Coding Project (Graduate)

Application: Working with Artificial Intelligence (AI) / Business Intelligence (BI) Applications

Reasoning AI systems can analyze complex data, connect patterns, and draw conclusions. They simulate human-like reasoning and decision-making processes. Autonomous vehicles are a prime example, utilizing reasoning AI for safe navigation.

By now, you should be comfortable with programming in Java. This project will be to enhance an already existing program to ensure a proper reasoning AI/BI system. For this you will consider the incorporation of game matrix with the corresponding payoffs that was assigned to your specific group in your proposal. Do not forget that based on your proposal, you should consider six marketing strategies in which three strategies should be considered for one player, and the other three strategies should be considered for the second player.

As I mentioned, you will be working with an existing program, and you will try to modify the program to produce results based on the assigned game matrix with the corresponding payoffs in your assigned proposal instructions. You cannot publish the program under your name even after modification without proper accreditation. You cannot claim the program entirely. You can only claim the parts that you will modify that is accompanied with proper accreditation.

The Retrieved Source's URL is (2019):

https://github.com/Axel-Jacobsen/GeneticPrisonersDilemma/tree/master

Important ****

In this application, the following strategy files have been programmed, using boolean values of False and True. The ‘True’ bollean value corresponds to ‘AlwaysCooperate’; and the ‘False’ Boolean value corresponds to ‘AlwaysDefect’. This enables the players to recognize which strategy has been taken. In this program, no strategy has been associated to specific player. That means each player can play the same strategy that other plays. However, since, you should consider 3 strategies for one player and 3 other strategies for the other player, you have to make sure that only respective strategies are played by the corresponding players.

AlwaysCooperate

AlwaysDefect

GeneticMemory

GeneticOneMove

GeneticStrategy

Grudger

Simpleton

Strategy

TitForTat

TitForTatTat

Important ****

Now, you should consider your six marketing strategies. Assuming, you have considered your six marketing strategies, you should create them similar to ‘AlwaysCooperate’ and ‘AlwaysDefect’. However, you will not use ‘AlwaysCooperate’ and ‘AlwaysDefect’. Since there are 3 strategies are for one player and 3 others for the other player, I recommend you to use ‘int’ values. Do not forget that you have to make sure that only respective strategies are played by the corresponding players.

Lets say that you considered the following strategies. However, you need to specify marketing strategies. As an example, one marketing strategy is ‘AlwaysLowPrice’.

AlwaysX → for player 1

AlwaysY → for player 1

AlwaysZ → for player 1

AlwaysG → for player 2

AlwaysM → for player 2

AlwaysW → for player 2

Modify the following strategies, based on the six strategies above:

TitForTat

GeneticOneMove

For GeneticOneMove, if we look at the current state of coding, but with three ranges, let's say;

1. If random generator is between 0.0 and 0.35 and the weight is within this range, we go with the strategy that is the least profitable.

2. If random generator is between 0.36 and 0.7; the weight is within this range; the random generator is less than weight, we go with the least profitable(we do not change strategy stated in point 1). However, if the random generator is more than weight, we go with a strategy that is a little more profitable.

3. If random generator is between 0.71 and 1.0; the weight is within this range; the random generator is less than weight, we do not change strategy stated in point 2. However, if the random generator is more than weight, we go with a strategy that is a most profitable.

This is similar to prisoners' dilemma move in GeneticOneMove for if the random generator is less than the weight, the we cooperate; otherwise defect.

Extremely Important for coming up with the right values for choosing strategies

You need to get rid of random generator in determining strategies. You need to use normalization method for the values between 0.0 and 1.0 to come up with proper scores in determining strategies. If you do not understand this section or you need some clarifications, you need to come to my office. Wrong calculations will result in losing grades.

$$x_{\text{normalized}} = (x - x_{\text{minimum}}) / (x_{\text{maximum}} - x_{\text{minimum}})$$

Note,

In your project, you need to show each individual calculations for determining the strategies. Show you calculations in the 'GeneticOneMove' class through a commented section.

Once completed, you need to make the presentation on how you programmed and what the results are for the six marketing strategies, if players play along with TitForTat and GeneticOneMove.

Submit your enhanced program in a zipped file along with your final presentation before last day of the term and before coming to class. You will present this on the last day.