MARS Soccer Simulation 2D Team Description Paper 2024

Mehdi torshani [1], Mohammad Maziani [2], Mohammad Ghasemi [3], Mohammad Bonakdar Tehrani [4], Fakhteh Hosseinkhani [5], Mahdieh GhasemPour [6], Afsaneh Imani [7], Zahra Samadi [8] ,and Mercedeh Ghasemi[9]

Arman Afzar Gostar Company, Center for Research and Development of New Technologies research@armanafzar.com Mehdi.torshani@gmail.com
Mars Team (armanafzar.com)
Dr.Amir Bahram Peyman,Dr.Tolou Shadlou

Abstract. Report examines the records and performance of the Mars team in the two-dimensional football simulation game.The Mars team began its activities in robotics in 2006. This team has developed various algorithms in the ation league using the Agent2D base code. In this report, we will discuss part of this team's goals in decision-making methods and the player learning system. The algorithm discussed in this report is a decision tree that makes the best decision for each player using the highest probability and the best training data.
Keywords: RoboCup · Soccer Simulation 2D · Behavior Predictor.

1. Introduction

In this report, we will examine the decision-making structure of the Mars team. With the importance of correct decision-making in two-dimensional simulations, the use of machine learning-based methods has a significant impact on improving the decision-making structure of players. The Mars team has implemented and tested various methods to improve this process for years. Among these methods, a structure based on neural networks can be mentioned, which this team has used in various leagues for many years using this algorithm. Other methods like the CBR algorithm have also been used by this team. However, one of the significant concerns that have always been problematic in these methods is the speed of learning training examples and the ability to classify input data based on training information. This report presents a method that trains the decision-making structure in the fastest way and enables the classification of input data for players.

2. Decision Tree

One of the common methods for decision-making in Multi-Agent systems is the use of a decision tree. An essential feature of the decision tree is the trainable structure of this algorithm. Training examples for this tree can have a significant impact on data classification. For efficient and rapid training of the tree, it is crucial that the features selected as the main root of the tree are the best choice. We will continue to examine the methods of choosing roots in the decision-making structure of one of the players' capabilities.

2.1. Selection of Training Features

In the decision-making structure, the selected features for training the algorithm vary for each player's capability. With this presence, one of the capabilities was chosen as an example to better examine this algorithm. Features are selected as a probabilistic combination of events that affect each capability. For example, we will examine the features relevant to passing. {distance, path, isgoalkeeper, ispenaltyarea, ...}

2.2. Main Root Selection

For selecting the main root in the tree, the most critical feature is the calculation of information gain. For this purpose, we examine the impact of each feature on increasing the likelihood of establishing a decision tree. In each stage of tree growth, this process is repeated for labeling the roots.

$$\text{Entropy } = -p_{+\log}(p_2) - p_{-\log}(p_2)$$

$$\text{Entropy } (s) = \sum_{i=1}^{c-p} - p\log\,(p_2)$$

$$\text{Gain } (s,a) = \text{ Entropy } (s) - \sum_{v\in vadus|a)} \frac{|s_v|}{s} \text{Entropy } (s_v)$$

Fig1: To calculate the Information Gain, we first assess the level of entropy for each feature. Entropy is determined by examining the ratio of the probability of positive events to negative events.

2.3. Classification using training data

is the most crucial process of the pr oposed algorithm. This process involves using training data to form a decision tree and pruning it to provide an appropriate algorithm for classifying input data. To enhance the decision-making process, we use a fuzzy structure instead of a traditional logical tree. This method allows us to classify a larger set of input data with fewer training and testing samples. Additionally, the training data can be improved during the execution of the algorithm.

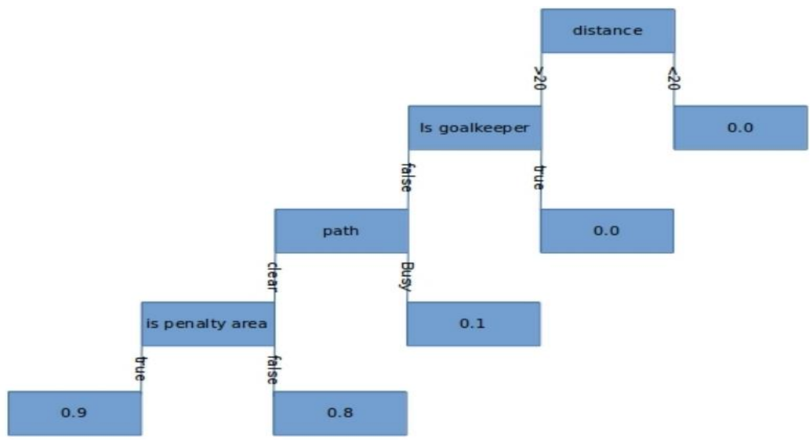

Fig2. An example of creating a decision tree from hypothetical training data.

In the decision tree that has been populated with hypothetical data, the acceptable sequences for making a decision are as follows:

- (distance < 20) ∧ (isgoalkeeper = false) ∧ (path = clear) ∧ (ispenaltyarea = true) = 0.9
- (distance < 20) ∧ (isgoalkeeper = false) ∧ (path = clear) ∧ (ispenaltyarea = false) = 0.8
- (distance < 20) ∧ (isgoalkeeper = false) ∧ (path = Busy) = 0.1

Fig3. Decision Tree Rules With the rule derived from our tree we can classify the input data. There are various methods for classifying data, and depending on our tree's output model (fuzzy), we chose the KNN method. In this method, each input data receives its weight in the decision tree according to the training data and due to the limited training data and the variety of situations that arise in the game. After weighting, the input sample is matched with similar training data using the KNN method. The combination of these two algorithms for classification is used solely for the purpose of overlapping the limited training data. Here, if the training sample matches the input data perfectly, the data is placed in its category, otherwise, it is assigned to the nearest category using KNN.

2.4. Pruning the Decision Tree

One of the main problems in decision trees is Overfitting, which we address in this section using test data in conjunction with training data. This process continues until the tree grows normally and prevents over-complication, during which irrelevant training data are removed.

3. Results

Conclusion Using the decision tree in our simulation examples, we have improved significantly in terms of speed in learning from training examples. In addition to speed improvement, another notable aspect is the high flexibility of this algorithm in unknown situations. As a result, in the experimental model, we achieved a better output than the previous decision-making method (neural network). The current algorithm is in the process of being fully implemented and collecting a complete set of training samples.

References

1. Akiyama H, Nakashima T. Helios base: An open source package for the robocup soccer 2d simulation. InRobot Soccer World Cup 2013 Jun 24 (pp. 528-535). Springer, Berlin, Heidelberg.
2. Fukushima T, Nakashima T, Akiyama H. Similarity Analysis of Action Trajectories Based on Kick Distributions. InRobot World Cup 2019 Jul 2 (pp. 58-70). Springer, Cham.
3. Akiyama, H., Nakashima, T., Fukushima, T., Suzuki, Y., Ohori, A.: Helios2019: Team description paper. In: RoboCup 2019 Symposium and Competitions, Sydney, Australia (2019)
4. Nakashima T, Akiyama H, Suzuki Y, Ohori A, Fukushima T. Helios2018: Team description paper. InRoboCup 2018 Symposium and Competitions: Team Description Papers, Montreal, Canada 2018 Jul.
5. The provided references are related to machine learning, specifically focusing on decision trees, K-Nearest Neighbors (K-NN), and other classification techniques. These sources cover foundational concepts, methodologies, and comparative analyses of different machine learning algorithms:
6. Tom Mitchel's "Machine Learning" (1997) by McGraw Hill is a seminal text that provides a comprehensive introduction to the field of machine learning, including decision trees among other algorithms.
7. "Decision Tree in Machine Learning" by Dr. Saeed Shiri is likely a source that discusses the application and theory behind decision trees in machine learning, offering insights into how these models can be used for data classification and prediction.
8. Wei-Yin Loh's work on "Classification and Regression Trees" delves into the specifics of decision trees, including their use in both classification and regression problems, highlighting the versatility of decision trees in handling various types of data. Piyush Rai's "Supervised Learning: K-Nearest Neighbors and Decision Trees" (2011) likely discusses the fundamentals of supervised learning with a focus on K-NN and decision trees, explaining how these algorithms can be applied to solve real-world problems.
9. Oliver Sutton's "Introduction to k Nearest Neighbour Classification and Condensed Nearest Neighbour Data Reduction" (2012) focuses on the KNN algorithm, detailing its classification mechanism and discussing data reduction techniques that can improve its efficiency.
10. The comparative study by Sayali D. Jadhav and H. P. Channe (2014) on "K-NN, Naive Bayes and Decision Tree Classification Techniques" likely evaluates the performance of these algorithms, offering insights into their strengths and weaknesses in different scenarios.

11. These references collectively provide a robust foundation for understanding key machine learning algorithms, particularly decision trees and KNN, and their applications in data classification and analysis