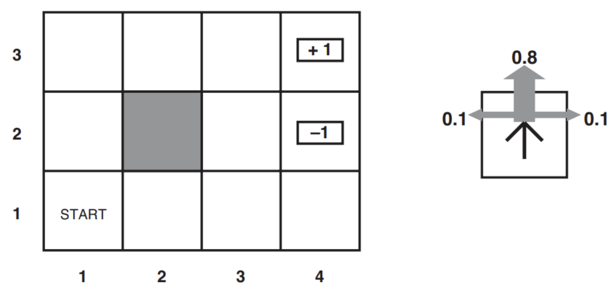
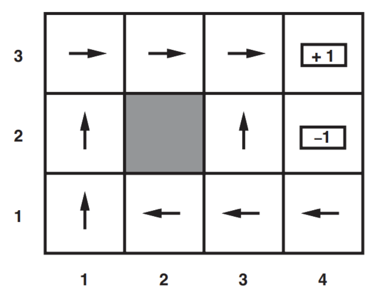
CAS CS 440
Lec 28

Policy Learning II

1. Takeaways from Last Time
    a. World is stochastic
        i. There is a possibility that the agent does not follow the plan but may go to other directions
            1. In this case, it follows 80% of time but goes to the wrong direction 20% of time
        ii. Plans mean nothing now
        iii. Transition model is probabilistic
            1. Markovian assumption (the current move does not depend on the history but is new case every time)
    b. 

    c. 

    d. A policy $\pi : S \rightarrow A$ is a map from states to actions
        i. Optimal policy (from state s) $\pi_s^* = \underset{\pi}{\operatorname{argmax}} U^\pi(s)$ → tells the best move

$$U^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$

→ depending on utility function
            1. Discount the reward at each state → gamma
            2. So that we do not care more about the future state
            3. Discounting factor is typically less than 1
            4. The utility function is bounded

ii. Infinite horizon = no move budget → optimal policy is stationary
   1. I do not care where I start (unlimited number of moves)
   2. The best action to choose does not change when I am at the state
iii. Infinite horizon + discounted rewards → optimal policy independent of starting state

e. $$\pi^*(s) = \underset{a \in Actions(s)}{\operatorname{argmax}} \sum_{s'} \Pr[s' | s, a] U(s')$$

   i. Pick 'a' with best expected outcome → argmax
   ii. What to do at state 's' → $\pi^*(s)$
   iii. All actions available in s → Actions(s)
   iv. How good is s' → $U(s')$
   v. How likely to get to s' from 's' using 'a' → $\Pr[s' | s, a]$
      1. Where does that state lead
   vi. All ways of resolving 'a' in 's' → $\sum$

2. How to Calculate Optimal Policies?
   a. Lots of research
   b. Today:
      i. Value iteration
      ii. Policy iteration
3. Value Iteration
   a. General idea:
      i. Calculate U(s) for each state
      ii. Use utilities to select optimal action in each state
   b. Observation:
      i. $$U(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$
      ii. The utility of a state is related to neighbor's utility
      iii. Assuming optimal action is chosen:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} \Pr[s' | s, a] U(s')$$

         1. The action I do, added with the future states with gamma value
      iv. This is called the Bellman Equation

c. If we know U(s) for each state, we can select the optimal action

3	0.812	0.868	0.918	+1
2	0.762		0.660	–1
1	0.705	0.655	0.611	0.388
	1	2	3	4

$$
\begin{aligned}
U(1,1) = -0.04 + \gamma \max[\ & 0.8\, U(1,2) + 0.1U(2,1) + 0.1U(1,1), \\
& 0.8\, U(1,1) + 0.1U(1,1) + 0.1U(1,2), \\
& 0.8\, U(1,1) + 0.1U(1,1) + 0.1U(2,1), \\
& 0.8\, U(2,1) + 0.1U(1,2) + 0.1U(1,1)\ ]
\end{aligned}
$$

Reward for acting

d.
   i. Populate each state with its utility
   ii. We punish the agent for existing and want to solve it quickly, which is why it starts at -0.04

4. Value Iteration
   a. How to get these utilities?
      i. Each state gets its own bellman equation (for that state)
      ii. n states → n equations
      iii. The n equations collectively have n unknowns
      iv. Can we solve?
         1. No, due to the max
   b. Problem: Bellman is nonlinear: cannot represent with linear algebra
      i. Good news: can solve iteratively!
      ii. Initially set each (nonterminal) state's utility to zero
      iii. Apply update (simultaneously to every state) until convergence!

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

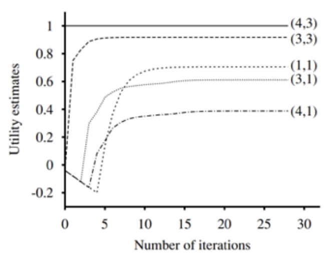
5. The Value Iteration Algorithm

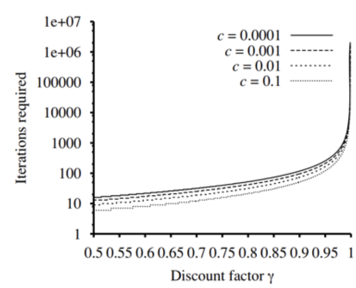
function VALUE-ITERATION(mdp, ϵ) **returns** a utility function
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$,
    rewards $R(s)$, discount γ
    ϵ, the maximum error allowed in the utility of any state
  local variables: U, U', vectors of utilities for states in S, initially zero
    δ, the maximum change in the utility of any state in an iteration

  repeat
    $U \leftarrow U'; \delta \leftarrow 0$
    for each state s **in** S **do**
      $U'[s] \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s']$ Run the Bellman Equation
      if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$ Calculate Error between steps
  until $\delta < \epsilon(1-\gamma)/\gamma$
  return U

   a.

6. Value Iteration
   a. Value iteration propagates information from terminal states to nonterminal states using the bellman equation
   b. Since rewards don't change over time in this world, this algorithm converges!
      i. The rewards are constant
   c. Extra bonus: upon convergence, utilities are unique solutions!
      i. Guaranteed to be optimal policy (stationary for infinite horizons)





   d.

7. Why does Value Iteration Converge?
   a. The bellman equation is a contraction function
      i. f is a contraction function iff

$$\forall x, y \ \ d\big(f(x),\ f(y)\big) \le k d(x, y) \ \ (0 \le k < 1)$$

      ii. "The outputs of applying f to x and y is "close" (at least by a constant factor) to the original values x and y

Example: division (by c)

$$\left|\frac{x}{c} - \frac{y}{c}\right| \le \frac{1}{c}|x - y|$$

      iii.

iv. Contractions have one “fixed point” z where the contraction has no effect:

1. f(z) = z

v. $\forall y \neq x \quad d\big(f(y), f(x)\big) \leq kd\big(y, x\big) \leq d(y, x)$

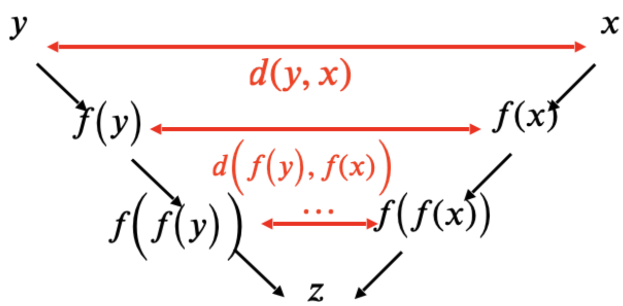
1. (i.e. all points y get closer to fixed point x)

b. For bellman (in vector form):

i. $\vec{u}_{i+1} \leftarrow B\left(\vec{u}_i\right)$

c. Using l_∞ norm:

$$\frac{\left\| B\left(\vec{u}_i\right) - B\left(\vec{u}_i'\right) \right\|_\infty}{\left\| \vec{u}_i - \vec{u}_i' \right\|_\infty} \leq \gamma$$

d. 

8. Value Iteration and Inaccurate Utilities

a. If our utilities are wrong (i.e. inaccurate), policy iteration still works

i. Comes from being a contractor

ii. $\left\| B\left(\vec{u}_i\right) - \vec{u}^* \right\|_\infty \leq \gamma \left\| \vec{u}_i - \vec{u}^* \right\|_\infty$ where ‘u’ is the true utility values

iii. $\left\| \vec{u}_i - \vec{u}^* \right\|_\infty$ is the error in our current utilities ‘u_i’

iv. Takes $\left\lceil \frac{\log\left(\frac{2R_{max}}{\epsilon(1-\gamma)}\right)}{\log\frac{1}{\gamma}} \right\rceil$ iterations to get ‘u_i’ within ϵ of u’

b. Can also terminate early

i. Don’t need exactly utilities to be correct

ii. Just need to be able to infer correct actions

iii. Policy loss = $\left\| U^{\pi_i} - U^{\pi^*} \right\|_\infty$ utility lost by policy π_i instead of following π^*

iv. Bounded by error in utilities:

$$\left\| \vec{u}_i - \vec{u}^* \right\|_\infty \leq \epsilon \rightarrow \left\| U^{\pi_i} - U^* \right\|_\infty < \frac{2\epsilon\gamma}{1-\gamma}$$

$$\frac{-R_{max}}{1-\gamma} \leq U(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(S_t) \right] \leq \frac{+R_{max}}{1-\gamma}$$

c.

$$\left\| \vec{u}_0 - \vec{u}^* \right\|_\infty \leq \frac{2R_{max}}{1-\gamma}$$
$$\left\| \vec{u}_1 - \vec{u}^* \right\|_\infty \leq \gamma \frac{2R_{max}}{1-\gamma}$$
$$\left\| \vec{u}_2 - \vec{u}^* \right\|_\infty \leq \gamma^2 \frac{2R_{max}}{1-\gamma}$$
$$\vdots$$
$$\left\| \vec{u}_t - \vec{u}^* \right\|_\infty \leq \gamma^t \frac{2R_{max}}{1-\gamma} \leq \epsilon$$

d.

$$\left\| \vec{u}_i - \vec{u}_i^{\prime} \right\|_\infty < \frac{\epsilon(1-\gamma)}{\gamma} \rightarrow \left\| \vec{u}_i - \vec{u}^* \right\|_\infty < \epsilon$$

Halting criteria for VI