Mathematics Letters
2025, Vol. 11, No. 3, pp. 66-70
https://doi.org/10.11648/j.ml.20251103.12

SciencePG
Science Publishing Group

Research Article

Use of Reinforcement Learning to Gain the Nash Equilibrium

Reza Habibi*

Deaprtment of Banking, Iran Banking Institute, Tehran, Iran

Abstract

Reinforcement learning (RL) is a type of machine learning where an agent learns optimal behavior through interaction with its environment. It is a machine learning training method that trains software to make certain desired actions. Nash equilibrium (SNE) is a combination of actions of the different players, in which no coalition of players can cooperatively deviate. Each player chooses the best strategy among all options. Nash equilibrium occurs when each player knows the strategy of their opponent and uses that knowledge. Nash equilibrium occurs in non-cooperative games when two players have optimal game strategies such that no matter how they change their strategy. This paper explores the application of reinforcement learning algorithms within the domain of game theory, with a particular focus on their convergence properties toward Nash equilibrium. We analyze q-learning approach in 2-agent environments, highlighting their capacity to learn optimal strategies through iterative interactions. Our theoretical investigation examines the conditions under which these algorithms converge to Nash equilibrium, considering factors such as learning rate schedules. The insights gained contribute to a deeper understanding of how reinforcement learning can serve as a powerful tool for equilibrium computation in complex strategic environments, paving the way for advanced applications in economics, automated negotiations, and autonomous systems.

Keywords

Q-learning, Nash Equilibrium, Game Theory, Reinforcement Learning

1. Introduction

In the field of game theory, learning in repeated games refers to the process through which players adapt their strategies over multiple iterations of a game based on past experiences and outcomes. In repeated games, players engage in the same game multiple times, allowing them to observe the actions of their opponents and modify their own strategies accordingly. Learning in repeated games contains the strategy adjustment which means players may change their strategies based on the results of previous rounds. This could involve being more cooperative if they perceive that it leads to better overall outcomes or becoming more competitive if they feel that cooperation is being exploited.

Some other concepts are reputation effects, evolutionary learning, folk theorem and communication and coordination. Overall, learning in repeated games emphasizes the dynamic nature of strategic decision-making, where past interactions inform future choices, and players continuously adapt to the behaviors of others, [5]. The learning algorithms are too important and indicate that players might use specific algorithms or heuristics to guide their decisions, such as fictitious

*Corresponding author: habibi1356@gmail.com (Reza Habibi), habibi1356@yahoo.com (Reza Habibi)

Received: 29 August 2025; **Accepted:** 13 October 2025; **Published:** 31 October 2025

play. One of these algorithms is the reinforcement learning. It is a type of machine learning where an agent learns to make decisions by interacting with an environment.

The goal of the agent is to maximize a cumulative reward over time through trial and error. It has been successfully applied in various fields, including robotics, recommendation systems, and autonomous vehicles, to name a few. It's particularly powerful in situations where the optimal solution is difficult to define or where the agent must balance exploration (trying new things) and exploitation (using known strategies), see [7, 14] and references therein. Reinforcement learning contains some components like agent, environment, state, action, reward feedback, and value function. The core process of reinforcement learning involves the agent observing the current state of the environment, taking an action based on its policy, receiving a reward, and then updating its policy to improve future actions. This cycle continues, allowing the agent to learn and adapt over time, see [7, 12].

Various types of reinforcement learning approaches are applied to repeated games. Most of them are presented as multi-agent reinforcement learning setting; see [4, 10] and references therein. To review a few, authors [1, 15] studied the presence of multiple Nash equilibrium in general sum games and proposed the q-learning method and studied its convergence to Nash equilibriums. Authors [3] introduced the individual q-learning reinforcement learning algorithm and using the stochastic approximation technique studied its asymptotic behavior, showed that strategies will converge to Nash equilibriums almost surely in almost all 2-player games. Author [2, 11] applied the reinforcement learning applications in poker game which is a multi-player extensive form game with incomplete information. For comprehensive review in this field, see [4] and [6, 9].

Here, we use the notations and results of [3, 8] for two-player game where payoffs (rewards) are presented in m*m bi-matrix. First, suppose that $m = 2$ and consider the following matrix

Table 1. *Payoff Matrix.*

	L	**R**
L	(a_{11}, b_{11})	(a_{12}, b_{12})
R	(a_{21}, b_{21})	(a_{22}, b_{22})

Let α_t (β_t) be the probability of choosing L by raw (column) player; i.e., player 1 (player 2). Let

$$\pi_{1t} = (\alpha_t, 1-\alpha_t)^T$$

$$\pi_{2t} = (\beta_t, 1-\beta_t)^T$$

where $t = t_n \in [0, \infty), n = 1, \ldots, N$.

The payoff (reward) matrix of players 1 and 2 are A, B, respectively,

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix},$$

The rest of paper is organized as follows. The q-learning procedure is given in the next section. The matching the pennies game is studied in section 4. Section 4 concludes.

2. Q-learning Procedure

Following [3, 13], let

$$q_i(t) = (q_{iL}(t), q_{iR}(t))^T, i = 1, 2$$

be the vector of q- value functions for i-th player, $i = 1, 2$, at time t. Following their Lemma 1, it is seen that

$$\frac{dq_{1L}}{dt} = a_{11}\beta_t + a_{12}(1-\beta_t) - q_{1L}$$

$$\frac{dq_{1R}}{dt} = a_{21}\beta_t + a_{22}(1-\beta_t) - q_{1R}$$

In the matrix form, it is seen that

$$\frac{d}{dt}q_1(t) = A\pi_{2t} - q_1(t).$$

Similarly, it is seen that

$$\frac{d}{dt}q_2(t) = B\pi_{1t} - q_2(t).$$

It is easy to see that these results are true for $m > 2$. The above equation has following solution as

$$q_1(t) = A\int_0^t e^{-(t-s)}\pi_{2s}ds + q_1(0)e^{-t},$$

where assuming $q_1(0) = 0$, then

$$q_1(t) = A\int_0^t e^{-(t-s)}\pi_{2s}ds.$$

Similarly, assuming $q_2(0) = 0$, then

$$q_2(t) = B\int_0^t e^{-(t-s)}\pi_{1s}ds.$$

Following [3], for updating procedure, let

$$\alpha_n = \alpha_{t_n}$$

and τ be the temperature parameter. For $m = 2$, using the Boltzmann action selection and representing it, in the $logit$ form, it is seen that

$$\text{logit}(\alpha_{n+1}) := \log\left(\frac{\alpha_{n+1}}{1-\alpha_{n+1}}\right) = \frac{1}{\tau}1^T q_1(t_n),$$

$$\text{logit}(\beta_{n+1}) = \frac{1}{\tau}1^T q_2(t_n),$$

where $1^T = (1 \quad -1)$. Then,

$$\text{logit}(\alpha_{n+1}) = \frac{1}{\tau}1^T A\int_0^{t_n} e^{-(t_n-s)}\pi_{2s}ds,$$

$$\text{logit}(\beta_{n+1}) = \frac{1}{\tau}1^T B\int_0^{t_n} e^{-(t_n-s)}\pi_{1s}ds.$$

The following proposition summarizes the above discussion:

Proposition 1. For two-player game with payoff matrices A, B, for player 1 and 2, respectively, then the updating procedure for α_n, β_n are given by

$$\text{logit}(\alpha_{n+1}) = \frac{1}{\tau}1^T A\int_0^{t_n} e^{-(t_n-s)}\pi_{2s}ds,$$

$$\text{logit}(\beta_{n+1}) = \frac{1}{\tau}1^T B\int_0^{t_n} e^{-(t_n-s)}\pi_{1s}ds.$$

3. Matching the Pennies

For the matching the pennies, we have $A = -B = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$, $1^T A = -1^T B = 21^T$. Then,

$$\text{logit}(\alpha_{n+1}) = \frac{2}{\tau}\int_0^{t_n} e^{-(t_n-s)}1^T\pi_{2s}ds$$

$$= \frac{2}{\tau}\int_0^{t_n} e^{-(t_n-s)}(2\beta_s - 1)ds,$$

$$\text{logit}(\beta_{n+1}) = \frac{2}{\tau}\int_0^{t_n} e^{-(t_n-s)}(1-2\alpha_s)ds.$$

3.1. Nash Convergence

Hereafter, the convergence to Nash equilibrium is studied. To this end, let

$$\theta_n = \text{logit}(\alpha_n).$$

It is easy to see that

$$|e^{t_n}\theta_{n+1} - e^{t_{n-1}}\theta_n| = \frac{2}{\tau}\int_{t_{n-1}}^{t_n} e^s|2\beta_s - 1|ds \leq$$

$$\leq \frac{2}{\tau}\int_{t_{n-1}}^{t_n} e^s ds = \frac{2}{\tau}(e^{t_n} - e^{t_{n-1}}).$$

Let $\delta_n = t_n - t_{n-1} > 0$. Thus,

$$\left|\theta_{n+1} - e^{-\delta_n}\theta_n\right| \leq \frac{2}{\tau}\left(1 - e^{-\delta_n}\right).$$

Therefore,

$$\left|\theta_{n+1} - \theta_n + (1 - e^{-\delta_n})\theta_n\right| \leq \frac{2}{\tau}\left(1 - e^{-\delta_n}\right).$$

As δ_n tends to zero, and $1 - e^{-\delta_n} \leq \frac{\tau}{2}$, then

$$|\theta_{n+1} - \theta_n| \to 0.$$

Since the $logit$ is a continous function, thus

$$|\alpha_{n+1} - \alpha_n| \to 0.$$

The following proposition summarizes the above discussion where proof is omitted.

Proposition 2. Assuming $1 - e^{-\delta_n} \leq \frac{\tau}{2}$, then as δ_n tends to zero, then

$$|\alpha_{n+1} - \alpha_n| \to 0.$$

3.2. Recursive Relation

Here, using simplifying assumption, computational complexity is derived. Assuming

$$\beta_s = \beta_{j-1} \text{ for } s \in [t_{j-1}, t_j),$$

then

$$\theta_{n+1} = \frac{2}{\tau}\sum_{j=1}^{n}\int_{t_{j-1}}^{t_j} e^s(2\beta_{j-1} - 1)ds =$$

$$= \frac{2}{\tau}\sum_{j=1}^{n}\left(2\beta_{j-1} - 1\right)(e^{t_j} - e^{t_{j-1}}).$$

Assuming $t_j - t_{j-1} = \delta_n, j \leq n$, then

$$\theta_{n+1} = e^{-\delta_n}\theta_n + \frac{(1-e^{-\delta_n})}{\tau}(\beta_{n-1} - 0.5).$$

The following proposition generalizes the above discussion:

Proposition 3. For general $2 * 2$ payoff bi-matrix A, then

$$\theta_{n+1} = e^{-\delta_n}\theta_n + \frac{(1-e^{-\delta_n})}{\tau}(l_1\beta_{n-1} + l_2),$$

where

$$l_1 = a_{11} + a_{22} - (a_{12} + a_{21})$$

$$l_2 = a_{12} - a_{22}.$$

Remark 1. Notice that

$$|\theta_{n+1} - \theta_n| = \left(1 - e^{-\delta_n}\right)\left|\frac{l_1\beta_{n-1} + l_2}{\tau} - \theta_n\right|,$$

where assuming $1 - e^{-\delta_n} \leq \frac{\tau}{2}$, again $|\theta_{n+1} - \theta_n| \to 0$ is derived. Let $\zeta_n = \text{logit}(\beta_n)$. Also, notice that

$$2\beta_{n-1} - 1 = \frac{e^{\zeta_{n-1}-1}}{e^{\zeta_{n-1}+1}}.$$

3.3. Simulations

For matching the pennies game, let $\delta_n = 0.1, \tau = 0.2, \alpha_0 = 0.3, \beta_{0.} = 0.8$. The blue and red lines are α_n, β_n, respectively.

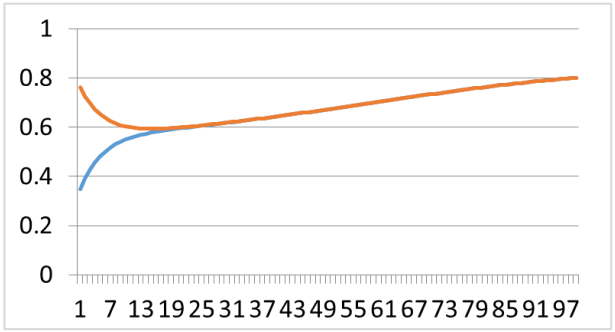

Figure 1. *Convergences of α_n, β_n.*

Remark 2. For example, for the prisoner's dilemma game with $A = \begin{bmatrix} -2 & -10 \\ -1 & -5 \end{bmatrix}$, then, $l_1 = 4$, $l_2 = -5$ and

$$\theta_{n+1} = e^{-\delta_n}\theta_n + \frac{(1-e^{-\delta_n})}{\tau}(4\beta_{n-1} - 5).$$

3.4. Comparisons

Here, the rate of convergence of reinforcement learning and gradient descent methods are compared in the matching the pennies game. The gradient descent sequential updating procedure is given by

$$|\alpha_{n+1} - \alpha_n| = 4|\beta_n - 0.5|\zeta,$$

where ζ is the step parameter, see [6]. For the reinforcement learning, it is seen that

$$|\theta_{n+1} - \theta_n| = (1 - e^{-\delta_n})|\frac{4}{\tau}(\beta_n - 0.5) - \theta_n| \cong,$$

$$4|\beta_n - 0.5|\frac{(1-e^{-\delta_n})}{\tau}.$$

Since the $logit$ function is a one to one function, therefore, it is enough to compare ζ and $\frac{(1-e^{-\delta_n})}{\tau}$. As

$$\tau\zeta \leq 1 - e^{-\delta_n},$$

then, the gradient descent method has faster rate of convergence.

4. Conclusions

In conclusion, the exploration of reinforcement learning (RL) within the framework of game theory has yielded promising advancements in both fields. RL algorithms, with their ability to learn optimal strategies through trial and error, provide a powerful tool for addressing the complexities inherent in multi-agent environments. This paper has showcased the diverse applications of RL in game theory, from approximating Nash equilibrium in complex games to designing adaptive agents capable of outperforming traditional game-theoretic approaches.

However, the integration of RL and game theory is not without its challenges. The non-stationary of multi-agent environments, the curse of dimensionality in large games, and the difficulty in ensuring convergence to desirable equilibrium concepts remain significant hurdles. Future research should focus on developing more robust and scalable RL algorithms specifically tailored for game-theoretic settings. This includes exploring techniques such as multi-agent learning with communication, hierarchical RL for complex strategy spaces, and the development of novel exploration-exploitation strategies designed to navigate the challenges of interacting with dynamically changing opponents.

Ultimately, the synergy between reinforcement learning and game theory holds immense potential for understanding and influencing strategic interactions in a wide range of real-world domains, including economics, robotics, and social science. By continuing to refine and expand the theoretical foundations and practical applications of this interdisciplinary field, we can unlock new possibilities for creating intelligent, cooperative, and adaptive systems capable of thriving in complex multi-agent environments. This ongoing research will not only advance our understanding of strategic decision-making but also pave the way for the development of more intelligent and autonomous agents capable of navigating the complexities of the world around us.

Abbreviations

Logit	Logit Function

Author Contributions

Reza Habibi is the sole author. The author read and approved the final manuscript.

Conflicts of Interest

The author declares no conflicts of Interest.

References

[1] Ballard, D. and Zhu, S. (2022). Overcoming non-stationary in un-communication learning. In: *International Conference on Machine Learning 2002*, 354-363.

[2] Brown, N. and Sandholm, T. (2017). Dynamic threshold and pruning for regret minimization. In: *International Conference in Machine Learning 2017*, 793-802.

[3] Collin-Dufresne, P. and Fos, V. (2012). Insider trading, stochastic liquidity and equilibrium prices. Technical report. Columbia University and University of Illinois.

[4] Cristofol, M. and Roques, L. (2016). Simultaneous determination of the drift and diffusion coefficients in stochastic differential equations. Technical report. Institut de Math ematiques de Marseille, France.

[5] Gyungmin, P. (2022). Insider trading, stock volatility and market liquidity in the Korean capital market. Studies in Business and Economics 17, 175-189.

[6] Harris, L. (1998). Optimal dynamic trading in the presence of insider information. Journal of Financial Markets 1, 123-148.

[7] Kyle, A. S. (1985). Continuous auctions and insider trading. Econom terica 53, 1315-1335.

[8] Leslie, D. S., and Collins, E. J. (2025). Individual q-learning in normal form games. *SIAM Journal on Control and Optimization* 44, 1-20.

[9] Mailath, G. & Samuelson, L. (2016). *Repeated games and reputations: long-run relationships*. Oxford University Press. USA.

[10] Morris, S. and Shin, H. S. (1998). Unique equilibrium in a model of self-fulfilling currency attacks. American Economic Review 88, 587–597.

[11] Nguyen, T. T., Nguyen, N. D., Nahavandi, S. (2020). Deep reinforcement learning for multi-agent systems: a review of challenges, solutions and applications. *IEEE Transactions on Cybernetics* 20, 3826-3839.

[12] Osborne, M. and A. Rubinstein, A. (1994). A course in game theory. Cambridge. MIT Press.

[13] Rahman, M., & Mollah, M. (2019). Mathematical modeling of insider trading: a game theoretical approach. Journal of Risk and Financial Management 12, 138- 158.

[14] Singh, S., Kearns, M., and Mansour, Y. (2013). Nash convergence of gradient dynamics in general-sum games. *Technical Report*. AT&T Labs and Tel Aviv University.

[15] Sutton, R. S., and Barto, A. G. (2009). *Reinforcement learning: an introduction*. MIT Press. Cambridge. UK.