CS8803-O03 Reinforcement learning Project 3 report

Rohan D. Kekatpure
Email: rdk@gatech.edu

I. Introduction

Multi-agent Q-learning is a nontrivial departure from regular (deterministic action, single Q function) Q learning in two ways: (1) state values are functions of multiple Q's (2) optimal policies are allowed to be stochastic. The Greenwald paper [1] demonstrates the inadequacy of regular Q learning and convergence of multi-agent Q learning learning. Project 3 aims to deepen our understanding of multi-agent (adversarial or cooperative) Q learning through reproduction of Greenwald's results for a 2×4 grid game called soccer.

II. Quick theory tour

The paper presents four multi-agent Q learning options:

1) **Regular:** Ignore the opponent's Q function and actions. Agents are only coupled through rewards. The policy is deterministic and Q function at each state for each agent is simply a vector of n values ($n = 5$ in our case). The state value is the max for Q_i:

$$V_i = \max_{a \in A_i} Q_i(s, a) \tag{1}$$

2) **Friend Q:** Ignore the opponent's Q function, but consider its actions. Optimal policy is deterministic, and Q function at each stage is a $n \times n$ matrix. The value for each state is the max across this matrix:

$$V_i = \max_{\vec{a} \in A_1 \times A_2} Q_i(s, \vec{a}) \tag{2}$$

3) **Foe Q:** Ignore opponent Q function, consider its actions, but calculate the value function using **maximin** instead of simple max. This requires linear programming (LP). Maximin also allows **stochastic** optimal policies represented as a probability distribution over n action values. The Q is an $n \times n$ matrix and the state value is:

$$V_i = \max_{\pi \in \mathrm{PD}(A)} \min_{o \in O} \sum_{a \in A} \pi_a Q_i(s, a, o) \tag{3}$$

4) **CE Q:** Consider joint actions and compute state value as a function of agents' Q values

$$V_i = f_i(Q_1, Q_2) \tag{4}$$

where the functions f_i are linear combinations of Q_i and determined through linear programming.

III. Implementation methodology

A. Overview

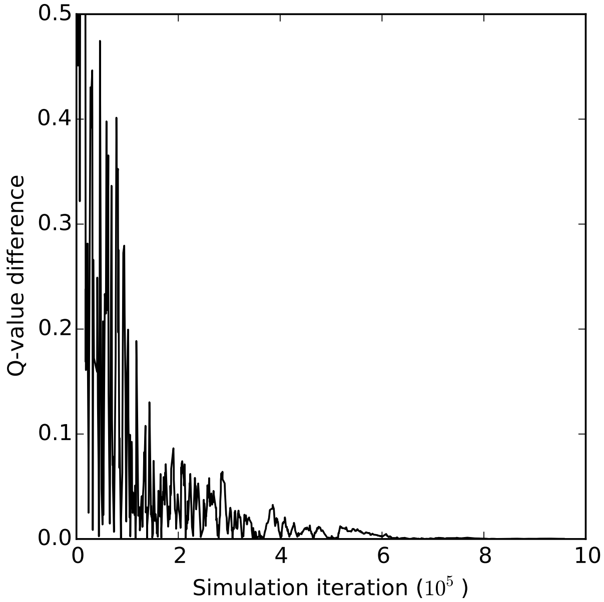
B. Software dependencies

IV. Experiments

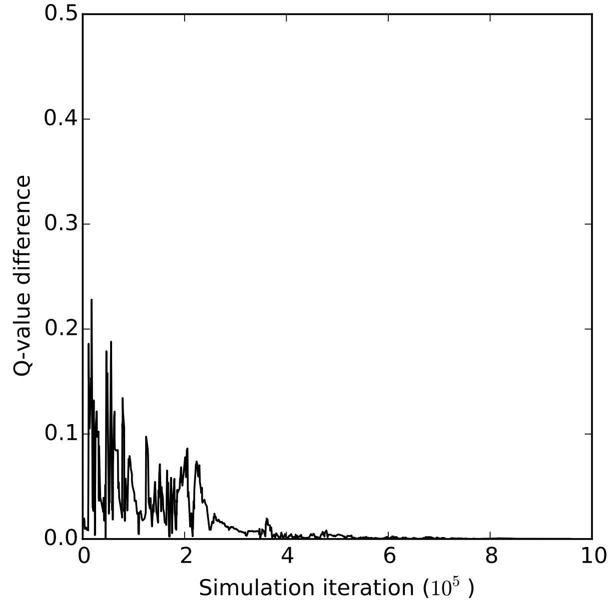
Summary

References

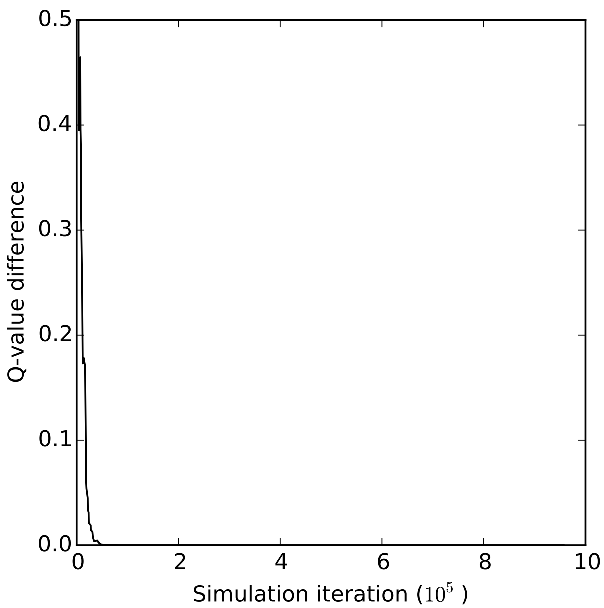
[1] A. Greenwald and K. Hall, "Correlated Q learning," *ICML*, 2001.



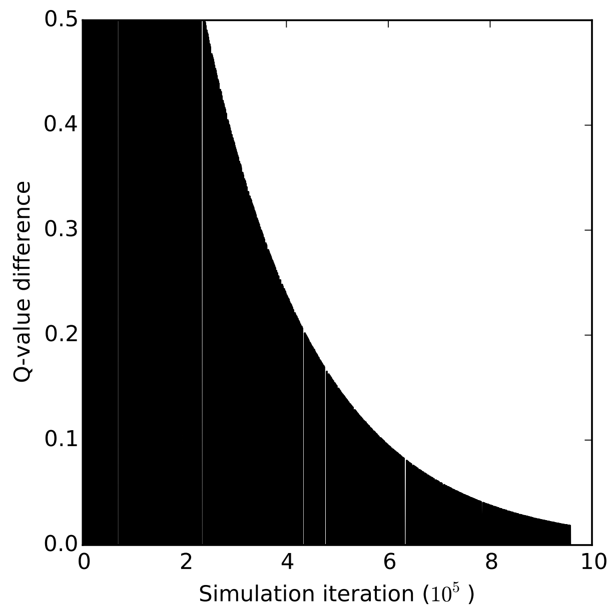
(a) CE-Q



(b) Foe-Q



(c) Friend-Q



(d) Q-learning