6.390 Introduction to Machine Learning

Recitation Week #12

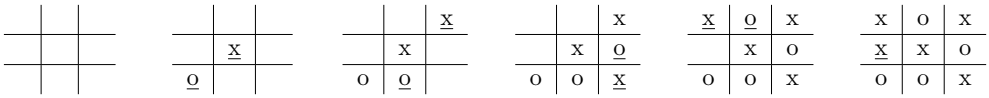
Issued November 28, 2022

1. Tic-tic-tic is a game for two players who take turns marking the spaces in a 3×3 grid. One player uses an 'X' for their mark and the other uses an 'O'. For both players, the object of the game is to place three of their marks sequentially in a row–either horizontally, vertically, or diagonally.

   Suppose that we are the player using the X marks, and that the O player is a (possibly stochastic) algorithm. We do not know the strategy or reward function that is being used by the algorithm. The initial state of the board is empty and we make the first move. We can select *any* of the nine squares on our turn.

   If there are any remaining squares, it is our turn to play. If we select an occupied square that already has an X or an O in it, reward is zero, we remain in the same state and it's still our turn. When we place an X in an empty square, then an O appears in some other empty square, if one is available. When we place an X in the final empty square, the O player will return the full board to us, and the game will end if the player with the X marks takes action "end game" from this state. Also, if the X player chooses "end game" when the empty spaces are not yet all filled in, the reward is zero, we remain in the same state, and it's still our turn.

   Note, unlike in tic-tac-toe, that the game does not end once either player places three of their marks in a row. Instead, it ends when the X-player explicitly ends the game from a full board. The diagram shows a full game of tic-tic-tic from our perspective with the X marks (the underlined marks denotes the play from both players on the current turn):



   (a) First, we need to decide how to represent the state space for tic-tic-tic.

      i. Jody suggests letting the state space be all possible 3×3 grids in which each square contains one of the following: a space, an O, or an X. Is this a valid state space representation? Is it a good state space representation? For each question, why or why not?

ii. Dana suggests using all possible 3×3 grids of X's, O's, and empty spaces such that either: (a) the number of O's and the number of X's are equal or (b) there are 5 X's and 4 O's. Is Dana's suggestion better or worse for tabular Q-learning than Jody's? Is it a good state space representation? For each question, why or why not?

(b) What is a good choice of action space for tic-tic-tic?

(c) Suppose we would like a reward function for tic-tic-tic that, at the end of the game, gives $+1$ reward to the X player for every three-in-a-row of X's and -1 reward to the X player for every three-in-a-row of O's. Sketch out the reward function. What are the possible rewards for a single game?

(d) Using Dana's state space, your action space from (b) and reward function from (c), do we have a complete description of an MDP? If yes, write out the MDP; if not, elaborate on what might be missing.

(e) Your friend Exo is really good at tic-tic-tic and you'd like to learn from them. You sit down and watch them play this game for a long time, and you observe their sequence of state-action pairs and their rewards. Which of the following machine-learning problem formulations is most appropriate, for you to learn from your friend? Would you use all their games for this learning purpose?

1. supervised regression (describe the loss function)
2. supervised classification (describe the loss function)
3. reinforcement learning of a policy

Explain your answer.

(f) Barney wants to solve a tic-tic-tic problem that is exactly the same as the above game (i.e., the final score is the number of unique three-in-a-row/column/diagonal X's minus the number of unique three-in-a-row/column/diagonal O's), except that it is played on a 101×101 grid.

i. Is it better for Barney to use tabular Q-learning or neural-net Q-learning? Explain.

ii. Suppose Barney were to use neural-net Q-learning; would it help for him to start with a convolutional layer? If your answer is yes, describe four 3×3 convolutional filters that would be particularly helpful for this problem.

(g) As specified in part (c), is the 3×3 tic-tic-tic game guaranteed to end? Why or why not?

(h) After attempting to apply Q-learning to the 3×3 tic-tic-tic problem for some time, you decide that you'd like to find a way for the agent to learn on its own not to attempt to place an X in an occupied square. Propose a way to modify the problem description and reward function to promote this behavior.

(i) Suppose you apply Q-learning with discount factor $\gamma = 1$ to the 3×3 tic-tic-tic problem now with modifications from part (h), and your actions can either select an unfilled square or a filled square, per the original assumptions. Is Q-learning guaranteed to result in a winning strategy?