Value function RL in Markov Games

phoenix

April 19, 2020

Contents

1 Introduction

Markov games are a model for multiagent envs to study MARL. This paper describes a few RL value based algos and their convergence guarantees. This borrows concepts from game theory for reasoning.

1.1 Single agent envs

In an MDP framework, the state is always known while which action was taken will be based on the reward and state quantified via the value function.

1.1.1 MDPs and single agents

MDPs is given as the following <S, A, T, R, β>

- S is the finite set of states
- A is the finite set of actions
- T : S X A $\rightarrow \pi$(S)
- T(s, a, s') is the transition function
- R : S x A $\rightarrow$ R
- β is the discount factor (small values mean near term is more important).

In an MDP, an agent should act in way to maximize some measure of the long term discounted reward. You learn a policy, Π: S $\rightarrow \pi$(A) which gives a probability dist over the actions. For every MDP, there is a deterministic stationary policy.

1. Theorem 1 In a single agent env, an agent following the Q learning update will converge to the optimal Q-function. Furthermore, if it follows a GLIE (greedy in the limit with infinite exploration) policy and the policy is optimal, it will converge in behavior.

   A GLIE is one with the following If a state is visited infinitely often, then each action in that state is

   - chosen infinitely often (with probability 1).
   - In the limit (as t $\rightarrow \infty$), the learning policy is greedy with respect to the learned Q-function (with probability 1).

1.2 Multiagent envs

In the MDP model, a decision making agent interacts with its env, represented as a probabilistic transition function. In this view, we have to keep the other agents fixed in their behavior.

1.2.1 Markov games

An n-player Markov game is defined by a <S,A_1, A_2, ..., A_n, T, R_1, R_2,..., R_n, β> where

- A_1, A_2, ..., A_n is a collection of actions available to each agent.
- T: S x A_1, A_2, ..., A_n $\rightarrow \pi(S)$
- R_i : S X A_1, A_2, ..., A_n $\rightarrow$ R

Like the single agent env, the agents here try to maximize their expected long term discounted reward. If the set of agents adopts stationary policies (π_1, π_2, ..., π_n), then the Q function is pretty similar to the single agent case.

$$\begin{aligned} Q_i^{\pi}(s, a_1, a_2, ..., a_n) = R_i(s, a_1, ..., a_n) \\ + \beta \sum_{s' \in S} T(s, a_1, ..., s') \\ . \sum_{a_1' .., a_n'} \pi_1(s', a_1') ... \pi(s', a_n') Q_i^{\pi}(s', a_1', ... a_n') \end{aligned} \tag{1}$$

The above equation is just the Q function defined over the joint actions for each agent where each agent gets its reward according to its reward function and the transitions depend on the joint action taken by the set of agents.

$$\begin{aligned} Q_{\Delta i}^{\pi}(s, a_1, a_2, ..., a_n) = R_i(s, a_1, ..., a_n) \\ + \beta \sum_{s' \in S} T(s, a_1, ..., s') \\ \max_{a_i'} . \sum_{a_1' .., a_n - a_i} \pi_1(s', a_1') ... \pi(s', a_n') Q_{\Delta i}^{\pi}(s', a_1', ... a_n') \end{aligned} \tag{2}$$

All the other policies are held fixed and only the ith agent's policy is not. This turns it into a single agent env/

1. Theorem 2 In a multiagent env, an agent following the Q learning update rule will converge to the optimal response Q-function as long as the other agents converge in behavior. Furthermore, if the agent follows a GLIE policy and its best response policy is unique, it will converge in behavior.

Caveat: This does not imply that two simultaneous Q-learners will converge to mutual best responses.

Another problem is that in an MDP (single agent env), there is a single optimal policy. This means that there is no state from which any other policy can achieve a better value. Every MDP has at least one optimal policy and one is stationary and deterministic.

For Markov games, this is not true because you have other agents in the env and your policy depends on that. How to deal with this? The answer is to define an optimal behavior in Nash equilibrium. A set of policies is in Nash equilibrium if each is the best response to the others. That is,

$$\sum_{a_1 \ldots a_n} \pi(s, a_1) \ldots \pi(s, a_n) Q_i^{\pi}(s, a_1, \ldots, a_n) \tag{3}$$

is equal to

$$\max_{a_i'} . \sum_{a_1' .., a_n - a_i} \pi_1(s', a_1') \ldots \pi(s', a_n') Q_{\Delta i}^{\pi}(s', a_1', \ldots a_n') \tag{4}$$

for every agent. In human, that means that every agent attempts to maximize their own stonks! The thing is that every markov game has a Nash equilibrium in stationary policies. However, they may not be deterministic and are mostly stochastic. Rock, paper, scissors is one example where there is not deterministic policy that is optimal.

1.3 Nash Q learning

Define $\text{Nash}_\text{i}(\text{s}, \text{Q}_1, \ldots , \text{Q}_\text{n})$ to be a one stage Nash equilibrium policy for the agent i in state s, where the total payoff to the agent is defined by the Q function in state s. Define $\text{Val}_\text{i}(\text{s}, \text{Q}_1, .. \text{Q}_\text{n})$ to be the value obtained by the agent i in state s

$$\begin{aligned} Val_i(s, Q_1, \ldots Q_n) = \sum_{a_1 \ldots a_n} & Nash_1(s, Q_1 \ldots Q_n)[a1]] \\ & .Nash_n(s, Q_1 \ldots Q_n)[an]] \\ & Q_i[s, a_1 \ldots a_n] \end{aligned} \tag{5}$$

In Zorg-speak, the value received by the agent i is the exp value of the agent's future reward. The expected value is taken over all the possible joint

actions of the n agents where we expect to choose action in accordance with the Nash equilibrium policy. Value need not be unique as there may be multiple Nash equilibria with different values. Another issue is that even if the game admits only one Val_i, the Nash policy may not achieve this as the other policies may not admit this value function. One of the way to extend Q learning in Markov games is to use the Nash equilibrium values in place of the max to estimate each agent's Q function.

$$Q_i[s, a_1...a_n] = (1-\alpha)Q_i[s, a_1...a_n] + \alpha(r + \beta Val_i[s', Q_1...Q_n]) \tag{6}$$

The above is not known to converge even if the game has a unique value.

1.4 Adversarial equilibria

Now, an adversarial equilibria in an n-player game is defined as a saddle point of policy. If any of the agents deviate from this, they hurt only help themselves and help everyone else.

$$\begin{aligned} &\sum_{a_1...a_n} \pi_1(s, a_1)...\pi_n(s, a_n)Q_i[s, a_1, \ldots, a_n] \\ &\geq \sum_{a_1...a_n} \pi_1(s, a_1)...\pi_n(s, a_n)\pi'_i(s, a_i)Q_i[s, a_1, \ldots, a_n] \end{aligned} \tag{7}$$

In Pokemon speak, you'd rather maintain quid pro quo. Another way to put this mathematically is

$$\begin{aligned} &\sum_{a_1...a_n} \pi_1(s, a_1)...\pi_n(s, a_n)Q_i[s, a_1, \ldots, a_n] \\ &\leq \sum_{a_1...a_n} \pi'_1(s, a_1)...\pi'_n(s, a_n)\pi_i(s, a_i)Q_i[s, a_1, \ldots, a_n] \end{aligned} \tag{8}$$

In villain speak, you want everyone else to shoot themselves in the foot.

1.4.1 Theorem 3

In a multi-agent env, an agent following the Nash Q-learning update will converge to the optimal Q-function with prob 1 as along as all the Q-functions encountered have adversarial equilibria and these are used for the update rule. If the limit equilibrium is unique, a GLIE policy will result in convergence to optimal behavior.

Now, the problem with the above is that it's difficult to ensure that there's an adversarial equilibria. Also, the intermediate Q functions may not have

adversarial equilibria even if we somehow initialized immediate rewards and the Q values to have adversarial equilibria.

1.5 Zero Sum games.

This is a specialization of Markov games where two agents have diametrically opposed goals. R_1 (s, a_1, a_2) = - R_2(s, a_1, a_2). Thus, there is only a single reward funciton, R_1. Agent 1 wants to maximize R_1 while agent 2 wants to minimize the same.

The notion of Nash equilibrium in this a two player zero sum game is essentially each of the policy is evaluated with respect to a policy that makes it look the worst. Note that this performance measure prefers conservative strategies - Maximize rewards (government paychecks) in the worst case (global pandemics).

$$\begin{aligned} &\sum_{a_1 \ldots a_2} \pi_1(s, a_1)\pi_2(s, a_n)Q_1[s, a_1, a_2] \\ &\geq \sum_{a_1 \ldots a_2} \pi_1'(s, a_1)\pi_2(s, a_n)Q_1[s, a_1, a_2] \end{aligned} \tag{9}$$

$$\begin{aligned} &\sum_{a_1 \ldots a_2} \pi_1(s, a_1)\pi_2(s, a_n)Q_1[s, a_1, a_2] \\ &\leq \sum_{a_1 \ldots a_2} \pi_1(s, a_1)\pi_2'(s, a_n)Q_1[s, a_1, a_2] \end{aligned} \tag{10}$$

The above satisfy the adversarial equilibrium condition.

1.6 Minimax Q-learning

This is another value function algo. The value function is given as

$$Val_1[s, Q_1] = \max_{\pi_1(s,.) \in \pi(A)} \min_{a_2 \in A_2} \sum_{a_1, a_2} \pi_1(s, a_1)Q_1[s, a_1, a_2] \tag{11}$$

The interpretation is that your opponent takes the worst possible action for you and you try to take the best you can do in that position.

$$Q_1[s, a_1, a_2] = (1 - \alpha)Q_1[s, a_1, a_2] + \alpha(r_1 + \beta Val_1(s, Q_1)) \tag{12}$$

1.6.1 Theorem 4

In a two player zero sum multiagent env, an agent following the minimax Q-learning update rule will converge to the optimal Q-function with prob 1. Furthermore, if the agent GLIEs and the limit equilibrium is unique, its behavior will converge with prob 1.

There are additional guarantees. Even if there isn't a unique limit equilibrium, a minimax GLIE Q agent converges to a policy that is optimal regardless of its opponent. The policy used in the minimax Q learning update rule achieves the largest value possible in the absence of the absence of the opponent's policy.

1.6.2 Notes

Minimax Q learning is found to be slow in practice. Folks have argued that a single agent Q learning may be better though it can be tricked into learning a suboptimal policy.

1.7 Coordination equilibrium

This is for cooperative games where agents are working towards a common goal. A coordination equilibrium in an n-player game is defined as one for which all agents achieve their max possible payoff. So, if policies are in coordination equilibrium, we've

$$\sum_{a_1 \ldots a_n} \pi_1(s, a_1) \ldots \pi(s, a_n) Q_i[s, a_1, \ldots a_m] = \\ = \max_{a_1 \ldots a_n} Q_i[s, a_1 \ldots a_n] \tag{13}$$

for all agents. If there is a coordination equilibrium, there is deterministic coordination equilibrium. This follows from the fact that the value function of every agent is a convex combination of values in the Q functions (?)

Again, no agent has any incentive to switch form the coordination equilibrium.

1.7.1 Theorem 5

In a multiagent env, an agent following the Nash Q learning update rule will converge to the optimal Q function. GLIE follows like usual.

Again, Nash Q is difficult to follow so we've a setup where this is achievable.

1.8 Team Markov Games

Again, we've a reward functions such that $R_1 == R_2$ for a given state and action pair. In this case, a reward received by an agent is received by every other agent. So, only 1 Q function is learned

1.8.1 Team Q learning

Define the value function as below.

$$Val_1[s, Q_1] = \max_{a_1 \ldots a_n} Q_1[s, a_1 \ldots a_n] \tag{14}$$

And the Q update rule is given by

$$Q_1[s, a_1 \ldots a_n] = (1 - \alpha) Q_1[s, a_1 \ldots a_n] + \alpha(r_1 + \beta Val_1(s, Q_1)) \tag{15}$$