CS188 Spring 2011 Section 4: Minimax and MDPs

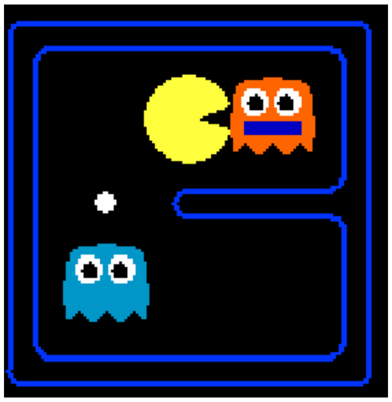
1 Suicidal Pacman

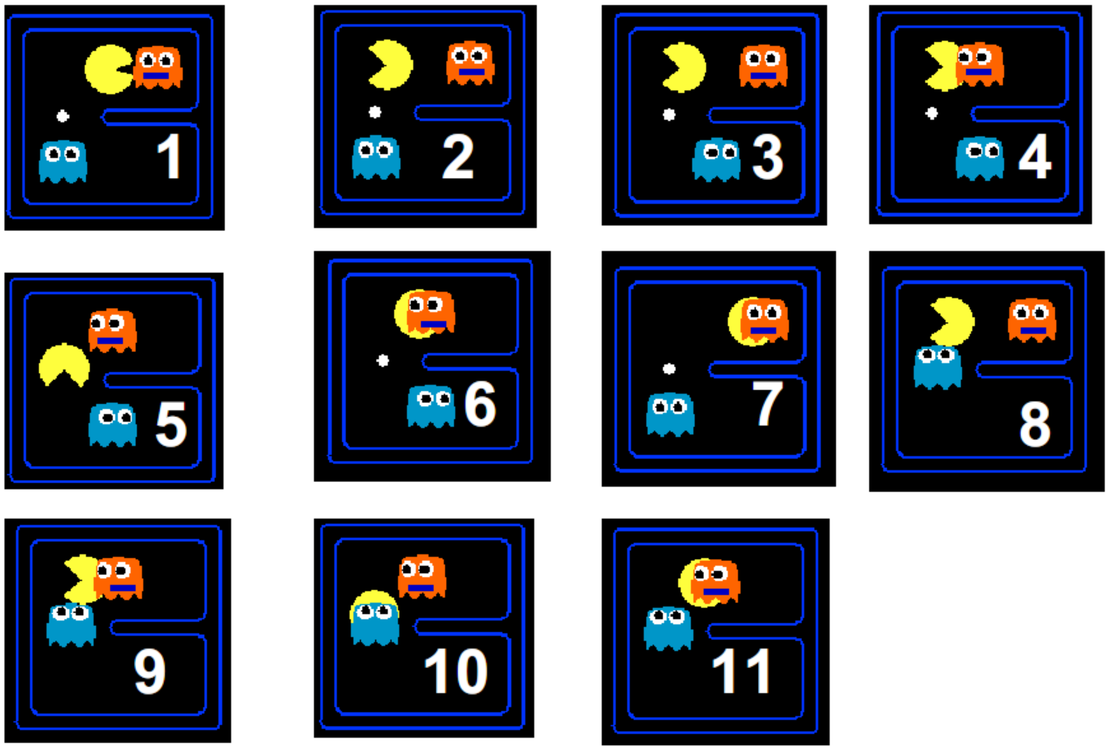
Pacman is sometimes suicidal when doing a minimax search because of its worst case analysis. We will build here a small expectimax tree to see the difference in behavior.

Consider the following rules:

- Ghosts cannote change direction unless they are facing a wall. The possible actions are east, west, south, and north (not stop). Initially, they have no direction and can move to any adjacent square.
- We use random ghosts which choose uniformly between all their legal moves.
- Assume that Pacman cannot stop
- If Pacman runs into a space with a ghost, it dies before having the chance to eat any food which was there.
- The game is scored as follows:
  - -1 for each action Pacman takes
  - 10 for each food dot eaten
  - -500 for losing (if Pacman is eaten)
  - 500 for winning (all food dots eaten)

Given the following "trapped" maze, build the expectimax tree with max and chance nodes clearly identified. Use the game score as the evaluation function at the leaves. If you don't want to make little drawings, all possible states of the game have been labeled for you on the next page: use them to identify the states of the game. Pacman moves first, followed by the lower left ghost, then the top right ghost.





(a) Build the expectimax tree. What is Pacman's optimal move?

(b) What would pacman do if it was using minimax instead?

(c) By changing the probabilities of action for the ghosts, can you get expectimax to make the same decision as minimax?

(d) Now say you are using the following alternate game score components:

- -1 for Pacman making a move
- -1.5 for losing
- 0 for eating food
- 0.3 for winning

Use this new game score as your evaluation function at the leaves. Note this yields a monotonic transformation of the original utilities: a function which preserves the ordering of the state according to their utility. Could this change the decision of Pacman using expectimax?

2 High-Low as an MDP

The game High-Low is a card game played with an infinite deck containing three types of cards: 2,3,4. You start with a 3 showing, and say either *higher* or *lower*. Then, a new card is flipped; if you say *higher* and the new card is higher in value than your current card, you win the points shown on the new card. Similarly, if you say *lower* and the new card is lower in value than your current card, you win the points shown on the new card. If the new card is the same value as your current card, you don't get any points. Otherwise, the game ends. Your current card is then discarded and the new card becomes your current card. An example of a game is $[3, high, 4, low, 2, high, 3, low, 4, end]$

The deck contains different proportions of 2,3, and 4 cards, p_2, p_3, p_4 respectively (where $p_2 + p_3 + p_4 = 1$), which you may or may not know.

(a) Assuming you know p_2, p_3, p_4, formulate High-Low as an MDP:

States:

Actions:

Rewards:

R(s,a,s')	s'		
(s,a)	2	3	4
(2,low)			
(2,high)			
(3,low)			
(3,high)			
(4,low)			
(4,high)			

Transitions:

T(s,a,s')	s'			
(s,a)	2	3	4	end
(2,low)				
(2,high)				
(3,low)				
(3,high)				
(4,low)				
(4,high)				

(b) Write down non-trivial Bellman equations for $Q^*(3, high)$, the utility of saying *high* after seeing 3, $Q^*(3, low)$, the utility of saying *low* after seeing 3, and $V^*(3)$, the value of seeing 3 under the optimal policy. Assume a discount factor $\gamma = 1$. You may use the variables $p_2, p_3, p_4, Q^*(2, a), Q^*(3, a), Q^*(4, a)$ in your answer for $Q^*(3, a)$ and $p_2, p_3, p_4, V^*(2), V^*(3), V^*(4)$ in your answer for $V^*(3)$.

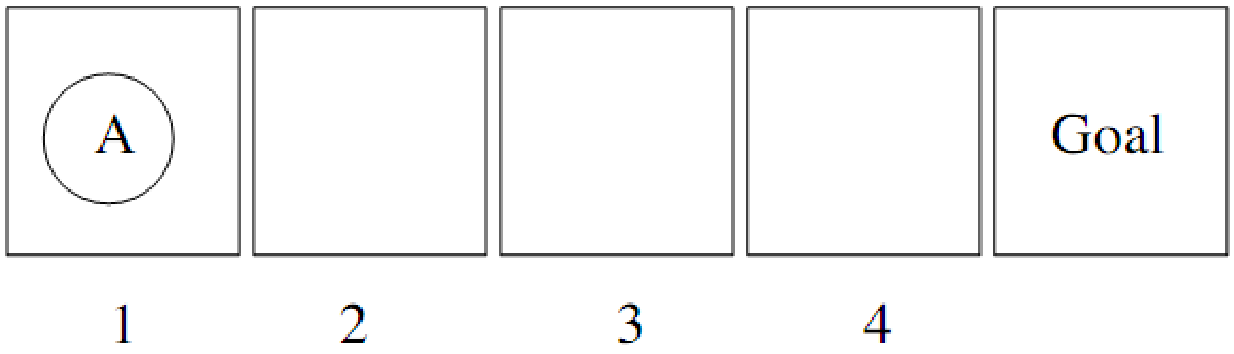
$Q^*(3, high) =$

$Q^*(3, low) =$

$V^*(3) =$

3 Soccer

A soccer robot A is on a fast break toward the goal, starting in position 1. From positions 1 through 3, it can either shoot (S) or dribble the ball forward (D). From 4 it can only shoot. If it shoots, it either scores a goal (state G) or misses (state M). If it dribbles, it either advances a square or loses the ball, ending up in M. When shooting, the robot is more likely to score a goal from states closer to the goal; when dribbling, the likelihood of missing is independent of the current state.



In this MDP, the states are 1,2,3,4,G and M, where G and M are terminal states. The transition model depends on the parameter y, which is the probability of dribbling success. Assume a discount of $\gamma = 1$.

$$\begin{aligned} T(k,S,G) &= \frac{k}{6} \\ T(k,S,M) &= 1-\frac{k}{6} \\ T(k,D,k+1) &= y \quad \text{for } k \in \{1,2,3\} \\ T(k,D,M) &= 1-y \quad \text{for } k \in \{1,2,3\} \\ R(k,S,G) &= 1 \end{aligned}$$

Rewards are 0 for all other transitions.

(a) What is $V^{\pi}(1)$ for the policy π that always shoots?

(b) What is $Q^*(3,D)$ in terms of y?

(c) Using $y = \frac{3}{4}$, complete the first two iterations of value iteration.

i	$Q_i(1,S)$	$Q_i(2,S)$	$Q_i(3,S)$	$Q_i(4,S)$
0	0	0	0	0
1				
2				

i	$Q_i(1,D)$	$Q_i(2,D)$	$Q_i(3,D)$
0	0	0	0
1			
2			

i	$V_i^*(1)$	$V_i^*(2)$	$V_i^*(3)$	$V_i^*(4)$
0	0	0	0	0
1				
2				

(d) After how many iterations will value iteration compute the optimal values for all states?

(e) For what range of values of y is $Q^*(3,S) \geq Q^*(3,D)$?