Autonomous Agents 1
Assignment 2

By Group 4: Gieske, Gornishka, Koster, Loor

December 3, 2014

Contents

1 Introduction **3**

2 Theory **4**
2.1 Independent Q-learning 4
2.2 Minmax Q-Learning 4
2.3 Other learning 4

3 Implementation **4**

4 Analysis **5**
4.1 Independent Q-Learning 5
4.2 Minmax Q-learning 5
4.3 Other Learning 5

5 Conclusion **6**

6 Future work **7**

7 Files attached **8**

8 Sources **8**

1 Introduction

2 Theory

2.1 Independent Q-learning

2.2 Minmax Q-Learning

2.3 Other learning

3 Implementation

The implementation consists of the following files:

Agents_new
: This file contains implementions of the Agent class, the Prey class and the Predator class. Both the predator and the prey inherit functions of the Agent class. The Agent class contains functions any agent needs, such as a set of actions, a policy and other functions. As the predator is the agent is the agent this implementation focuses on, the predator class contains more functions than the predator class.

Helpers
: This file contains many helper functions. These functions aid in computation and decision making, but cannot (and need not) be part of a specific class.

Other_objects
: This file contains the Policy and Environment classes. The environment of the game as well as the rules are implemented in the Environment class. The Policy class contains the implementation of Q-Learning, Sarsa, ϵ-greedy, softmax action selection and more functions that help in determining and optimizing a policy as well as choosing an action of this policy.

Newstate
: This file contains the Game class as well as a demonstration function. The Game class instantiates the game, initialized the predator and the prey and assigns policies to these. The game is run N times and the result is printed. The demonstration function also performs Q-Learning, Sarsa and Monte Carlo. It also uses ϵ-greedy and softmax action selection. The results are printed in the command line and graphs are used for analysis.

4 Analysis

This section discusses the results of the implementations. In order to display and compare results, graphs are used. The title describes which parameters are analysed and the legend shows which color represents which setting of said parameters.

4.1 Independent Q-Learning

4.2 Minmax Q-learning

4.3 Other Learning

5 Conclusion

6 Future work

7 Files attached

- newstate.py
- agents_new.py
- other_objects.py
- helpers.py ...

8 Sources

References

[1] Peter Auer, Thomas Jaksch, and Ronald Ortner. "Near-optimal regret bounds for reinforcement learning". In: *Advances in neural information processing systems*. 2009, pp. 89–96.

[2] Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 1998.

[3] Richard Dearden, Nir Friedman, and Stuart Russell. "Bayesian Q-learning". In: *AAAI/IAAI*. 1998, pp. 761–768.

[4] Karun Rao and Shimon Whiteson. "V-MAX: Tempered Optimism for Better PAC Reinforcement Learning". In: *AAMAS 2012: Proceedings of the Eleventh International Joint Conference on Autonomous Agents and Multi-Agent Systems*. 2012, pp. 375–382.