Game Theoretical Foundations for Multi-agent Reinforcement Learning

Sanjana Gupta, Jinming Li, Chengcheng Li

University of Michigan

March 29, 2021

Overview

1. Introduction to MARL
2. Nash Q-Learning
3. Mathematical Programming with TD Error

Introduction to MARL

- MARL addresses the sequential decision-making problem of having **multiple agents** that operate in a common stochastic environment.
- Applications: Autonomous Driving and Traffic coordination [Campos-Rodriguez et al. (2017)]; Healthcare [Shakshuki and Reid (2015)]; Robotic Control [Kober et al. (2013)].

Introduction to MARL

- The main difference between MARL and SARL is that the evolution of the environmental state and the reward function that each agent receives are now influenced by the joint actions of all agents.

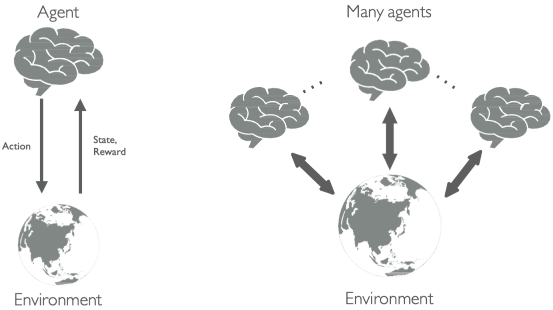


Figure: Credit: [Yang & Wang (2020)]

Stochastic Games

To describe a multi-agent system, we introduce the concept of Stochastic Game (MA-version of MDP). Denote $\mathcal{G} = (S, A^1, ..., A^n, r^1, ..., r^n, p)$:

- n: number of players (agents)
- S: the set of environment states
- A^i: the set of actions of player i. Denote $A = A^1 \times A^2 ... \times A^n$.
- $p : S \times A \to \Delta(S)$: the transition probability mapping.
- $r^i : S \times A \to \mathbb{R}$: the reward function of player i.

Stochastic Games

At each time step t, given environment state S_t,

- Each player chooses strategy $\sigma_t^i \in \Delta(A^i)$ simultaneously and independently.
- Each player plays action a_t^i according to σ_t^i.
- Each player gets reward $r_t^i = r^i(S_t, a_t^1, a_t^2, ...a_t^{i-1}, a_t^{i+1}, ..., a_t^n)$.

Some observations:

- Player i's reward is related to actions of all the other players.
- Player i's 'optimal' strategy should also involve thinking of what other players might do.

Game Theory

For the moment let's focus on game at each time step t, where each player tries to maximize the expected reward function. Game Theory provides solution concept as well as solver to analyze games at each time step t.

- Definition of Nash Equilibrium.
- Existence of Nash Equilibrium.
- Solve Nash Equilibrium from games.

Game Theory

Best Response

Given all other players' strategies $\sigma^{-i} = (\sigma^1, \sigma^2, ..., \sigma^{i-1}, \sigma^{i+1}, ..., \sigma^n)$, then σ^i is a best response for player i if

$$\mathbb{E}[r^i(\sigma^i; \sigma^{-i})] = \max_{\tilde{\sigma}^i \in \Delta(A^i)} \mathbb{E}[r^i(\tilde{\sigma}^i; \sigma^{-i})]$$

Nash Equilibrium

Joint strategy $\sigma = (\sigma^1, \sigma^2, ..., \sigma^n)$ is a Nash Equilibrium if every σ^i is a best response for player i.

Nash Equilibrium describes a situation where no player wants to 'deviate' from the currently adopted strategy.

Game Theory

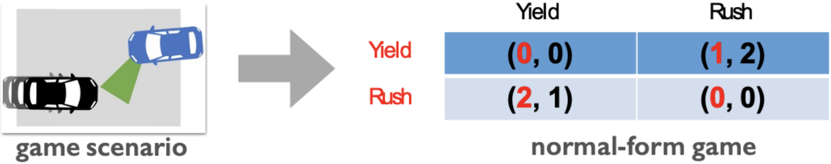
Existence of Nash Equilibrium

If action set A is finite, then there must exist (at least one) Nash Equilibrium.

- That means for each game at time step t, there exists an optimal strategy $\sigma = (\sigma^1, \sigma^2, ..., \sigma^n)$ for all players.
- Nash Equilibrium is not unique.

Game Theory

A simple example of traffic intersection [Yang & Wang (2020)]:



	Yield	Rush
Yield	(0, 0)	(1, 2)
Rush	(2, 1)	(0, 0)

- Nash Equilibria can be solved by finding best response for each player.
- Two Nash Equilibria are: (Yield, Rush) and (Rush, Yield).

Challenges of Solving NE

However, in general solving Nash Equilibrium for can be computationally challenging. Alternatively researchers analyze particular types of games:

- Two-player games
- Zero-sum games
- General-sum games

1. Introduction to MARL
2. Nash Q-Learning
3. Mathematical Programming with TD Error

From SA to MA

- Q-learning is the most commonly used and well-established model-free method in SARL.
- Difficulty from SA to MA Q-learning:
    - The environment in MARL consists of other agents who are similarly adapting, thus the environment is no longer stationary, and the familiar theoretical guarantees no longer apply.
    - Non-stationarity of the environment is not generated by an arbitrary stochastic process, but rather by other agents, who might be presumed rational or at least regular in some ways.

Nash Q-Values

Definition (Nash Q-Function)

Agent i's Nash Q-function is defined over $(s, a^1, ..., a^n)$, as the sum of Agent i's current reward plus its future rewards when all agents follow a joint Nash equilibrium strategy. That is

$$\begin{aligned} Q_*^i(s, a^1, ..., a^n) &= r^i(s, a^1, ..., a^n) \\ &\quad + \beta \sum_{s' \in S} P(s' \mid s, a^1, ..., a^n) v^i(s', \pi_*^1, ..., \pi_*^n). \end{aligned}$$

Differences between MARL Q-learning and SARL Q-learning

- Joint actions VS individual actions ($Q(s, a^1, ..., a^n)$ VS $Q(s, a)$).
- Optimal Q-value: Current reward + Future rewards when all agents play specified Nash equilibrium strategies from the next period onward VS current reward + future rewards by playing the optimal strategy from the next period onward.

Nash Q-Learning Algorithm

- In Nash Q-learning, the agent must observe both its own and other agents' rewards.

Definition (Stage Game)

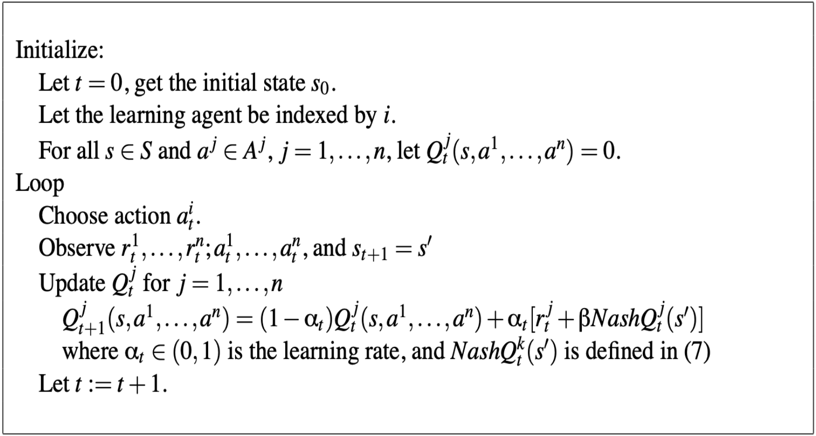
An n-player stage game is defined as $(M^1, ..., M^n)$, where for $k = 1, ..., n$, $M^k = \{r^k(a^1, ..., a^n) \mid a^1 \in A^1, ..., a^n \in A^n\}$.

Let σ^{-k} be the product of strategies of all agents other than k.

Definition (NE in Stage Game)

A joint strategy $(\sigma^1, ..., \sigma^n)$ constitutes a Nash equilibrium for the stage game $(M^1, ..., M^n)$ if, for $k = 1, .., n$, $\sigma^k \sigma^{-k} M^k \geq \hat{\sigma}^{k-k} M^k$ for all $\sigma^k \in \hat{\sigma}(A^k)$.

Nash Q-Learning Algorithm



```
Initialize:
    Let $t = 0$, get the initial state $s_0$.
    Let the learning agent be indexed by $i$.
    For all $s \in S$ and $a^j \in A^j$, $j = 1, \ldots, n$, let $Q_t^j(s, a^1, \ldots, a^n) = 0$.
Loop
    Choose action $a_t^i$.
    Observe $r_t^1, \ldots, r_t^n; a_t^1, \ldots, a_t^n$, and $s_{t+1} = s'$
    Update $Q_t^j$ for $j = 1, \ldots, n$
        $Q_{t+1}^j(s, a^1, \ldots, a^n) = (1 - \alpha_t) Q_t^j(s, a^1, \ldots, a^n) + \alpha_t [r_t^j + \beta NashQ_t^j(s')]$
        where $\alpha_t \in (0, 1)$ is the learning rate, and $NashQ_t^k(s')$ is defined in (7)
    Let $t := t + 1$.
```

where $\text{Nash}Q_t^k(s') = \pi^1(s') \ldots \pi^n(s') \cdot Q_t^k(s')$.

Complexity of Nash Q-Learning Algorithm

- Let $|S|$ be the number of states, and let $|A_i|$ be the size of agent i's action space A_i. Assuming $|A^1| = ... = |A^n| = |A|$, the total number of entries in Q^k is $|S| \cdot |A|^n$.

- The learning agent has to maintain n Q-tables, the total space requirement is $n|S| \cdot |A|^n$.

- The time complexity is dominated by the calculation of Nash equilibrium used in the Q-function update. The complexity of finding an equilibrium in matrix games is unknown.

- Commonly used algorithms for 2-player games have exponential worst-case behavior, and approximate methods are typically employed for n-player games (McKelvey and McLennan, 1996).

Convergence of Nash Q-Learning Algorithm

Definition (Global Optima of Stage Games)

A joint strategy $(\sigma^1, ..., \sigma^n)$ of the stage game $(M^1, ..., M^n)$ is a global optimal point if for all k, $\sigma M^k \geq \hat{\sigma} M^k \quad \text{for all} \quad \hat{\sigma} \in \sigma(A)$.

Note that a global optimal point is always a Nash equilibrium.

Definition (Saddle Point of Stage Game)

A joint strategy $(\sigma^1, ..., \sigma^n)$ of the stage game $(M^1, ..., M^n)$ is a saddle point if for all k,

$$\sigma^k \sigma^{-k} M^k \geq \hat{\sigma}^k \sigma^{-k} M^k \quad \text{for all} \quad \hat{\sigma}^k \in \sigma(A^k).$$
$$\sigma^k \sigma^{-k} M^k \leq \sigma^k \hat{\sigma}^{-k} M^k \quad \text{for all} \quad \hat{\sigma}^{-k} \in \sigma(A^{-k}).$$

Note that all saddle points of a stage game are equivalent in their values.

Convergence of Nash Q-Learning Algorithm

The convergence results requires the following three assumptions.

1. Every state $s \in S$ and action $a^k \in A^k$ for $k = 1, ..., n$, are visited infinitely often.
2. The learning rate α satisfies the following conditions for all $s, t, a^1, ..., a^n$:
   (a) $0 \leq \alpha(s, a^1, ..., a^n) < 1$, $\sum_{t=1}^{\infty} \alpha_t(s, a^1, ..., a^n) = \infty$, $\sum_{t=0}^{\infty} [\alpha_t(s, a^1, ..., a^n)]^2 < \infty$, and the latter two hold uniformly and with probability 1.

   (b) $\alpha_t(s, a^1, ..., a^n) = 0$ if $(s, a^1, ..., a^n) \neq (s, a^1, ..., a^n)$.
3. One of the following holds during training: (A). Every stage game $(Q_t^1(s), ..., Q_t^n(s))$, for all t and s, has a global optimal point, and agents' payoffs in this equilibrium are used to update their Q-functions. (B). Every stage game $(Q_t^1(s), ..., Q_t^n(s))$, for all t and s, has a saddle point, and agents' payoffs in this equilibrium are used to update their Q-functions.

Convergence of Nash Q-Learning Algorithm

Theorem

Under Assumptions 1–3, *the sequence* $Q = (Q_t^1, ..., Q_t^n)$, *updated by*

$$Q_{t+1}^k(s, a^1, .., a^n) = (1 - \alpha_t)Q_t^k(s, a^1, .., a^n) + \\ \alpha_t\Big(r_t^k + \beta\pi^1(s')...\pi^n(s')Q_t^k(s')\Big), \quad k = 1, ..., n.$$

where $(\pi^1(s), ..., \pi^n(s))$ *is the appropriate type of Nash equilibrium solution for the stage game* $(Q_t^1(s), ..., Q_t^n(s))$, *converges to the Nash* Q*-value* $Q_* = (Q_*^1, ..., Q_*^n)$.

The convergence proof relies on fixed point theorem by building a contraction map that can converge to Q_*.

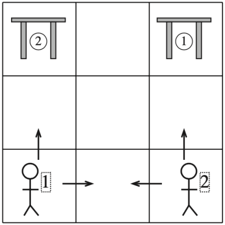
Discussions on the Convergence of Nash Q-Learning

- The proof crucially depends on the restriction on stage games during learning because otherwise the Nash equilibrium operator is in general not a contraction operator.

- In general, it would be easy for Assumption 3 to be violated in reality. Suppose we start with an initial stage game that satisfies the assumption, $Q_0^i(s, a^1, a^2) = 0$, for all s, a^1, a^2 and $i = 1, 2$ and the stage game has both a global optimal point and a saddle point. During learning, elements of Q_0^1 are updated asynchronously, thus the property would not be preserved for (Q_t^1, Q_t^2).

- However, numerical experiments have shown that the convergence is not sensitive to properties of the stage games during learning.

Experiments in Grid-World Games



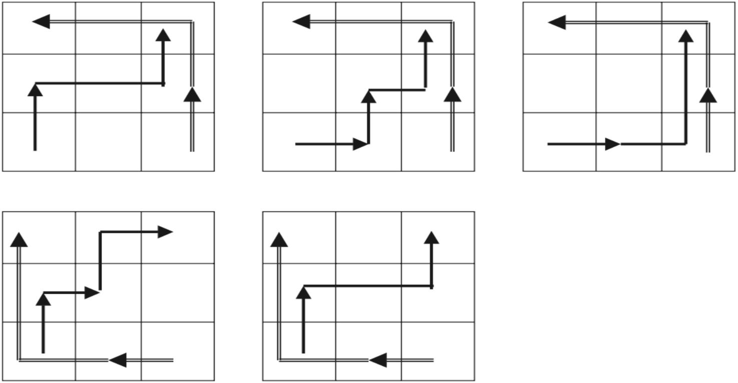
6	7	8
3	4	5
0	1	2

- $A_i = \{\text{Left, Right, Down, Up}\}$, $S = \{(0,1),(0,2),...,(8,7)\}$, where a state $s = (l_1, l_2)$ represents the agents' joint location.
- If two agents attempt to move into the same cell (excluding a goal cell), they are bounced back to their previous cells.
- Let $L(l, a)$ be the potential new location resulting from choosing action a in position l, The reward function is, for $i = 1, 2$,

$$r_t^i = \begin{cases} 100 & \text{if} \quad L(l_t^i, a_t^i) = \text{Goal}_i \\ -1 & \text{if} \quad L(l_t^1, a_t^1) = L(l_t^2, a_t^2) \quad \text{and} \quad L(l_t^2, a_t^2) \neq \text{Goal}_j, j = 1, 2 \\ 0 & \text{otherwise.} \end{cases}$$

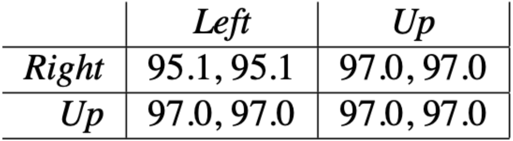
Nash *Q*-Values

- Note that the policy defines a path, that is, a sequence of locations from the starting position to the final destination.
- Two shortest paths that do not interfere with each other constitute a Nash equilibrium, since each path (strategy) is a best response to the other.



Theoretical Nash Q-Values

- The value of the game for agent 1 is defined as its accumulated reward when both agents follow their Nash equilibrium strategies, $v^1(s_0) = \sum_t \beta^t E[r_t \mid \pi_*^1, \pi_*^2, s_0]$.
- With initial state $s_0 = (02)$, and $\beta = 0.99$, $v_1(s_0) = 0 + 0.990 + 0.9920 + 0.993 \cdot 100 = 97$.



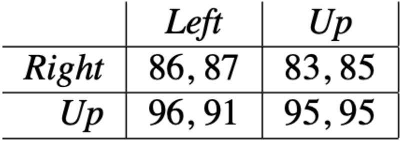
	Left	*Up*
Right	95.1, 95.1	97.0, 97.0
Up	97.0, 97.0	97.0, 97.0

Learning Process

- The learning agent 1 initializes $Q^1(s, a^1, a^2) = 0$ and $Q^2(s, a^1, a^2) = 0$ for all s, a^1, a^2.
- A game starts from the initial state $(0, 2)$. After observing the current state, agents act simultaneously. They then observe the new state, both agents' rewards, and actions.
- When at least one agent moves into its goal position, the game restarts. In new episode, each agent start at a random position.
- The training stops after 5000 episodes. Each episode on average takes eight steps. So one experiment usually requires 40,000 steps.

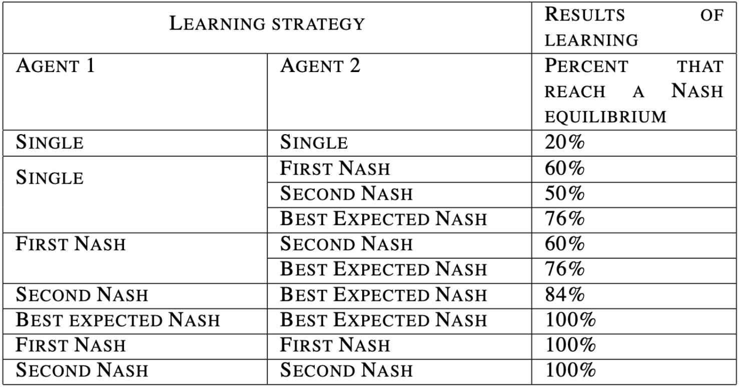
Learned Q Values

- The total number of state-action tuples is 424. Thus each tuple is visited 95 times on average.
- Learning rate is defined as the inverse of the number of visits, $\alpha_t(s, a^1, a^2) = 1/(n_t(s, a^1, a^2))$, where $n(s, a^1, a^2)$ is the number of times the tuple (s, a^1, a^2) has been visited.
- Results in table below are close to the theoretical derivations.



	Left	*Up*
Right	86, 87	83, 85
Up	96, 91	95, 95

Learning Performance



Learning strategy		Results of learning
Agent 1	Agent 2	Percent that reach a Nash equilibrium
Single	Single	20%
Single	First Nash	60%
	Second Nash	50%
	Best Expected Nash	76%
First Nash	Second Nash	60%
	Best Expected Nash	76%
Second Nash	Best Expected Nash	84%
Best expected Nash	Best Expected Nash	100%
First Nash	First Nash	100%
Second Nash	Second Nash	100%

1. Introduction to MARL
2. Nash Q-Learning
3. Mathematical Programming with TD Error

TD error in SARL

- Recall that in SARL, at time $t \in \mathbb{R}$

$$V(X_t) \sim R_t + \gamma V(X_{t+1})$$

- Substituting estimates of $V(X.)$, the TD error is

$$\delta_{t+1} = R_t + \gamma \hat{V}(X_{t+1}) - \hat{V}(X_t)$$

TD error in MARL

- Define the probability transition matrix induced by policy π

$$p(s'|s,\pi) = \sum_{a^1 \in A^1} \sum_{a^2 \in A^2} \cdots \sum_{a^n \in A^n} p(s'|s,a^1,\cdots,a^n) \prod_{k=1}^{n} \pi^k_{sa^k}$$

  where $\pi^k_{sh} \in \mathbb{R}$ is the probability assigned by agent k to action $h \in A^k$.
- TD Error at state s is defined as

$$\delta_s = \sum_{k=1}^{n} v^k(s,\pi) - r^k(s,\pi) - \beta \sum_{s' \in S} P(s'|s,\pi) v^k(s',\pi)$$

- Minimize TD error jointly over policy π and $v^\cdot$.

TD error in MARL: Optimization problem

- Matrix form of the optimization problem for $n = 2$ players

$$\min_{z=((V^1)^T,(V^2)^T,\pi^1,(\pi^2)^T)} f(z) = \sum_{k=1}^{2} \mathbf{1}_{|S|}^T \Big[V^k - \Big(R^k(\pi) + \beta P(\pi) V^k \Big) \Big]$$

$$\begin{array}{rrl} & \text{(a)} & \pi^2(s)^T \left[R^1(s) + \beta \sum_{s'} P(s'|s) v^1(s') \right] \le v^1(s) \mathbf{1}_{|A^1|}^T, \quad \forall s \in S \\ \text{s.t.} & \text{(b)} & \left[R^2(s) + \beta \sum_{s'} P(s'|s) v^2(s') \right] \pi^1(s) \le v^2(s) \mathbf{1}_{|A^2|}, \quad \forall s \in S \\ & \text{(c)} & \pi^1(s) \ge 0, \quad \pi^1(s)^T \mathbf{1}_{|A^1|} = 1, \quad \forall s \in S \\ & \text{(d)} & \pi^2(s) \ge 0, \quad \pi^2(s)^T \mathbf{1}_{|A^2|} = 1, \quad \forall s \in S \end{array}$$

- $\mathrm{R}^1(s) = [R^k(s, a^1, a^2)]_{a^1,a^2}$ is the reward matrix for agent k in state s. $\mathrm{R}^k(\pi) = \left\langle \pi^2(s)^T \mathrm{R}^k(\mathrm{s}) \pi^1(\mathrm{s}) : \mathrm{s} \in \mathrm{S} \right\rangle$ is the expected reward vector over all states under joint policy π.

- $V^k = [v^k(s) : s \in S]; \qquad P(s'|s) = [P(s'|s, a^1, a^2)]_{a^1,a^2} \in \mathbb{R}^{|A^2| \times |A^1|}$

TD error in MARL: Optimization problem

Theorem

Consider a point $\hat{z}^T = ((\hat{V}^1)^T, (\hat{V}^2)^T, \hat{\pi}^1, (\hat{\pi}^2)^T)$.
Then the strategy part $(\hat{\pi}^1, \hat{\pi}^2)$ *of* $\hat{z}^T$ *forms a (Nash) equilibrium point of the general-sum discounted game, if and only if* $\hat{z}$ *is the global minimum of the optimization problem with* $f(z) = 0$.

Corollary

Let $\hat{z}$ *be feasible with an objective function value* $f(z) = \gamma > 0$. *Then* $(\hat{\pi}^1, \hat{\pi}^2)$, *the strategy part of* $\hat{z}$ *forms an* ϵ*-equilibrium with* $\epsilon <= \frac{\gamma}{1-\beta}$.

Optimization problem: challenges

- Problem is non-linear as constraints (a), (b) are quadratic in V & π.
- The feasible region is non-convex.
- Only the global optimum corresponds to the NE of Stochastic games, while the common gradient-descent type of methods can only guarantee convergence to a local minimum.
- All methods (including the formulations below) are tractable in small SGs with only 2 players and at most tens of states.

Other formulations

- There are other formulations of mathematical programs to solve MARL, eg via Bellman dynamic programming.
- [Murray & Gordon (2007)] Extend the Bellman equation to MARL providing the exact form of the feasible set.
- [Dermed & Isbell (2009)] Extend this by formulating it as a multi-objective linear program. They provide convergence guarantees.
- Unfortunately, these algorithms can not run with more than 4 agents, but can accommodate many states.

Summary

MARL is different from SARL as players need to consider other players' behaviours. In this presentation, we have talked about:

- Stochastic Games
- Solution concept from Game Theory perspective
- Learning Algorithms (Nash Q-learning and Minimizing TD-Error)
- Challenges of MARL from a Game Theory perspective

References

Hu & Wellman (2003)
Nash Q-learning for general-sum stochastic games.
Journal of machine learning research (4) 1039–1069.

Mac Dermed, L., & Isbell Jr, C. L.
Solving Stochastic Games.
NIPS (pp. 1186-1194).

Murray, C., Gordon, G.
Finding correlated equilibria in general sum stochastic games.
Carnegie Mellon University, School of Computer Science, Machine Learning Department.

Yang & Wang (2020)
An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective
arXiv preprint 12(3)2011.00583.

The End