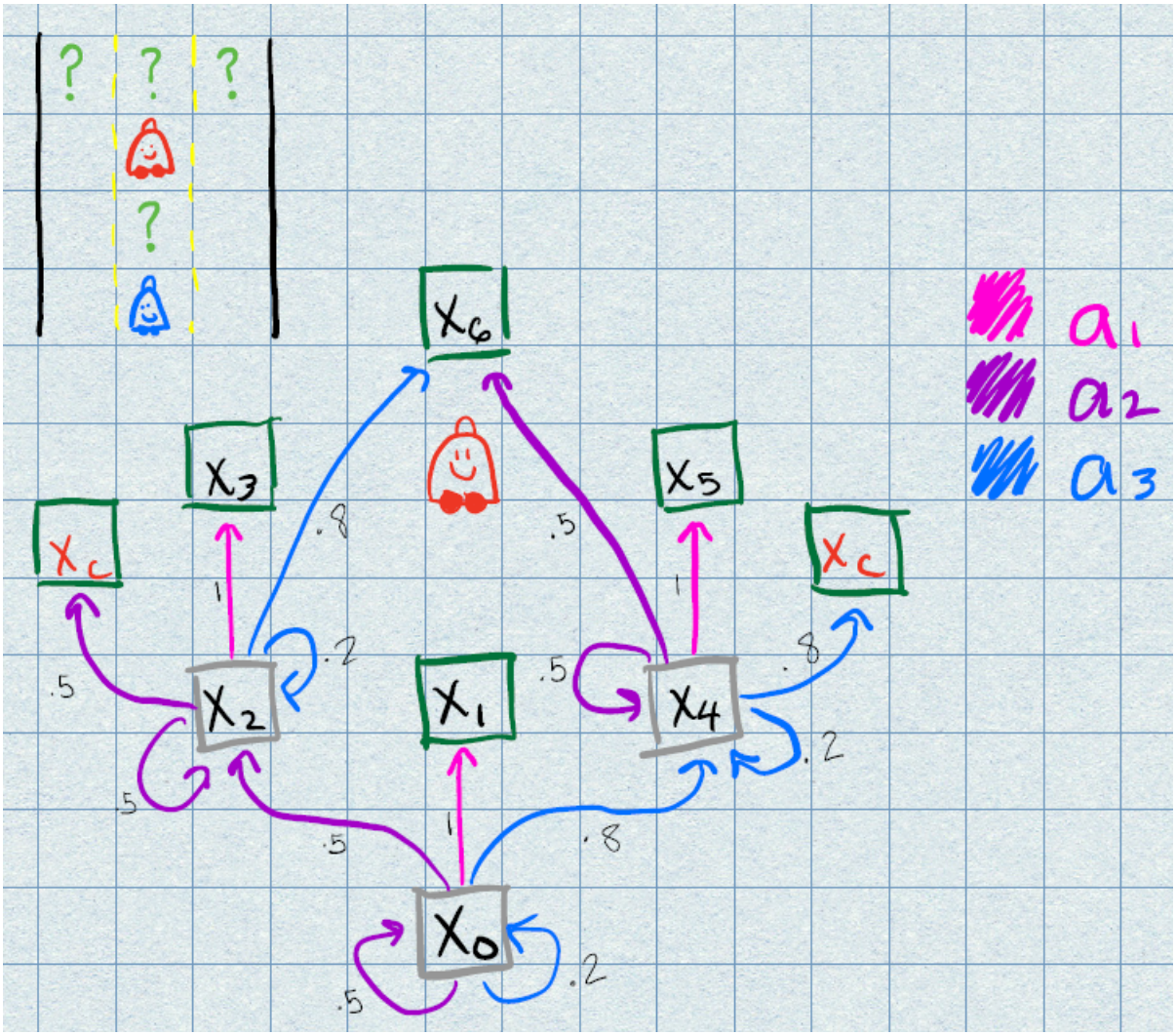
Let's walk through a more complicated example of approximate value iteration!

Suppose we have a simple driving scenario, where we are driving on a three lane road with a slow moving car in front of us. We'd like to decide whether or not we should follow this car, change lanes, or try to over-take. At each time step, we can execute a right or left lane change or stay in our lane. For each of these three high-level actions, we assume that we have implemented a low-level controller that can follow the command. A visualization of the scenario, and the associated MDP, is shown below.



If we choose to stay in our lane (a_1), we proceed forward by one state with probability 1. If we lane change left, we will shift to the left with probability 0.5 or we will stay in our current state with probability 0.5. If we lane change right, we will shift to the right with probability 0.8 or we will stay in our current state with probability 0.2.

If we perform a left (right) lane change in the left-most (right-most) lane, we enter a collision state. Terminal states (i.e., we no longer transition to future states) are if we continue to stay in our lane (x_1), continue in the left-most or right-most lane (x_3 or x_5, respectively), complete an overtaking maneuver (x_6), or collide with the road edge (x_c).

If we stay in the middle lane (reach x_1), we get reward R_S. If we move to the left lane (reach x_3), we get reward R_L. If we move to the right lane (reach x_5), we get reward R_R. If we perform an overtake (reach

x_6), we get reward R_O. If we exit the road (reach x_c), we get penalty $-R_C$. Formally:

$$R(x,a) = \begin{cases} R_S & \text{if } x_1 \\ R_L & \text{if } x_3 \\ R_R & \text{if } x_5 \\ R_O & \text{if } x_6 \\ -R_C & \text{if } x_c \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Some things to think about:

- Is this a reasonable model? How could it be improved?
- How do we get those transition probabilities?
- What are some of the built in assumptions?

Recall that we iteratively update our value approximation:

$$\hat{V}_k(x_i) = \gamma \max_u \left[r(x_i,u) + \sum_j \hat{V}_{k-1}(x_j) p(x_j|x_i,u) \right] \tag{2}$$

where k is the iteration, x_i is the state we are evaluating, $r(x_i,u)$ is the immediate payoff, x_j are possible future states, $\hat{V}_{k-1}(x_j)$ is the approximate value of states you may transit to, and $p(x_j|x_i,u)$ is the transition function.

Let's initialize the vector representation of $\hat{V}_0$:

$$\hat{V}_0 = \begin{bmatrix} \hat{V}_0(x_0) & \hat{V}_0(x_1) & \hat{V}_0(x_2) & \hat{V}_0(x_3) & \hat{V}_0(x_4) & \hat{V}_0(x_5) & \hat{V}_0(x_6) & \hat{V}_0(x_c) \end{bmatrix} \tag{3}$$

$$\hat{V}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

For the following iterations, we will write the maximization of a vector, to show the evaluations for each possible action, i.e.:

$$\max_u \left[r(x_i,u) + \sum_j \hat{V}_{k-1}(x_j) p(x_j|x_i,u) \right] = \max_{u \in \{a_1,a_2,a_3\}} \begin{bmatrix} r(x_i,a_1) + \sum_j \hat{V}_{k-1}(x_j) p(x_j|x_i,a_1) \\ r(x_i,a_2) + \sum_j \hat{V}_{k-1}(x_j) p(x_j|x_i,a_2) \\ r(x_i,a_3) + \sum_j \hat{V}_{k-1}(x_j) p(x_j|x_i,a_3) \end{bmatrix} \tag{5}$$

Iteration 1:

$$\hat{V}_1(x_0) = \gamma \max \begin{bmatrix} 0 + \hat{V}_0(x_1) \cdot p(x_1|x_0, a_1) \\ 0 + \hat{V}_0(x_0) \cdot p(x_0|x_0, a_2) + \hat{V}_0(x_2) \cdot p(x_2|x_0, a_2) \\ 0 + \hat{V}_0(x_0) \cdot p(x_0|x_0, a_3) + \hat{V}_0(x_4) \cdot p(x_4|x_0, a_3) \end{bmatrix} = \begin{bmatrix} 0 + 0 \cdot 1 \\ 0 + 0 \cdot 0.5 + 0 \cdot 0.5 \\ 0 + 0 \cdot 0.2 + 0 \cdot 0.8 \end{bmatrix} \tag{6}$$

$$= \gamma \cdot 0 \tag{7}$$

$$\hat{V}_1(x_1) = \gamma \max \begin{bmatrix} R_s + 0 \\ R_s + 0 \\ R_s + 0 \end{bmatrix} \tag{8}$$

$$= \gamma \cdot R_s \tag{9}$$

$$\hat{V}_1(x_2) = \gamma \max \begin{bmatrix} 0 + \hat{V}_0(x_3) \cdot p(x_3|x_2, a_1) \\ 0 + \hat{V}_0(x_2) \cdot p(x_2|x_2, a_2) + \hat{V}_0(x_c) \cdot p(x_c|x_2, a_2) \\ 0 + \hat{V}_0(x_2) \cdot p(x_2|x_2, a_3) + \hat{V}_0(x_6) \cdot p(x_6|x_2, a_3) \end{bmatrix} = \begin{bmatrix} 0 + 0 \cdot 1 \\ 0 + 0 \cdot 0.5 + 0 \cdot 0.5 \\ 0 + 0 \cdot 0.2 + 0 \cdot 0.8 \end{bmatrix} \tag{10}$$

$$= \gamma \cdot 0 \tag{11}$$

$$\hat{V}_1(x_3) = \gamma \max \begin{bmatrix} R_L + 0 \\ R_L + 0 \\ R_L + 0 \end{bmatrix} \tag{12}$$

$$= \gamma \cdot R_L \tag{13}$$

$$\hat{V}_1(x_4) = \gamma \max \begin{bmatrix} 0 + \hat{V}_0(x_5) \cdot p(x_5|x_4, a_1) \\ 0 + \hat{V}_0(x_4) \cdot p(x_4|x_4, a_2) + \hat{V}_0(x_6) \cdot p(x_6|x_4, a_2) \\ 0 + \hat{V}_0(x_4) \cdot p(x_4|x_4, a_3) + \hat{V}_0(x_c) \cdot p(x_c|x_4, a_3) \end{bmatrix} = \begin{bmatrix} 0 + 0 \cdot 1 \\ 0 + 0 \cdot 0.5 + 0 \cdot 0.5 \\ 0 + 0 \cdot 0.2 + 0 \cdot 0.8 \end{bmatrix} \tag{14}$$

$$= \gamma \cdot 0 \tag{15}$$

$$\hat{V}_1(x_5) = \gamma \max \begin{bmatrix} R_R + 0 \\ R_R + 0 \\ R_R + 0 \end{bmatrix} \tag{16}$$

$$= \gamma \cdot R_R \tag{17}$$

$$\hat{V}_1(x_6) = \gamma \max \begin{bmatrix} R_O + 0 \\ R_O + 0 \\ R_O + 0 \end{bmatrix} \tag{18}$$

$$= \gamma \cdot R_O \tag{19}$$

$$\hat{V}_1(x_c) = \gamma \max \begin{bmatrix} -R_C + 0 \\ -R_C + 0 \\ -R_C + 0 \end{bmatrix} \tag{20}$$

$$= -\gamma \cdot R_C \tag{21}$$

So our approximate value estimate is:

$$\hat{V}_1 = \begin{bmatrix} 0 & \gamma R_S & 0 & \gamma R_L & 0 & \gamma R_R & \gamma R_O & -\gamma R_C \end{bmatrix} \tag{22}$$

Iteration 2:

$$\hat{V}_2(x_0) = \gamma \max \begin{bmatrix} 0 + \hat{V}_1(x_1) \cdot p(x_1|x_0, a_1) \\ 0 + \hat{V}_1(x_0) \cdot p(x_0|x_0, a_2) + \hat{V}_1(x_2) \cdot p(x_2|x_0, a_2) \\ 0 + \hat{V}_1(x_0) \cdot p(x_0|x_0, a_3) + \hat{V}_1(x_4) \cdot p(x_4|x_0, a_3) \end{bmatrix} = \begin{bmatrix} 0 + \gamma R_S \cdot 1 \\ 0 + 0 \cdot 0.5 + 0 \cdot 0.5 \\ 0 + 0 \cdot 0.2 + 0 \cdot 0.8 \end{bmatrix} \tag{23}$$

$$= \gamma^2 R_S \tag{24}$$

$$\hat{V}_2(x_1) = \gamma \cdot R_s \tag{25}$$

$$\hat{V}_2(x_2) = \gamma \max \begin{bmatrix} 0 + \hat{V}_1(x_3) \cdot p(x_3|x_2, a_1) \\ 0 + \hat{V}_1(x_2) \cdot p(x_2|x_2, a_2) + \hat{V}_1(x_c) \cdot p(x_c|x_2, a_2) \\ 0 + \hat{V}_1(x_2) \cdot p(x_2|x_2, a_3) + \hat{V}_1(x_6) \cdot p(x_6|x_2, a_3) \end{bmatrix} = \begin{bmatrix} 0 + \gamma R_L \cdot 1 \\ 0 + 0 \cdot 0.5 - \gamma R_C \cdot 0.5 \\ 0 + 0 \cdot 0.2 + \gamma R_O \cdot 0.8 \end{bmatrix} \tag{26}$$

$$= \gamma^2 R_L \text{ or } 0.8\gamma^2 R_O \tag{27}$$

$$\hat{V}_2(x_3) = \gamma \cdot R_L \tag{28}$$

$$\hat{V}_2(x_4) = \gamma \max \begin{bmatrix} 0 + \hat{V}_1(x_5) \cdot p(x_5|x_4, a_1) \\ 0 + \hat{V}_1(x_4) \cdot p(x_4|x_4, a_2) + \hat{V}_1(x_6) \cdot p(x_6|x_4, a_2) \\ 0 + \hat{V}_1(x_4) \cdot p(x_4|x_4, a_3) + \hat{V}_1(x_c) \cdot p(x_c|x_4, a_3) \end{bmatrix} = \begin{bmatrix} 0 + \gamma R_R \cdot 1 \\ 0 + 0 \cdot 0.5 + \gamma R_O \cdot 0.5 \\ 0 + 0 \cdot 0.2 - \gamma R_C \cdot 0.8 \end{bmatrix} \tag{29}$$

$$= \gamma^2 R_L \text{ or } 0.5\gamma^2 R_O \tag{30}$$

$$\hat{V}_2(x_5) = \gamma \cdot R_R \tag{31}$$

$$\hat{V}_2(x_6) = \gamma \cdot R_O \tag{32}$$

$$\hat{V}_2(x_c) = -\gamma \cdot R_C \tag{33}$$

At this iteration, we start to see the impact of our design choices for our payoff (or reward) function as noted in blue. If we assume that $R_O >> R_L > R_R$, we find the updated value estimate to be:

$$\hat{V}_2 = \begin{bmatrix} \gamma^2 R_S & \gamma R_S & 0.8\gamma^2 R_O & \gamma R_L & 0.5\gamma^2 R_O & \gamma R_R & \gamma R_O & -\gamma R_C \end{bmatrix} \tag{34}$$

You may now note that if we were to execute another iteration, we would have to more carefully look at the relative values (or assign values) of our payoff parameters given that we now have nonzero entries for all values. I encourage you to execute a few more iterations and determine the corresponding optimal policy. If you play around with different values in the payoff function, you should observe that different optimal policies will arise!

NOTE 1: I did this by hand but, as you can see, it can get pretty difficult to keep track of everything very quickly. A quick implementation is quite easy! (PS This will be an assignment on your next MP.)

NOTE 2: I follow formulation from the Probabilistic Robotics textbook. Other formulations may nest the discount factor differently. Does this affect the optimal policy?