Learning to Navigate in a 3D Environment

Nurulhidayati Haji Mohd Sani[1(✉)], Somnuk Phon-Amnuaisuk[1,2], Thien Wan Au[1], and Ee Leng Tan[3]

[1] School of Computing and Informatics, Universiti Teknologi Brunei, Jalan Tungku Link, Gadong BE1410, Brunei Darussalam
nurulhidayati.sani@gmail.com

[2] Centre for Innovative Engineering, Universiti Teknologi Brunei, Jalan Tungku Link, Gadong BE1410, Brunei Darussalam

[3] Sesame World Technology, Berakas, Brunei Darussalam

Abstract. In this paper, we investigate the knowledge acquisition and the learning ability of an agent in a three-dimensional (3D) environment using data mining techniques. We apply three data mining techniques: naïve Bayes, decision tree and apriori; to a human-controlled navigation and then investigate the characteristic of knowledge discovered from each of these techniques. The results shows that the agent is able to learn to navigate automatically in the environment but with different outcomes and limitations.

Keywords: Navigation · Machine learning · Apriori · Decision tree · Naïve bayes

1 Introduction

Our work is motivated by the goal of building a machine capable of learning to automatically navigate in an unknown environment. A self-navigating machine, can be useful, especially in a hazardous situation for example. This paper investigates the knowledge acquisition and learning abilities of a goal-directed agent using data mining techniques. We model an agent that is able to automatically navigate its way in a 3D environment using knowledge discovered by mining human-controlled navigation dataset.

Initially, humans are responsible in teaching the agent on how to navigate through the environment by controlling and selecting the best action for the agent in respect to the state of the environment. The players actions that navigates the agent to the goal are recorded over 4090 runs. This forms a substantial dataset that we apply data mining (DM) techniques (naïve Bayes [1], decision tree [2], apriori [3]) and examine emerging patterns which will formulate the knowledge for the agent. This paper investigates (i) whether the agent can navigate in a new environment using knowledge learned from the human-controlled navigation dataset, and (ii) the nature of the agents performance based on these three knowledge discovery techniques.

The rest of the paper is organized as follows. In Sect. 2, we present the related works. Section 3 defines and formulates the problem and provides an overview

C. Sombattheera et al. (Eds.): MIWAI 2016, LNAI 10053, pp. 271–278, 2016.
DOI: 10.1007/978-3-319-49397-8_23

of the environment and agent. Section 4 outlines the experimental design. Our experimental results and discussions are reported in Sect. 5. Finally, we conclude our findings and propose our future work in Sect. 6.

2 Related Works

A self-navigated agent must be able to learn from their experiences; how to avoid obstacles and to find the optimal path to its destination. Many intelligent computing techniques have been investigated e.g., reinforcement learning [4], Dempster-Shafers theory of evidence [5], fuzzy logic rules [5], etc.

There are many types of techniques and approaches involved in machine learning that have been studied by researchers in the field of Artificial Intelligence (AI). Inclusively, many of these works have shown that knowledge can be learned from observed data. A previous work used history replays of Real-Time Strategy (RTS) games as the dataset to learn the gamers behavior [6]. In addition to learning behavior, Derezynski et al. [7] identified future actions as well as simulate their possible future actions and/or identified novel strategies based on the dataset obtained from logs of past games. The study [8] that built a probabilistic model that uses the historic behavior of gamers in a commercial social videogame as their dataset and provides a real-time estimation of next action expected. Although these works offer useful information to the research fields, few others modeled this knowledge into an automated agent capable of proficiency gaming. Such study is made by Weber et al. [9] where he developed an autonomous agent using a Goal-Driven Autonomy (GDA) model that can learn to perform tasks based on the demonstration in a RTS game. Similarly, Gemine et al. [10] used set of recorded games and applied supervised learning to teach an agent to learn building-production strategies. More recently, the application of Artificial Neural Network (ANN) in learning to play a Tetris game is investigated [11]. The ANN is also exploited in [12] to create an autonomous and adaptive first person shooter agent that uses reinforcement learning based on an observed environment.

One method of extracting knowledge is DM [13]. Knowledge can be described as a relationship between stimuli and response, e.g., a proposition describing antecedent and consequence. Many data mining techniques can find patterns in this shape. For example, a study is made where the state of filled and unfilled Tetris board was used as the condition to decide the actions of where to place the next tetromino in a Tetris gameplay [14]. Through several gameplays, and once enough data is collected, apriori algorithm is applied to mine stimuli and response patterns. While this technique has been shown to work extremely well in applications, the resulting predicted outcome relies entirely on the knowledge discovered based on previous selected actions performed which in turn yields a single outcome. The resulting outcome in [6,7], however, uses probabilistic models which varies based on the possibility of occurrences and provides a variety of selections which in turn might give a better performance to the application.

In this work, we consider the agent to learn to navigate based on knowledge and probabilistic approaches and to investigate the differences of these approaches in a new environment.

3 Problem Formulation

In this section, we discuss the components used to formulate the problem described. The environment used in this study is formulated first. This is followed by an explanation of the agent used to navigate the environment.

3.1 3D Virtual Environment

The environment setup for this experiment is a real world human setting modeled in a 3D virtual world representation using a Unity3D application. The environment is a virtual 3D space, populated with m game objects O_m. An object may be an agent A, a goal G, or obstacle such as wall W. The agent would navigate its way to the goal while avoiding obstacles. For every state, the agent is able to perform k actions, which in this case is a movement of one unit in any of the eight directions (N, NE, E, SE, S, SW, W, NW) in the environment but cannot pass through obstacles. The game ends when the agent reaches the goal.

3.2 Agent

The agent is modeled similarly to a human, that can perceive its environment through sensors and react accordingly through actions [15]. In this experiment, the agent is assumed to be able to: (i) analyse the environment through perceptors; (ii) acquire knowledge through learning (iii) perform actions based on the agent's knowledge.

4 Experimental Design

Two experiments will be carried out for this study as outlined in the systems architecture (see Fig. 1). In the first experiment, we (human) manually choose the actions and control the agent based on the current state of the environment. During this experiment, data is being collected for analysis. The collected data is analysed using DM techniques to obtain the knowledge for the agent. The

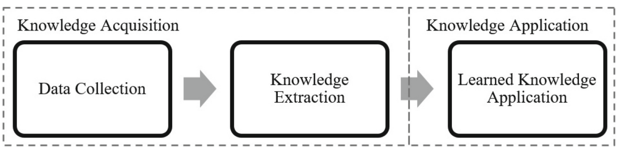
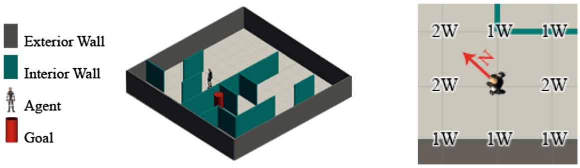


Fig. 1. Learning to navigate in a 3D environment system architecture



(a) (b)

Fig. 2. (a) The outline of the environment: An environment, an agent, a goal and obstacles (exterior and interior walls); (b) Recording the environment as a data, taken from the agents north direction. The data reads: 2W, 1W, 1W, 2W, 1W, 1W, 1W, 2W;

result based on the first experiment should provide the basic knowledge for the agent. In the second experiment, the agent applies the knowledge learned through interacting with the environment. Figure 2(a) shows the outline of the environment (top view). The area of the environment is a 3D space with a size of $10 \times 10 \times 2$. For every new gameplay, the exterior 20 walls are positioned stationery to form a squared-shaped room with five walls each side. One goal, one agent and 15 interior walls are randomly positioned in the environment are randomly positioned within the squared room. If a goal is reached, the position of the walls remains and while the agent and the goal is relocated randomly. This state can be repeated number of times until the player decides to end or play a new game. This parameter setup is shown in Table 1.

Table 1. Parameter settings for the experiment

Parameter settings	Values	Remarks
Grid size	$10 \times 10 \times 2$	Default Unity3D unit
Number of agent, A	1	
Number of goal, G	1	
Number of walls, W	35	20 Walls surrounding the environment 15 randomly generated walls
Number of actions, k	8	Each action is a movement to one of the eight directions (N, NE, E, SE, S, SW, W, NW)

Below are summary of the process involved in the knowledge acquisition stage which focuses on obtaining the knowledge to train the agent outlined in Fig. 3(a):

i Agent percepts the environment using its sensors.
ii Human selects and controls the actions performed by the agent using one of the directional buttons (*up*, *left* or *right*) based on the state of the environment. For example, if the *right* key is pressed once, the agent will face the *NE* direction, twice *E* direction, thrice *SE* direction, etc. But data will only be captured when the forward (*up*) button is pressed.
iii The selected actions are recorded along with the state of the environment.

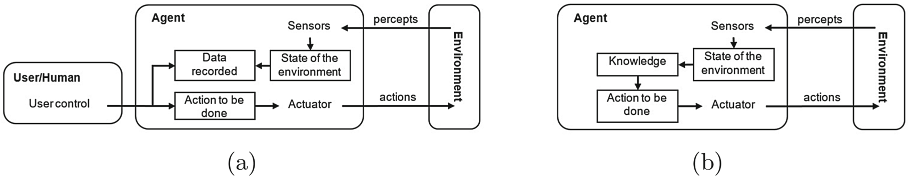


Fig. 3. (a) A Human Control Agent. It performs actions according to human control. State of environment and Human/user control information is recorded to be analysed for knowledge discovery; (b) A Knowledge-based Agent. Agent uses state of environment matched with policies to select which action to perform in environment.

The recorded data is analysed using DM techniques. This experiment is done separately. Patterns emerging from the analysis is extracted as a knowledge for the agent to perform in a new environment.

The knowledge application experiment (outlined in Fig. 3(b)) requires the agent to use the knowledge learned in a new environment without human intervention. This experiment was repeated three times, one for each of the three different DM techniques. Below summarizes the processes in the experiment:

i Agent percepts the environment using its sensors.
ii The agent matches the percepted environment to the knowledge, and uses the given projected outcome to perform its actions.

4.1 Recorded Data

We decided to record (i) the environment as viewed by the agent in 360 degrees around him with division analysis according to the cardinal and ordinal directions of a compass (i.e. eight basic directions in a compass), that consist of one of the following data: *1W*, *2W*, *1G*, *2G* (where 1 = near, 2 = far, W = wall and G = goal); and (ii) the k action taken by the agent.

4.2 Association and Classification Rule Mining

The following DM techniques were applied to the collected data for analysis:

i Association: Apriori
ii Classification: J48 C4.5 decision tree and naïve Bayes

Weka [13] explorer toolkit is implemented to perform these DM analysis. A ten-fold cross validation is used in our classification techniques and the confidence level of 0.75 is set for our association technique while all other options are set to default settings. The algorithms were conducted at different times throughout the study. The Weka simulation is conducted separately and all the three DM techniques are generated using the same dataset.

4.3 Evaluation Criteria

The evaluation was conducted by observing the agents performance based on two parameters: (i) the number of times the agent avoids an obstacle W; and (ii) The number of times the agent A approaches goal G if it is visible.

5 Results and Discussions

A reasonable pattern emerged from all the three DM techniques from a total of 4090 recorded training instances from 40 game sessions. In order to apply each of these knowledge to the agent, it requires an interpretation of the results. Some of the raw results as well as the translated results are summarized in Table 2.

Table 2. Sample of analysed raw and interpreted rule for apriori, decision tree and naive bayes

<table>
<tr><th>Technique</th><th colspan="4">Analysis Result</th><th colspan="4">Interpreted Result</th></tr>
<tr><td>Apriori</td><td colspan="4">• $angle\_N = 2G$ 518 $==>$ $action = N$ 518 $conf : (1)$
• $angle\_N = 1G$ $angle\_S = 2W$ 513 $==>$ $action = N$ 513 $conf : (1)$</td><td colspan="4">• IF $angle\_N == 2G$ THEN $action = N$
• IF $angle\_N == 1G$ AND $angle\_S = 2W$ THEN $action = N$</td></tr>
<tr><td>Decision Tree</td><td colspan="4">$angle\_N = 1W$
$\mid angle\_NE = 1W$
$\mid \mid angle\_NW = 1W$
$\mid \mid \mid angle\_E = 1W$
$\mid \mid \mid \mid angle\_SE = 1W$
$\mid \mid \mid \mid \mid angle\_W = 1W$
$: S(13.0/1.0)$
$\mid \mid \mid \mid \mid angle\_W = 2W$
$\mid \mid \mid \mid \mid \mid angle\_S = 1W$
$: W(9.0)$</td><td colspan="4">IF $angle\_N == 1W$
IF $angle\_NE == 1W$
IF $angle\_NW == 1W$
IF $angle\_E == 1W$
IF $angle\_SE == 1W$
IF $angle\_W == 1W$
THEN $action = S$
ELSE IF $angle\_W == 2W$
IF $angle\_S == 1W$
THEN $action = 2W$</td></tr>
<tr><td rowspan="6">Naive Bayes</td><td colspan="8">$angle\_N$</td></tr>
<tr><td>1W 12.0</td><td>64.0</td><td>91.0</td><td>1.0</td><td>109.0</td><td>55.0</td><td>16.0</td><td>29.0</td></tr>
<tr><td>2W 27.0</td><td>84.0</td><td>298.0</td><td>644.0</td><td>243.0</td><td>62.0</td><td>28.0</td><td>48.0</td></tr>
<tr><td>1G 1.0</td><td>1.0</td><td>1.0</td><td>213.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr>
<tr><td>2G 1.0</td><td>1.0</td><td>1.0</td><td>283.0</td><td>1.0</td><td>1.0</td><td>1.0</td><td>1.0</td></tr>
<tr><td>[$total$] 41.0</td><td>150.0</td><td>391.0</td><td>1141.0</td><td>354.0</td><td>119.0</td><td>46.0</td><td>79.0</td></tr>
</table>

Apriori and decision tree algorithms directly translate the results into knowledge. Naïve Bayes, however provides a more probabilistic approach, which requires us to use a random variable R to determine the actions to be performed for the agent. For example, suppose the analysed environment for N, NE, E, SE, S, SW, W, NW are *2W*, *1W*, *2W*, *1W*, *1W*, *1W*, *1W*, *1W*, respectively. The calculated probability and cumulative probability for each of the possible outcome is sampled as illustrated in Table 3 and we decide that the order of these

Table 3. Sample probability computed for the agent to select an action

Action	N	NE	E	SE	S	SW	W	NW
Probability	0.525	0.000	0.431	0.005	0.031	0.001	0.006	0.001
Cumulative probability	0.525	0.526	0.957	0.962	0.993	0.993	0.999	1

Table 4. Summary of experimental results

Technique	Criteria	Correctly performed	Incorrectly performed
Tree diagram	(i)	654 (100 %)	0 (0 %)
	(ii)	7 (100 %)	0 (0 %)
Nave Bayes	(i)	611 (98.5 %)	9 (1.5 %)
	(ii)	66 (93 %)	5 (7 %)

actions are stationary. If R, lets assume, equals to 0.61, based on the cumulative probability, then an action $k = E$ is selected.

Although the results given by apriori algorithm shows a reasonable knowledge to the agent, the limitation to this is that the knowledge is incomplete. Some of the environment percepted by the agent are unknown which results in action k to be *null*. Because of this, experiment to run the simulator is impracticable.

Table 4 shows summary of experiments. In a sample of 661 instances that has been recorded to run the interpreted result of decision tree, 654 (100 %) actions are performed as expected with 0 % or no incorrect performed action made. Although it shows an excellent result, the problem with this technique is that the agent will only perform the action stated repeatedly when logically, there are multiple choices. This behavior sometimes lead the agent into going in circles and end up in a loop. The seven actions stated in criteria (ii) where it found the goal most likely happened because the goal is already nearby when the game started.

The agent also performed rather well (>90 %) by using naïve Bayes technique with a collected sample of 691 instances. The 1.5 % and 7 % incorrect action is, by any chance, selected because although the probability of that action is close to 0, there is still a possibility for that action to be selected. The advantage of using this technique is that it gives more flexibility on the action chosen compared to that of apriori and decision tree but the drawback is that some of the actions are performed incorrectly.

6 Conclusions and Future Work

In this paper, we demonstrated a DM approach to study the learning ability of an agent to navigate in a 3D Environment. We collected a total of 4090 data and implemented apriori, decision tree and naïve Bayes by using Weka toolkit as the techniques used to acquire knowledge for the agent based on human controlling navigation dataset. Our result shows the agent can learn to navigate in the 3D Environment through this knowledge but with a few limitations.

As DM techniques allows knowledge discovery from dataset, possible future work is to include more actions to improve the immersion of the 3D environment and knowledge obtained by the agent can be further improvised, either by incorporating two or three techniques or by inserting extra knowledge to the learned knowledge.

References

1. Witten, I., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers Inc., San Francisco (2011)
2. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
3. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the Eleventh International Conference on Data Engineering, pp. 3–14, March 1995
4. Muhammad, J., Bucak, I.O.: An improved q-learning algorithm for an autonomous mobile robot navigation problem. In: International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE) (2013)
5. Jaafar, J., McKenzie, E.: Autonomous virtual agent navigation in virtual environments. Int. J. Comput. Electr. Autom. Control Inf. Eng. **4**, 91–98 (2010)
6. Hsieh, J.L., Sun, C.T.: Building a player strategy model by analyzing replays of real-time strategy games. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). IEEE (2008)
7. Dereszynski, E., Hostetler, J., Fern, A., Dietterich, T., Hoang, T.T., Udarbe, M.: Learning probabilistic behavior models in real-time strategy games. In: Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (2011)
8. Baldominos, A., Albacete, E., Marrero, I., Saez, Y.: Real-time prediction of gamers behavior using variable order Markov and big data technology: a case of study. Int. J. Interact. Multimedia Artif. Intell. **3**, 44–51 (2016)
9. Weber, B.G., Mateas, M., Jhala, A.: Learning from demonstration for goal-driven autonomy. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
10. Gemine, Q., Safadi, F., Fonteneau, R., Ernst, D.: Imitative learning for real-time strategy games. In: 2012 IEEE Conference on Computational Intelligence and Games (CIG), pp. 424–429. IEEE (2012)
11. Phon-Amnuaisuk, S.: Learning to play tetris from examples. In: Phon-Amnuaisuk, S., Au, T.W. (eds.) Computational Intelligence in Information Systems. AISC, vol. 331, pp. 255–264. Springer, Heidelberg (2015)
12. Wang, D., Tan, A.H.: Creating autonomous adaptive agents in a real-time first-person shooter computer game. In: IEEE Transactions on Computational Intelligence and AI in Games, vol. 7, pp. 123–138. IEEE (2015)
13. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2011)
14. Phon-Amnuaisuk, S.: Evolving and discovering tetris gameplay strategies. Procedia Comput. Sci. **60**, 458–467 (2015)
15. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson, New York (2010)