Learning Nash Equilibria in Zero-Sum Stochastic Games via Entropy-Regularized Policy Approximation

Yue Guan Qifan Zhang Panagiotis Tsiotras
Georgia Institute of Technology

Background

Two-agent zero-sum stochastic game:

- A tuple $\langle \mathcal{S}, \mathcal{A}^{\text{pl}}, \mathcal{A}^{\text{op}}, \mathcal{T}, \mathcal{R}, \gamma \rangle$
- The Player maximizes; the Opponent minimizes

Policy π^{pl} (π^{op}): mapping from $\mathcal{S}$ to $\mathcal{A}^{\text{pl}}$ ($\mathcal{A}^{\text{op}}$)

***Q*-function and value associated with** $\pi = (\pi^{\text{pl}}, \pi^{\text{op}})$:

$$\mathcal{Q}^{\pi}(s, a^{\text{pl}}, a^{\text{op}}) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(S_t, A_t^{\text{pl}}, A_t^{\text{op}}) \,|\, S_0 = s, A_0^{\text{pl}} = a^{\text{pl}}, A_0^{\text{op}} = a^{\text{op}}\right]$$

$$\mathcal{V}^{\pi^{\text{pl}},\pi^{\text{op}}}(s) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(S_t, A_t^{\text{pl}}, A_t^{\text{op}}) \,|\, S_0 = s\right]$$

Nash equilibrium:

- A coupled max-min optimization to find $(\pi^{\text{pl}*}, \pi^{\text{op}*})$:

$$\mathcal{V}^{\pi^{\text{pl}*},\pi^{\text{op}*}} = \max_{\pi^{\text{pl}}} \min_{\pi^{\text{op}}} \mathcal{V}^{\pi^{\text{pl}},\pi^{\text{op}}}$$

- Solved as Linear Programs at each state. **Expensive** to solve.

$$\max\ v \quad \text{subject to} \quad v\mathbf{1}^{\top} - \boldsymbol{\pi}^{\text{pl}}(s)\boldsymbol{\mathcal{Q}}^{\pi}(s) \le 0,\quad \mathbf{1}^{\top}\boldsymbol{\pi}^{\text{pl}}(s) = 1,\ \boldsymbol{\pi}^{\text{pl}}(s) \ge 0$$

$$\min\ u \quad \text{subject to} \quad u\mathbf{1} - \boldsymbol{\mathcal{Q}}^{\pi}(s)\boldsymbol{\pi}^{\text{op}^{\top}}(s) \le 0,\quad \mathbf{1}^{\top}\boldsymbol{\pi}^{\text{op}}(s) = 1,\ \boldsymbol{\pi}^{\text{op}}(s) \ge 0$$

Shapley's method (minimax-Q) iterates between two operators:

$$\left(\pi^{\text{pl}}_{\text{Nash}}, \pi^{\text{op}}_{\text{Nash}}\right) = \Gamma_{\text{Nash}}(\mathcal{Q}); \qquad \mathcal{Q} = \Gamma_1\left(\mathcal{Q}, \pi^{\text{pl}}_{\text{Nash}}, \pi^{\text{op}}_{\text{Nash}}\right)$$

computes Nash based on Q-estimate — updates Q-estimate based on computed Nash

Entropy-Regularized Policy Approximation

Fixed entropy regularization [1]

$$\mathcal{V}_{\text{KL}}^{\pi^{\text{pl}},\pi^{\text{op}}}(s) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(\mathcal{R}(S_t, A_t^{\text{pl}}, A_t^{\text{op}}) - \frac{1}{\beta^{\text{pl}}}\log\frac{\pi^{\text{pl}}(a_t^{\text{pl}}|s_t)}{\rho^{\text{pl}}(a_t^{\text{pl}}|s_t)} - \frac{1}{\beta^{\text{op}}}\log\frac{\pi^{\text{op}}(a_t^{\text{op}}|s_t)}{\rho^{\text{op}}(a_t^{\text{op}}|s_t)}\right)\right]$$

regulated policy; fixed inverse temperature; fixed reference policy; information cost

Closed-form max-min soft solution under regularization

Marginalization:

$$\mathcal{Q}_{\text{KL}}^{\text{pl}*}(s, a^{\text{pl}}) = \frac{1}{\beta^{\text{op}}}\log\sum_{a^{\text{op}}} \rho^{\text{op}}(a^{\text{op}}|s)\exp(\beta^{\text{op}}\mathcal{Q}_{\text{KL}}(s, a^{\text{pl}}, a^{\text{op}}))$$

Soft-Nash Policies:

$$\pi_{\text{KL}}^{\text{pl}*}(a^{\text{pl}}|s) = \frac{1}{Z^{\text{pl}}(s)}\rho^{\text{op}}(a^{\text{op}}|s)\exp\left(\beta^{\text{op}}\mathcal{Q}_{\text{KL}}^{\text{pl}*}(s, a^{\text{pl}}, a^{\text{op}})\right)$$

Two soft operators: inverse temperature dependent

$$\left(\pi_{\text{KL}}^{\text{pl}}, \pi_{\text{KL}}^{\text{op}}\right) = \Gamma_{\text{KL}}^{\beta}(\mathcal{Q}_{\text{KL}}, \rho); \qquad \mathcal{Q}_{\text{KL}} = \Gamma_2^{\beta}(\mathcal{Q}_{\text{KL}}, \rho)$$

NOT original NE — computes soft Nash — Updates $\mathcal{Q}_{\text{KL}}$ with reference ρ

Soft Nash Q2 Algorithm

SNQ2 learns two Q-values simultaneously:

(1) Original Q-value $\mathcal{Q}$

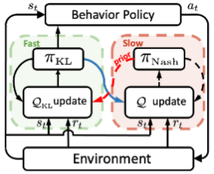
(2) Entropy-regularized Q-value $\mathcal{Q}_{\text{KL}}$

Slow Module:

- Learns standard Q-value and Nash policies
- Slow but produces Nash

Fast Module:

- Learns entropy-regularized Q-value and soft-optimal policies
- Fast but only an approximation of the Nash policies



A schematic of the SNQ2 algorithm

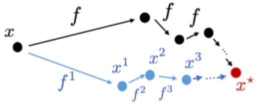
- Coupling of the two Modules:
  Use Nash policies from the slow module to update the priors used in the fast module
  Use soft-policies from the fast module to update Q-values in the slow module
- Actively **adapts entropy regularization**
  - Reduce inverse temperature β over time
  - Update reference policies using Nash policies from original Q-estimate
- A dynamic schedule scheme is introduced to balance the two modules
  - Observes the Q-difference between two updates
  - Decides when to perform Nash prior updates and reduce inverse temperature

Algorithm 1: SNQ2-Learning Algorithm

```
Inputs: Priors ρ, Learning rates α and η; initial prior update
  episode M = ΔM_0; Nash update frequency T;
Set Q(s, a^pl, a^op) = Q_KL(s, a^pl, a^op) = 0;
Set β^pl and β^op to some large values;
while Q not converged do
    while episode i not end do
        Compute π_KL(s_t) ← [Γ^β_KL(Q_KL, ρ)](s_t);
        Collect transition (s_t, a^pl_t, a^op_t, r_t, s_{t+1}) where
          a^pl_t ~ π^pl_KL(s_t), a^op_t ~ π^op_KL(s_t);
        if t mod T == 0 then
            Compute V(s_{t+1}) =
              max_{π^pl} min_{a^op} Σ_{a^pl} Q(s_{t+1}, a^pl, a^op) π^pl(a^pl|s_{t+1});
        else
            Compute
              V(s_{t+1}) = π^pl_KL(s_{t+1})^T Q(s_{t+1}) π^op_KL(s_{t+1});
        end
        Update Q(s_t, a^pl_t, a^op_t) with V(s_{t+1}) via (13);
        Update Q_KL(s_t, a^pl_t, a^op_t) via (12);
    end
    if i == M then
        Compute π_Nash ← Γ_Nash Q_t;
        Update priors ρ_new ← π_Nash;
        Update schedule as in Algorithm 3:
          ΔM, β_new = DS(ρ_new, ρ, β, ΔM, Q);
        Update next prior update schedule M += ΔM;
        Update priors ρ ← ρ_new,  β ← β_new;
        Decrease learning rates α and η;
    end
end
return Q(s, a^pl, a^op).
```

Challenges in convergence analysis:

- With decreasing β, the operators used to update Q-value changes
- Standard fixed-point argument cannot be directly applied



Convergence Analysis

Theorem 1 Let $(\mathcal{X}, \rho)$ be a complete metric space. Let $f^n: \mathcal{X} \to \mathcal{X}$ be a family of contraction operators such that for all $n = 1, 2, \ldots$ there exists $d^n \in (0,1)$, such that $\rho(f^n x, f^n y) \le d^n \rho(x, y)$ for all $x, y \in \mathcal{X}$. Assume that $\lim_{n\to\infty} d^n = d \in (0,1)$. Let $x \in \mathcal{X}$ be a starting point and let $x^n = f^n \cdots f^1 x$ be the result of sequentially applying the operators $f^1, \ldots, f^n$ to x. If the sequence of operators $\{f^n\}_{n=1}^{\infty}$ convergence pointwise to f, then f is also a contraction mapping with contraction factor d. Furthermore, if x^* is the fixed point of f, then for every $x \in \mathcal{X}$, $\lim_{n\to\infty} x^n = x^*$.

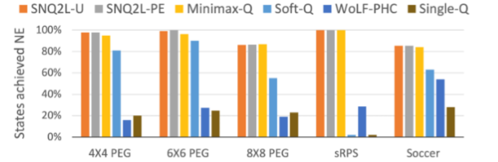
The convergence of SNQ2 can be shown through the following argument:

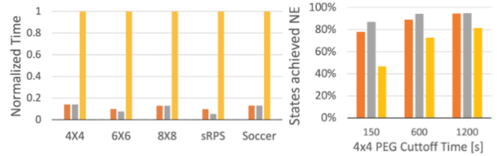
- As β approaches zero, the update rule of SNQ2 converges to Shapley's method
- Per Theorem 1, SNQ2 converges to the fixed point of Shapley's method, which is the Nash Q-value of the game.

FULL PAPER

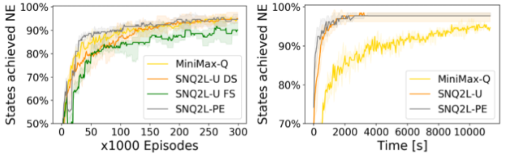
Numerical Experiments

Experiments are conducted in Pursuit-Evasion games (PEG), Sequential Rock-Paper-Scissor (sRPS) and Soccer games





- ➢ Without updating regularization, two-agent soft-Q [1] failed to converge to a Nash in sRPS.
- ➢ With updating regularization, SNQ2 achieves same level of convergence as Minimax-Q
- ➢ Significant reduction in learning time
- ➢ Warm starting (-PE) gives better convergence give the same cutoff time



- ➢ Similar episode-wise convergence trend as Minimax-Q
- ➢ Time-wise trend shows a significant speed up
- ➢ Warm starting (-PE) gives a better convergence trend comparing to uniform prior (-U)
- ➢ Dynamic scheduling (DS) improves episode-wise convergence speed

References

[1] Grau-Moya, J., Leibfried, F., & Bou-Ammar, H. (2018). Balancing two-player stochastic games with soft q-learning. *arXiv preprint arXiv:1802.03216.*

[2] Zhang, Q., Guan, Y., & Tsiotras, P. (2020). Learning Nash Equilibria in Zero-Sum Stochastic Games via Entropy-Regularized Policy Approximation. arXiv preprint arXiv:2009.00162.