Machine Learning Course (Spring 2024)

Homework II

Function Approximate

Assignment Date: **1 Farvardin** Submission Deadline: **15 Farvardin**

Contents

Objectives and Precautions

In this homework you will learn:

- implement basic AI in Tic Tac Toe Game
- Basic learning methods : Function Approximate

Also note that:

- Studying the Chapter 1 of Tom Michael's book is highly recommended before reading and implementing this assignment. In the first chapter of Tom Mitchel's book, the learning problem is reduced to function approximation problem and one specific class of learning Task is chosen to explain in details.
- Feel free to use your preferred programming languages, but we suggest using python.(You can't Use any library for Implementation).
- Provide a report for your Home Work and explain your features, code, and results. Your submission then will be a zip file containing your code files and your report. The name of the uploading file should be your Lastname-Firstname.
- Pay extra attention to the due date. It will not be extended,Be advised that submissions after the deadline would not grade.
- Feel free to ask your questions about Home Work.
- Using other students' codes or the codes available on the internet will lead to zero grades.

Good Luck.

1 Function Approximation (Tic-Tac-Toe Game)

In this assignment you should use the Mitchel's described approach to implement an algorithm to learn tic-tac-toe game. As you know, such tasks abstract as a graph that every board state is a node and every move is an edge. The goal is to find the best move for each state.

in this assignment, you should implement an algorithm to learn a specific edition of the tic-tac-toe game. As you know, such tasks abstract as a graph that every board state is a node and every move is an edge. The goal is to find the best move for each state. We can define a function V: S$\rightarrow$ R that assign to every state a value and based on V find the best move for each state. If we are able to find V the problem is solved but only few pairs $\langle$s, V(s)$\rangle$is available.There are many states that don't have the value of V Therefore, we should approximate this function only by these points (training examples). Because of the graph structure of the problem, this is possible to approximate the value of unknown-value states and one can use approximated values to find better approximations. So for unknown-value states, you should use approximated successor state's value.

1.1 Implement an algorithm, which learn to play tic-tac-toe (3$\times$ 3 board) with function approximation approach.

These steps will help you:

- choose linear representation for $\hat{V}$: $\hat{V}(\bar{x}) = \bar{w}^T \bar{x}$
- Find some good features($\bar{x}$) to describe the states of Tic-tac-toe game.
- Initialize all coefficients ($\bar{x}$) in $\hat{V}$ by zero.
- Start with empty board state and in each non-terminal state (s_t) choose your move based on $\hat{V}$ as follows:
  Among the moves someone can do in the current state, choose the one which leads to highest state value (value of successors of current state). Call these chosen successor state s_{t+1} and assign$V_{train}(s_t)$= $\hat{V}_{train}(S_{t+1})$. If (s_t) is a terminal state assign $\hat{V}_{train} = V(s)$. set value of the final state as follows:

$$V_{train}(s_t) = 100 \text{ If a player wins}$$
$$V_{train}(s_t) = -100 \text{ If a player loses}$$
$$V_{train}(s_t) = 0 \text{ If a player draws}$$

- Now you have new example: $\langle S_t , V_{train}(s_t) \rangle$.Based on this example update $\bar{w}$as follows:

$$w_i = w_i + \alpha(V_{train}(s_t) - \hat{V}(s_t))x_i$$

Assign a small value for α.Once you train the algorithm and find the weights (w_is),use these weights to run two agents to play with each other. (Save these codes in a separate file).

1.2 Based on the previous part, write an agent code that you can play with it. (Save this in a separate file).

1.3 Optional Part (Bonus).

- Provide a graphical playing game for your implementation.