a new MEU policy π_{i+1} using one-step look-ahead based on U_i

until no change in policy

Policy Iteration Algorithm

pick an initial policy π_0 (randomly)

repeat

    U ← POLICY-EVALUATION (π, U, mdp)

    unchanged ←TRUE

    for each state s in S do

        if $\max_a \sum P(s' \mid s,a)U[s'] > \sum P(s' \mid s,\pi[s])U[s']$)

            $\pi[s] \leftarrow \arg\max_a \sum P(s' \mid s,\pi[s])\, U[s']$

            unchanged ←FALSE

until unchanged

Recap

- Basics of decision-making
- Markov Decision Problems
- Concept of Utility
- Value and policy iteration