Expert Systems With Applications 56 (2016) 89–99

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Towards autonomous behavior learning of non-player characters in games

Shu Feng*, Ah-Hwee Tan

School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 27 January 2014
Revised 6 February 2016
Accepted 23 February 2016
Available online 4 March 2016

Keywords:
Behavior learning
Reinforcement learning
Imitative learning
Self-organizing neural network
Intelligent agent

ABSTRACT

Non-Player-Characters (NPCs), as found in computer games, can be modelled as intelligent systems, which serve to improve the interactivity and playability of the games. Although reinforcement learning (RL) has been a promising approach to creating the behavior models of non-player characters (NPC), an initial stage of exploration and low performance is typically required. On the other hand, imitative learning (IL) is an effective approach to pre-building a NPC's behavior model by observing the opponent's actions, but learning by imitation limits the agent's performance to that of its opponents. In view of their complementary strengths, this paper proposes a computational model unifying the two learning paradigms based on a class of self-organizing neural networks called Fusion Architecture for Learning and COgnition (FALCON). Specifically, two hybrid learning strategies, known as the Dual-Stage Learning (DSL) and the Mixed Model Learning (MML), are presented to realize the integration of the two distinct learning paradigms in one framework. The DSL and MML strategies have been applied to creating autonomous non-player characters (NPCs) in a first person shooting game named Unreal Tournament. Our experiments show that both DSL and MML are effective in producing NPCs with faster learning speed and better combat performance comparing with those built by traditional RL and IL methods. The proposed hybrid learning strategies thus provide an efficient method to building intelligent NPC agents in games and pave the way towards building autonomous expert and intelligent systems for other applications.

1. Introduction

Intelligent non-player characters (NPCs) in computer games can potentially make the games more challenging and enjoyable. As such, behavior modeling of non-player character (NPC) has become an important component in computer games, especially in first person shooting games (FPS) (Wang, Subagdja, Tan, & Ng, 2009; Wang & Tan, 2015).

In the game environment, each NPC is essentially an autonomous agent, which is expected to function and adapt by themselves in a complex and dynamic environment. Consequently, a popular approach to developing intelligent agents is through machine learning algorithms.

In particular, reinforcement learning (RL) is considered by many to be an appropriate paradigm for an agent to autonomously acquire its action policy through interacting with its environment in a dynamic process. In general, an RL agent makes responses to the environment in order to maximize the future expected rewards with respect to its goals and motivations. However, in a first person shooting game, an NPC without prior knowledge will perform poorly at the initial stage as they have to spend substantial time in exploring and learning the environmental information. Playing with these NPCs certainly takes the fun out of the game. Moreover, specific types of knowledge may be too complex to learn through reinforcement feedback.

To overcome these drawbacks, a possible remedy is to pre-insert domain knowledge into the learning agents, in order to increase learning efficacy, shorten convergence time as well as enhance NPCs' performance. Although there have been extensive works towards improving RL with prior knowledge, the methods for obtaining and integrating knowledge are still an open problem. Most of the earlier works complement reinforcement learning by direct inserting prior knowledge through either encoding domain knowledge in the learning architecture (Busoniu, Schutter, Babuska, & Ernst, 2010; Shapiro, Langley, & Shachter, 2001), adding prior knowledge as a rule base (Song, Gu, & Zhang, 2004), or using

* Corresponding author at: School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore.

E-mail addresses: feng0027@e.ntu.edu.sg (S. Feng), asahtan@ntu.edu.sg (A.-H. Tan).

http://dx.doi.org/10.1016/j.eswa.2016.02.043
0957-4174/

an added-on module to provide prior knowledge (Dixon, Malak, & Khosla, 2000; Moreno, Regueiro, Iglesias, & Barro, 2004). An obvious drawback of direct insertion is that the prior knowledge cannot be used in exploitation during learning and cannot adapt to changes in the environment.

In contrast to reinforcement learning, imitative learning with explicit supervisory teaching signals is a promising approach to acquiring complex behavior for autonomous agents. The knowledge learnt by imitation can be used readily as the agent's behavior model (Feng & Tan, 2010). Imitative learning and reinforcement learning can been seen as two complementary learning paradigms. While the former is effective and fast in acquiring patterns, it strictly relies on the training data and typically is not used in real time adaptation. On the other hand, reinforcement learning is good in learning from experience and adapting to the environment in real time. However, it is less effective for fast learning due to the lack of explicit teaching signals. In view of their complementary strengths, this work aims to combine the fast learning capability of IL with real-time adaptive ability of RL for a better performance.

Specifically, this paper presents two hybrid learning strategies, known as Dual-Stage Learning (DSL) and Mixed Model Learning (MML) to realize the integration of the two learning paradigms in one unified framework based on a class of self-organizing neural networks, namely Fusion Architecture for Learning and COgnition (FALCON) (Tan, 2004; Xiao & Tan, 2007). FALCON learns cognitive codes encoding multi-dimensional mappings simultaneously across the multi-modal pattern channels. By using competitive coding as the underlying adaptation principle, FALCON is capable of supporting multiple learning paradigms, including unsupervised learning, supervised learning and reinforcement learning (Tan, Carpenter, & Grossberg, 2007).

The DSL strategy combines imitative learning and reinforcement learning in two stages. In the imitative learning stage, FALCON learns from opponent behavior patterns to build the initial behavior model of an autonomous agent. Subsequently, in the reinforcement learning stage, the agent further adapts in real time through Q-learning while applying the prior knowledge for exploitation. Compared with DSL, the MML strategy combines the two learning paradigms tightly into one model, in which both IL and RL work in an interleaving manner sharing a common knowledge field.

We have evaluated various learning methods and strategies in a first person shooting game named Unreal Tournament(UT). Our experiments show that the NPCs learned with DSL and MML produce a higher level of performance compared with the traditional RL and IL methods.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 defines the problems addressed by this work. Section 4 explains the issues and challenges. Section 5 introduces the learning model and Section 6 presents the methods of DSL and MML. Section 7 reports the learning tasks and the experiments. Finally, section 8 concludes and discusses the future work.

2. Related work

In the past years, NPCs in computer games have developed significantly in terms of behavior modeling. This section reviews some of the related work.

For commercial games, the most commonly used method is via rule based approaches. Ji and Ma (2014) proposed a behavior tree to manage the controlling of behaviors. By designing complex intelligent role behaviors in logical ways, this method could easily integrate expert experience into the intelligent system. However, as it inherited the use of action and condition rules based on finite state machine, the agent is not able to adapt to different environment. Akbar, Praponco, Hariadi, Mardi (2015) applied Gaussian distribution with fuzzy logic to create natural variation actions of each NPC and select the optimal action. As the fuzzy method still has to follow logic rules, this method doesn't help NPCs to evolve and capture new knowledge.

In view of the limitations of the rule-based approaches, learning based techniques have attracted much attention for behavior modeling. David, van den Herik, Koppel, and Netanyahu (2014) used genetic algorithm for evaluation and search mechanism for decision makings. Stanley, Bryant, and Miikkulainen (2005) applied evolution algorithm to enable the NPCs evolve behaviors through interacting with players, thus keeping the game interesting. Although evolutionary algorithms can improve the performance continuously, the final solution cannot be guaranteed as the global optimal solution. Moreover, tuning parameters during learning is time consuming and not suitable for real-time video games.

On the other hand, many have applied imitative learning to create NPCs by mimicking the behavior patterns of human beings in order to achieve the human-like behaviors (Bauckhage, Thurau, & Sagerer, 2003; Feng & Tan, 2010; Zanetti & Rhalibi, 2004). These imitation based learning enables fast learning and is capable of acquiring complex behavior patterns. However, imitative learning requires specific observations to be available. Furthermore, in real time processing, imitative learning cannot associate the behavior with the underlying motivations or goals.

Besides imitative learning, many have applied reinforcement learning (RL) successfully to autonomous NPCs for learning strategies and behaviors in a dynamic environment. Especially in combat scenarios games, RL is good at helping NPCs to learn through experience with the supplement of necessary initial knowledge. Glavin and Madden (2015) applied reinforcement learning to enable NPCs to get experience from gaming experience, and improve their fighting skills over time based on the damage given to opponents. Ponce and Padilla (2014) applied MaxQ-Q based reinforcement learning within a hierarchical structure to enhance user's experience. Wang and Tan (2015) utilized reinforcement learning in a first person shooting game to learn behavior strategy and effectiveness of different weapons. However, using pure reinforcement learning, an initial stage of exploration is required and depending on the problem domain, this process may take a long time.

In view of the above issue, a popular way to initialize the learning agent is with human knowledge (Dixon et al., 2000; Unemi, 2000). They improve the performance of reinforcement learning by introducing relatively simple human knowledge such as intrinsic behaviors to reduce learning time. However, the prior knowledge is embedded and not represented in the same form as the learned knowledge. As such, the knowledge cannot be further modified in real-time learning. Bayesian based method is another principle way to incorporate prior knowledge into reinforcement learning (Doshi-Velez, Pfau, Wood, & Roy, 2015; Ghavamzadeh, Mannor, Pineau, & Tamar, 2015; Jonschkowski & Brock, 2014). Here the prior knowledge mostly refers to probability distributions. Taken from prior observations, the knowledge helps to start up the learning as well as to continuously guide decision makings. However, they need to be designed beforehand based on the specific problem domains, not to mention the experience based knowledge may include objective bias. Kengo, Takahiro, and Hiroyuki (2005) enhanced the reinforcement learning agent by applying goal state prior knowledge to the agent in order to modulate the decision making by giving priorities to the goal oriented actions. In the chosen game scenarios, prior knowledge is designed by the domain expert. Again, once it is applied, the knowledge fixed and cannot be further adapted. Framling (2007) introduced a reinforcement learning model with pre-existing knowledge. A bi-memory system including the concepts of short term and long term memories is proposed to modulate the exploration in state