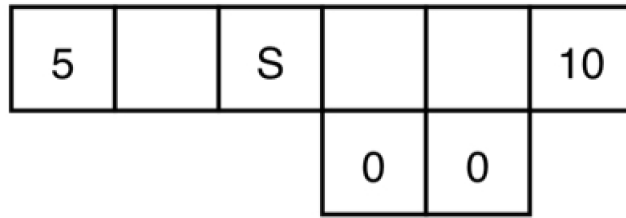


## 1 Odds and Ends

(a) Can all MDPs be solved using expectimax search? Justify your answer.

(b) Why might Q-learning be superior to TD learning of values?

## Q2. MDPs and RL

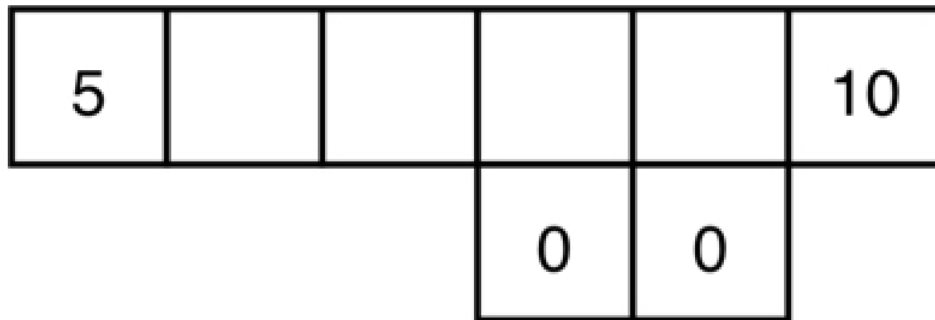


Consider the above gridworld. An agent is currently on grid cell  $S$ , and would like to collect the rewards that lie on both sides of it. If the agent is on a numbered square, its only available action is to Exit, and when it exits it gets reward equal to the number on the square. On any other (non-numbered) square, its available actions are to move East and West. Note that North and South are never available actions.

If the agent is in a square with an adjacent square downward, it does not always move successfully: when the agent is in one of these squares and takes a move action, it will only succeed with probability  $p$ . With probability  $1 - p$ , the move action will fail and the agent will instead move downwards. If the agent is not in a square with an adjacent space below, it will always move successfully.

For parts (a) and (b), we are using discount factor  $\gamma \in [0, 1]$ .

- (a) Consider the policy  $\pi_{\text{East}}$ , which is to always move East (right) when possible, and to Exit when that is the only available action. For each non-numbered state  $x$  in the diagram below, fill in  $V^{\pi_{\text{East}}}(x)$  in terms of  $\gamma$  and  $p$ .



- (b) Consider the policy  $\pi_{\text{West}}$ , which is to always move West (left) when possible, and to Exit when that is the only available action. For each non-numbered state  $x$  in the diagram below, fill in  $V^{\pi_{\text{West}}}(x)$  in terms of  $\gamma$  and  $p$ .

5					10
			0	0	

(c) For what range of values of  $p$  *in terms of*  $\gamma$  is it optimal for the agent to go West (left) from the start state ( $S$ )?

Range: \_\_\_\_\_

(d) For what range of values of  $p$  *in terms of*  $\gamma$  is  $\pi_{\text{West}}$  the optimal policy?

Range: \_\_\_\_\_

(e) For what range of values of  $p$  *in terms of*  $\gamma$  is  $\pi_{\text{East}}$  the optimal policy?

Range: \_\_\_\_\_

Recall that in approximate Q-learning, the Q-value is a weighted sum of features:  $Q(s, a) = \sum_i w_i f_i(s, a)$ . To derive a weight update equation, we first defined the loss function  $L_2 = \frac{1}{2}(y - \sum_k w_k f_k(x))^2$  and found  $dL_2/dw_m = -(y - \sum_k w_k f_k(x))f_m(x)$ . Our label  $y$  in this set up is  $r + \gamma \max_a Q(s', a')$ . Putting this all together, we derived the gradient descent update rule for  $w_m$  as  $w_m \leftarrow w_m + \alpha (r + \gamma \max_a Q(s', a') - Q(s, a)) f_m(s, a)$ .

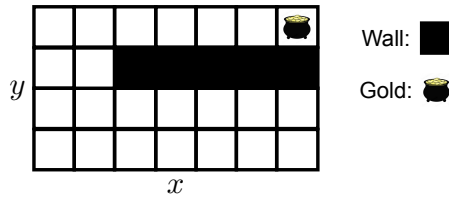
In the following question, you will derive the gradient descent update rule for  $w_m$  using a different loss function:

$$L_1 = \left| y - \sum_k w_k f_k(x) \right|$$

- (f) Find  $dL_1/dw_m$ . Show work to have a chance at receiving partial credit. Ignore the non-differentiable point.

- (g) Write the gradient descent update rule for  $w_m$ , using the  $L_1$  loss function.

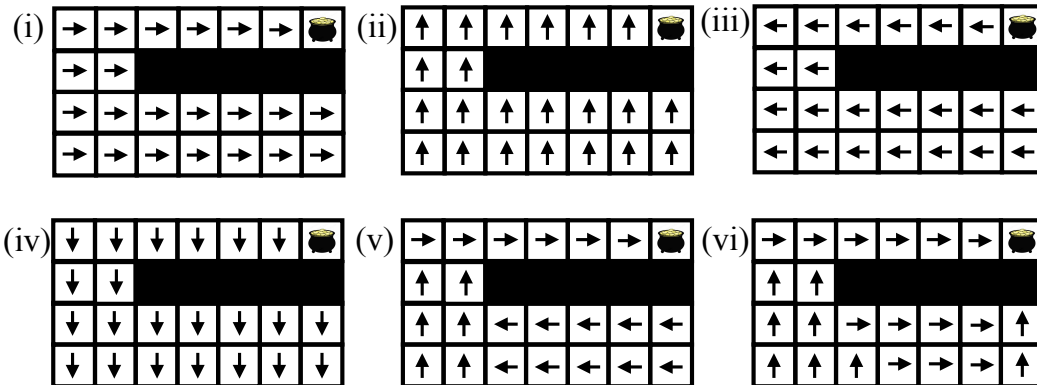
### Q3. MDPs & RL



Consider the grid-world MDP above. The goal of the game is to reach the pot of gold. As soon as you land on the pot of gold you receive a reward and the game ends. Your agent can move around the grid by taking the following actions: North, South, East, West. Moving into a square that is not a wall is always successful. If you attempt to move into a grid location occupied by a wall or attempt to move off the board, you remain in your current grid location.

Our goal is to build a value function that assigns values to each grid location, but instead of keeping track of a separate number for each location, we are going to use features. Specifically, suppose we represent the value of state  $(x, y)$  (a grid location) as  $V(x, y) = w^T f(x, y)$ . Here,  $f(x, y)$  is a feature function that maps the grid location  $(x, y)$  to a vector of features and  $w$  is a weight vector that parameterizes our value function (note that entries in  $w$  can be any real number, positive or negative).

In the next few questions, we will look at various possible feature functions  $f(x, y)$ . We will think about the value functions that are representable using each set of features, and, further, think about which policies could be extracted from those value functions. Assume that when a policy is extracted from a value function, ties can be broken arbitrarily. In our definition of feature functions we will make use of the location of the pot of gold. Let the gold's location be  $(x^*, y^*)$ . Keep in mind the policies (i), (ii), (iii), (iv), (v), and (vi) shown below.



(a) Suppose we use a single feature: the x-distance to the pot of gold. Specifically, suppose  $f(x, y) = |x - x^*|$ . Which of the policies could be extracted from a value function that is representable using this feature function? Assume the weights vector  $w$  is not allowed to be 0. Fill in all that apply.

- (i)     
  (ii)     
  (iii)     
  (iv)     
  (v)     
  (vi)

(b) Suppose we use a single feature: the y-distance to the pot of gold. Specifically, suppose  $f(x, y) = |y - y^*|$ . Which of the policies could be extracted from a value function that is representable using this feature function? Assume the weights vector  $w$  is not allowed to be 0. Fill in all that apply.

- (i)       (ii)       (iii)       (iv)       (v)       (vi)

(c) Suppose we use a single feature: the Manhattan distance to the pot of gold. Specifically, suppose  $f(x, y) = |x - x^*| + |y - y^*|$ . Which of the policies could be extracted from a value function that is representable using this feature function? Assume the weights vector  $w$  is not allowed to be 0. Fill in all that apply.

- (i)       (ii)       (iii)       (iv)       (v)       (vi)

(d) Suppose we use a single feature: the length of the shortest path to the pot of gold. Which of the policies could be extracted from a value function that is representable using this feature function? Assume the weights vector  $w$  is not allowed to be 0. Fill in all that apply.

- (i)       (ii)       (iii)       (iv)       (v)       (vi)

(e) Suppose we use two features: the x-distance to the pot of gold and the y-distance to the pot of gold. Specifically, suppose  $f(x, y) = (|x - x^*|, |y - y^*|)$ . Which of the policies could be extracted from a value function that is representable using this feature function? Assume the weights vector  $w$  must have at least one non-zero entry. Fill in all that apply.

- (i)       (ii)       (iii)       (iv)       (v)       (vi)