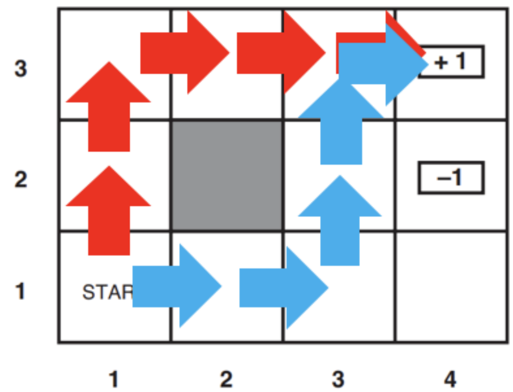
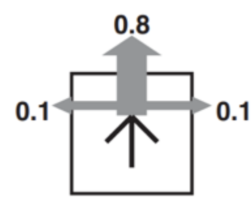
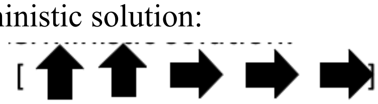
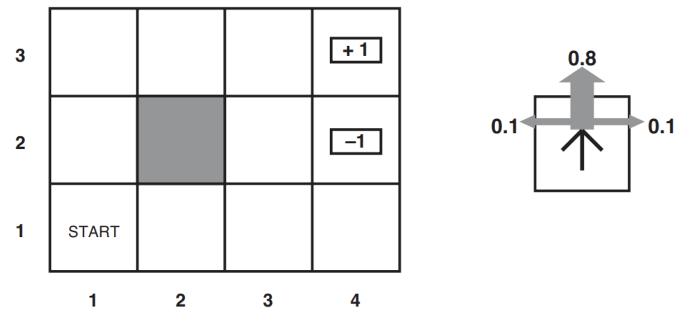
CAS CS 440
Lec 27

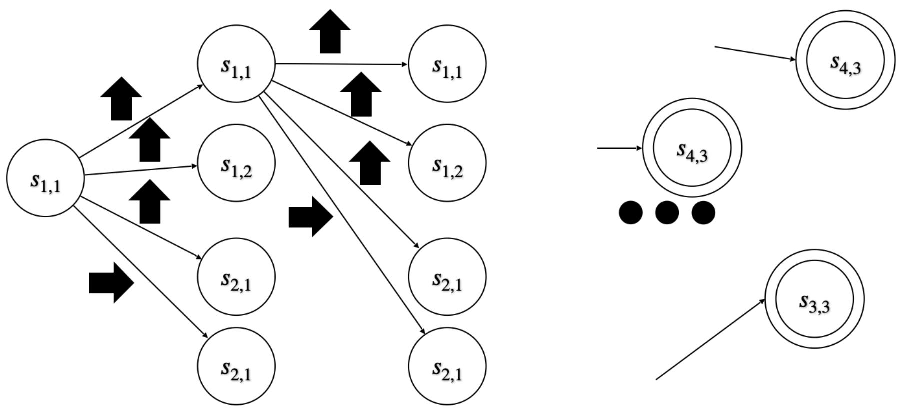
Policy Learning I

1. Sequential Problems
    a. The world
        i. 



            1. The world is described above
            2. You can only move left or right or up or down (no diagonals)
            3. If you decide to move up, the probability of going up is 80%, and 10% of the time, you will go left or right
            4. The action does not always lead to the result intended (the world can take my action and not do that action for some percentage)
            5. Would have done A* algorithm to run it
    b. Stochastic environment
        i. No longer an “easy” search
        ii. Deterministic solution:
            1.  → I could have taken this action
            2. Actions unreliable! May not go according to plan!
            3. Solution reaches +1 goal with Prob 0.32768
            4. Also reaches +1 goal (by other way around) with Probl 0.32768
    c. Transition model Pr[s’ | s, a] is a general pmf
        i. In deterministic world Pr[s’ | s, a] = 1 when s’ is the intended state from applying ‘a’ in ‘s’

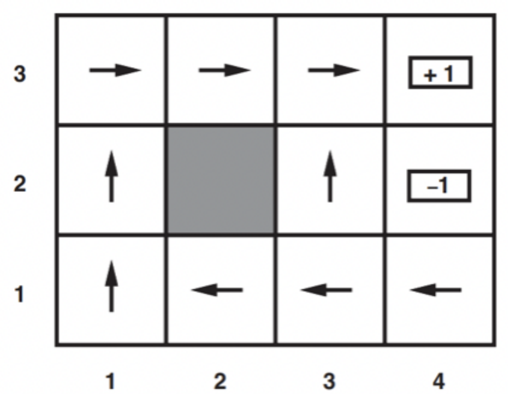
2. Trajectories
    a. 

    b. The observed sequence of states (and actions) is called a trajectory
    c. From a single start state, lots of trajectories possible
    d. 

3. Markovian Assumption & Utilities
    a. Transition model

$$\Pr\left[s' \middle| s, a\right] = \Pr\left[s' \middle| s_t,\ s_{t-1}\ ,\ \ldots,\ s_0, a\right]$$

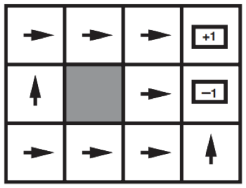
        i. Transition probability does not depend on history, only on current state and action
    b. Utility function?
        i. No longer dependent on a single state
        ii. Depends on sequence of states
        iii. Define a reward function: R(s)
            1. Must be bounded (can either be + or -)
4. Utilities from Rewards
    a. Many ways of doing this
    b. For now, consider utility = sum of rewards along that trajectory
    c. Utility of a 10-step trajectory (ending in +1 goal) = 0.6
        i. Negative reward incentivizes agent to find solution early
        ii. (We design our agents to maximize utility)

5. MDP
    a. A sequential decision problem
        i. Fully observable world
        ii. Stochastic env
        iii. Markovian transition model
        iv. Additive rewards
    b. This is called a Markov Decision Process
        i. Set of states (with initial state)
        ii. Actions for each state
        iii. Transition model
        iv. Reward function
6. Policies
    a. Cannot calculate fixed action sequence
    b. Instead, what action should we pick at each state?
    c. This function is called a policy $\pi: S \rightarrow A$ (maps states to actions)
        i. $\pi(s) = a$ means "policy π recommends action 'a' in state 's'
    d. A complete policy recommends (an) action(s) for every state
        i. Agent always knows what to do next
7. Optimal Policies
    a. Lets say we have policy π
        i. Every time we start a new "episode"
        ii. Trajectory may be different!
    b. 

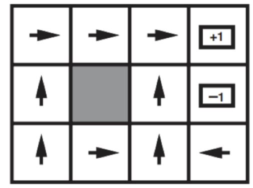
    c. The quality of a policy is the expected utility of possible trajectories generated by that policy
    d. An optimal policy maximizes this expected utility

$$\pi_s^* = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_{\tau \sim \pi}\left[U_h(\tau)\right]$$

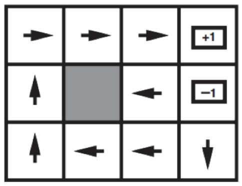
8. Optimal Policies are Sensitive to Rewards



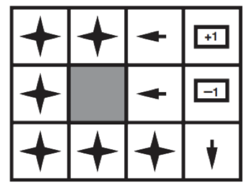
$R(s) < -1.6284$



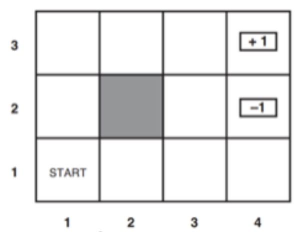
$-0.4278 < R(s) < -0.085$

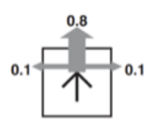
a. 

$-0.0221 < R(s) < 0$



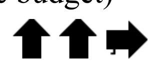
b. Depending on the reward function, we can
    i. Just try to reach the terminal state since every step is very painful
    ii. Try to reach +1 but the cost of going around of -1 is too high so just take the risk of going to -1 (do not go around)
    iii. Try to reach +1
    iv. Just never die and do not move

9. Finite Horizons
    a. A finite horizon = there is a fixed time N after which no further actions matter (i.e. we consider the game is over even without reaching terminal state)
    b. 



    c. Utility for a trajectory

$$U_h(s_0, s_1, \ldots s_{N+k}) = U_h(s_0, s_1, \ldots, s_N)\ \forall k > 0$$

    i. For example: let us set N = 3 (agent has a 3 move budget)
    ii. IF agent starts at (3,1) instead of (1,1) → π^* = [
    iii. π^* is sensitive to N! (optimal policy for finite horizons is nonstationary)

10. Infinite Horizons
    a. No maximum move budget
    b. No reason to behave differently in the same state at different times
    c. π^* is stationary
    d. π^* are simpler for infinite horizon settings than for finite horizons

11. $U_h(\tau)$ (utility)

   a. **Additive rewards**: The utility of a state sequence is
   $$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \cdots .$$

   b. **Discounted rewards**: The utility of a state sequence is
   $$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots ,$$

      i. Gamma will discount the value

      ii. Want the reward to care less in the future (further away from now, the less I should care)

   c. Discount factor $\gamma \in [0,1]$. When $\gamma = 1$, we consider the future states as equally valuable to where we are now. $\gamma = 0$ claims the future is insignificant

12. Problem with Additive Rewards

   a. Trajectories can go on forever

      i. Additive rewards → +- infinity

   b. Discounted rewards are finite (when $\gamma < 1$)

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

   d. If env contains terminal states and agent is guaranteed to hit them (eventually), trajectories are finite!

      i. Policy that guarantees hitting a terminal state is called a proper policy

13. Another Way to Write Optimal Policies

   a. Since discounted rewards is more general than additive:

      - $$\mathbb{E}_{\tau \sim \pi}\left[U_h(\tau)\right] \rightarrow U^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t)\right]$$

   b. $$\pi_s^* = \underset{\pi}{\operatorname{argmax}} U^{\pi}(s)$$