A Demonstration of Agent Learning with Action-Dependent Learning Rates in Computer Role-Playing Games

Maria Cutumisu, Duane Szafron

Department of Computing Science, University of Alberta
Edmonton, Canada
{meric, duane}@cs.ualberta.ca

Abstract

We demonstrate combat scenarios between two NPCs in the *Neverwinter Nights (NWN)* game in which an NPC uses a new learning algorithm ALeRT (Action-dependent Learning Rates with Trends) and the other NPC uses a static strategy (NWN default and optimal) or a dynamic strategy (dynamic scripting). We implemented the ALeRT algorithm in NWScript, a scripting language used by *NWN*, with the goal to improve the behaviours of game agents. We show how our agent *learns* and *adapts* to changes in the environment.

Introduction

Game agents or non-player characters (NPCs) have a multitude of roles during an interactive story. They can be the player character's (PC) friend, enemy, or they can just provide background entertainment. Since it is not enough to develop entertaining behaviours for a fixed scenario, NPCs should constantly adapt to new situations (e.g., different PCs, different environments). This is not a trivial task, because in a game there are usually hundreds or thousands of NPCs and game environments are changing at a fast pace, creating a plethora of different scenarios for the NPCs. Therefore, most games have NPCs with manually scripted actions that lead to repetitive and predictable behaviours. Learning techniques are rarely applied in commercial games due to algorithms that converge too slowly in a game environment, testing difficulties, and fears that agents can learn wrong behaviours. Dynamic scripting (Spronck et al. 2004) is a learning technique that successfully combines rule-based scripting with reinforcement learning (RL) (Sutton and Barto 1998). However, the rules have to be ordered and the agent cannot discover other rules that were not included in the rule-base.

Figure 1. Agents fighting in Spronck's arena.

We demonstrate ALeRT (Cutumisu et al. 2008), a new action-dependent learning rate algorithm based on Sarsa(λ) that provides agents with a mechanism to *learn* and to *adapt* to changes in the environment. We implemented this algorithm in *NWN* (NWN 2008) with the goal of showing that adaptive behaviours can be integrated in a commercial game to improve NPC behaviours. Our experiments evaluate learning rates and adaptability to new situations in a changing real-time game world. We use Spronck's pre-built arena combat module for *NWN* (NWN Arena 2008), shown in Figure 1, to evaluate the quality of our learning algorithm. We modified Sarsa(λ) to identify "safe" opportunities for fast learning using a technique based on the Delta-Bar-Delta measure (Sutton 1992) and to support action-dependent learning rates consistent with the WoLF principle of "win or lose fast" (Bowling and Veloso 2001). ALeRT speeds up the learning rate for an action when there is a trend for that action and slows it down otherwise.

Demonstration Overview

During the demonstration, we will show graphs from our experiments. For each experiment, we ran a set of trials. Each trial consisted of one or two phases of 500 episodes. In the first phase, we evaluated how quickly an agent is able to *learn* a winning strategy without prior knowledge. In the second phase, we evaluated how quickly an agent can *adapt* by discovering a new winning strategy after an equipment configuration change (*Melee*, *Ranged*, *Heal*). Each configuration has an optimal action sequence, shown in Table 1. We recorded the average number of wins for each agent for each group of 50 episodes. Since each experiment took at least ten hours (each combat round takes 6 seconds), we will show modules saved after various numbers of episodes to illustrate the changes in strategy.

Config	Melee Weapon	Ranged Weapon	Healing Potion	Enhancement Potion	Optimal Strategy
Melee	GS+1	HC+1, B	Cure Serious	Speed	S-M
Ranged	LS	LB+7, A+5	Cure Light	Speed	S-R
Heal	LS+3	HC+1, B	Heal	Speed	S-M*-H-M*

Table 1. Agent configurations and their optimal strategies.

We ran experiments between two agents with one of seven strategies: NWN, the default *NWN* agent; RL_0, RL_3, and RL_5, traditional Sarsa (λ) dynamic learning agents with $\alpha = 0.1$, $\varepsilon = 0.01$, $\gamma = 1$ fixed, but different values for λ (0, 0.3, and 0.5 respectively); ALeRT, our algorithm based on action-dependent learning rates that vary according to trends, with the parameters initially set to $\alpha = 0.2$, $\varepsilon = 0.02$, $\lambda = 0$ (fixed), and $\gamma = 1$; M1, Spronck's dynamic scripting agent (learning method 1), and, finally, OPT, the optimal strategy for each of the configurations (e.g., speed followed by repeated melee actions for the *Melee* configuration). The x-axis indicates the episode and the y-axis indicates an agent's average winning percentage at that episode, over the fifty previous episodes. Each data point represents the average win percentage over all trials.

ALeRT and M1 vs. Static Opponents

The results in Figure 2 compare ALeRT vs. a static agent to M1 vs. the same static agent for the *Melee* and *Ranged* configurations. We used the NWN and OPT static agents. The upper four traces show the results against NWN. M1 had a higher final winning rate (94%) than ALeRT (70%) against NWN for the *Melee* configuration and for the *Ranged* configuration (90% vs. 78%). These winning rates are more than 20% higher than ALeRT's winning rates, but ALeRT's behaviour is desirable in games, where an NPC should challenge, not defeat the PC. NWN performs poorly since if an agent starts with a sword equipped, it only selects from melee and heal, never from ranged or speed.

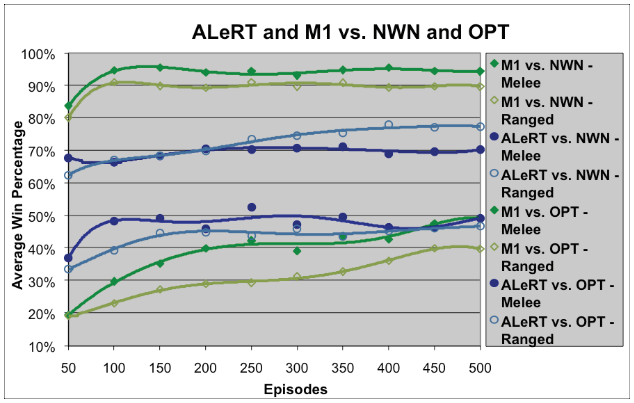

Figure 2. ALeRT and M1 vs. NWN and OPT.

The lower four traces show the results against OPT. ALeRT converged to OPT in both configurations, but M1 did not converge to OPT for *Ranged* by the end of the experiment. M1 converged more slowly than ALeRT for *Melee*, since the latter won 48% after the first 100 episodes and exceeded 46% after that. M1 won only 30% at episode 100 and did not reach 46% until episode 450 for *Melee*. For *Ranged*, ALeRT won 44% after episode 150 and 46% by episode 450, while M1 achieved its highest win rate (40%) after episode 450. For simplicity, we do not show the traces for RL_0, RL_3, and RL_5. Although the RL agents outperformed NWN, they did not converge to OPT.

ALeRT vs. M1, a Dynamic Opponent

To measure the dynamic agents' adaptability, we assessed how fast ALeRT and M1 recover after a change in configuration: *Melee-Heal*, *Melee-Ranged*, *Ranged-Melee*, *Ranged-Heal*, *Heal-Melee*, *Heal-Ranged*. We changed each agent's equipment configurations at episode 501 and ran 500 more episodes with the new configurations. We ran 50 trials for each of the combined configurations. The cumulative results over 300 trials are shown in Figure 3. ALeRT adapted faster in the first phase, but the major advantage of ALeRT over M1 is illustrated in the second phase. ALeRT adapted faster to changes in the environment, defeating M1 at a rate of 80% at episode 1000. Although ALeRT may not always find the optimal solution, it finds a policy that defeats the opponent. Figure 3 shows that RL_0 did not find the optimal strategy in the first phase against a dynamic opponent.

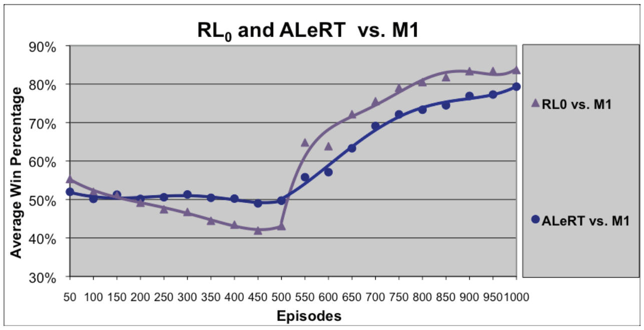

Figure 3. ALeRT and RL_0 vs. M1, a dynamic opponent.

Conclusions

We demonstrate ALeRT, an algorithm that makes three modifications to traditional RL techniques: it identifies trends in reward sequences, modifies the learning rate over time in an action-dependent way, and adjusts the exploration rate according to a loss or win of each episode. Our empirical evaluation shows that ALeRT adapts better than a dynamic opponent in a changing environment and it also performs well against static opponents.

References

Bowling, M., and Veloso, M. 2001. Rational and Convergent Learning in Stochastic Games. In *Proceedings of the 7th International Joint Conference on AI*, 1021-1026.

Cutumisu, M., Szafron, D., Bowling, M., and Sutton, R. S. 2008. Agent Learning Using Action-Dependent Learning Rates in Computer Role-Playing Games. In *Proceedings of the 4th AIIDE Conference*, October 22-24, 2008, Stanford, USA.

NWN 2008. http://nwn.bioware.com.

NWN Arena. 2008. http://www.cs.unimaas.nl/p.spronck /GameAI/OnlineAdaptation3.zip.

Spronck, P., Sprinkhuizen-Kuyper, I., and Postma, E. 2004. Online Adaptation of Game Opponent AI with Dynamic Scripting. *International Journal of Intelligent Games and Simulation* 3(1): 45–53.

Sutton, R.S. 1992. Adapting Bias by Gradient Descent: An Incremental Version of Delta-Bar-Delta. In *Proceedings of the 10th National Conference on AI*, 171-176.

Sutton, R.S., and Barto, A.G. eds. 1998. *Reinforcement Learning: An Introduction*. Cambridge, Mass.: MIT Press.