Deep Reinforcement Learning from Self-Play in Imperfect-Information Games

Johannes Heinrich et al

연세대학교 컴퓨터과학과
강효림

Index

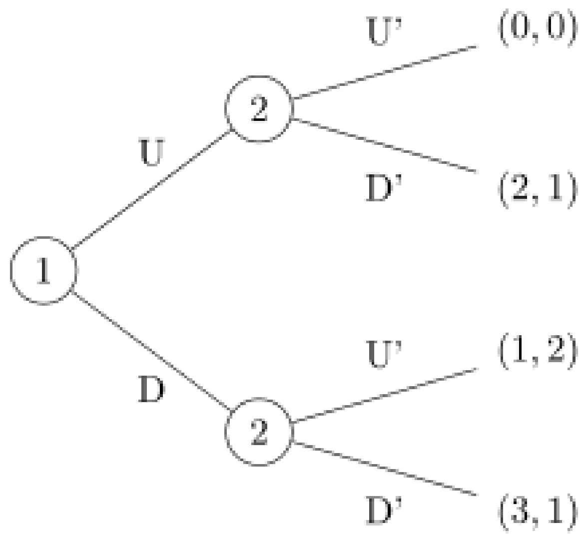
- 1. Background
  - Extensive-form game
  - Fictitious play
- 2. NFSP
- 3. Experiment

BACKGROUND

Extensive-Form Game

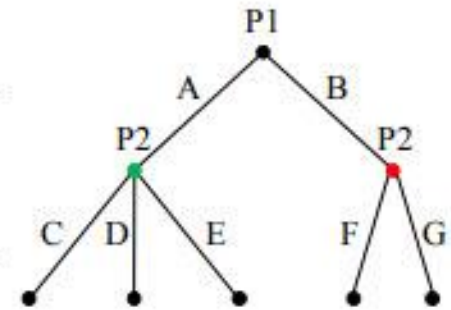
- 전개형 게임 – 수형도의 모양

- ex)



Extensive Form Game's Normal Form Rep'

- Example:



Exponential!

P2 \ P1	DH	DI	DJ	EH	EI	EJ
AF	a, b	a, b	a, b	e, f	e, f	e, f
AG	c, d	c, d	c, d	g, h	g, h	g, h
BF	i, j	k, ℓ	m, n	i, j	k, ℓ	m, n
BG	i, j	k, ℓ	m, n	i, j	k, ℓ	m, n
CF	p, q	p, q	p, q	p, q	p, q	p, q
CG	p, q	p, q	p, q	p, q	p, q	p, q

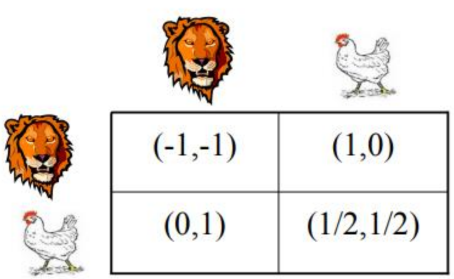
Alternative: Use Behavioural Strategies

- Pure Strategy
  - 결정론적인(deterministic) 정책
  - 모든 상황에 대한 deterministic plan
- Mixed Strategy
  - 가능한 Pure Strategy에 대한 확률분포
- $\pi^i(u)$ 및 이들의 확률분포로 이루어지는 Mixed Strategy로 Normal Form Rep. 를 대체

Fictitious Play

- Definition:
  1) 상대방의 Average Behaviour에 대하여
  2) Best Response를 행하는 Play
- Example:

Chicken



A	B
Chicken	Chicken
Lion	Lion
Lion	Chicken
Lion	Chicken
Lion	Chicken

이 때는
Random!

Generalized Weakened Fictitious Play

- Using Mixed Strategy Rep.

Definition 5. A generalised weakened **fictitious play** is a process of mixed strategies, $\{\Pi_t\}$, $\Pi_t \in \times_{i \in \mathcal{N}} \Delta^i$, s.t.

Best response toward others' average policy

$$\Pi^i_{t+1} \in (1-\alpha_{t+1})\Pi^i_t + \alpha_{t+1}(b^i_{\epsilon_t}(\Pi^{-i}_t) + M^i_{t+1}), \ \forall i \in \mathcal{N},$$

with $\alpha_t \to 0$ and $\epsilon_t \to 0$ as $t \to \infty$, $\sum_{t=1}^{\infty} \alpha_t = \infty$, and $\{M_t\}$ a sequence of $\boxed{\text{perturbations}}$ that satisfies $\forall T > 0$

$$\lim_{t \to \infty} \sup_k \left\{ \left\| \sum_{i=t}^{k-1} \alpha_{i+1} M_{i+1} \right\| \text{ s.t. } \sum_{i=t}^{k-1} \alpha_{i+1} \leq T \right\} = 0.$$

Randomly perturbed payoffs

- η_1, η_2 are iid with a smooth distribution f_x.
- As x approaches to 0, f_x becomes a unit mass at 0.

	H	T
H	$2+\eta_1, 2+\eta_2$	$\eta_1, 0$
T	$0, \eta_2$	1,1

Extensive Form Fictitious Play (XFP)

Lemma 6. *Let π and β be two $\boxed{\text{behavioural strategies,}}$ Π and B two $\boxed{\text{mixed strategies}}$ that are realization equivalent to π and β, and $\lambda_1, \lambda_2 \in \mathbb{R}_{\geq 0}$ with $\lambda_1 + \lambda_2 = 1$. Then for each information state $u \in \mathcal{U}$,*

$$\mu(u) = \pi(u) + \frac{\lambda_2 x_\beta(\sigma_u)}{\lambda_1 x_\pi(\sigma_u) + \lambda_2 x_\beta(\sigma_u)}(\beta(u) - \pi(u))$$

defines a behavioural strategy μ at u and μ is realization equivalent to the mixed strategy $M = \lambda_1 \Pi + \lambda_2 B$.

Extensive Form Fictitious Play (XFP)

- Generalized Weakened Fictitious Play

$$\Pi^i_{t+1} \in (1-\alpha_{t+1})\Pi^i_t + \alpha_{t+1}(b^i_{\epsilon_t}(\Pi^{-i}_t) + M^i_{t+1}),\ \forall i \in \mathcal{N},$$

- By the previous lemma:

Realization Plan

$$x_\pi(\sigma_u) = \prod_{(u',a)\in\sigma_u} \pi(u',a),$$

$$\beta^i_{t+1} \in b^i_{\epsilon_{t+1}}(\pi^{-i}_t),$$

$$\pi^i_{t+1}(u) = \pi^i_t(u) + \frac{\alpha_{t+1} x_{\beta^i_{t+1}}(\sigma_u)\left(\beta^i_{t+1}(u) - \pi^i_t(u)\right)}{(1-\alpha_{t+1})x_{\pi^i_t}(\sigma_u) + \alpha_{t+1}x_{\beta^i_{t+1}}(\sigma_u)}$$

$$\Rightarrow \quad \sigma(s,a) \propto \lambda_1 x_{\pi_1}(s)\pi_1(s,a) + \lambda_2 x_{\pi_2}(s)\pi_2(s,a) \quad \forall s, a,$$

XFP's Pseudocode

Algorithm 1 Full-width extensive-form fictitious play

function FICTITIOUSPLAY(Γ)
  Initialize π_1 arbitrarily
  $j \leftarrow 1$
  while within computational budget **do**
    $\beta_{j+1} \leftarrow$ COMPUTEBRS(π_j)
    $\pi_{j+1} \leftarrow$ UPDATEAVGSTRATEGIES(π_j, β_{j+1})
    $j \leftarrow j + 1$
  end while
  return π_j
end function

function COMPUTEBRS(π)
  Recursively parse the game's state tree to compute a best response strategy profile, $\beta \in b(\pi)$.
  return β
end function

function UPDATEAVGSTRATEGIES(π_j, β_{j+1})
  Compute an updated strategy profile π_{j+1} according to Theorem 7.
  return π_{j+1}
end function

Previous Slide's Formula

NFSP

XFP ➡ (N)FSP

Algorithm 1 Full-width extensive-form fictitious play

function FictitiousPlay(Γ)
  Initialize π_1 arbitrarily
  $j \leftarrow 1$
  while within computational budget **do**
    $\beta_{j+1} \leftarrow$ ComputeBRs(π_j)
    $\pi_{j+1} \leftarrow$ UpdateAvgStrategies(π_j, β_{j+1})
    $j \leftarrow j + 1$
  end while
  return π_j
end function

function ComputeBRs(π)
  Recursively parse the game's state tree to compute a best response strategy profile, $\beta \in b(\pi)$.
  return β
end function

function UpdateAvgStrategies(π_j, β_{j+1})
  Compute an updated strategy profile π_{j+1} according to Theorem 7.
  return π_{j+1}
end function

Reinforcement Learning

Supervised Learning

NFSP Pseudocode

Algorithm 1 Neural Fictitious Self-Play (NFSP) with fitted Q-learning

Initialize game Γ and execute an agent via RUNAGENT for each player in the game
function RUNAGENT(Γ)
  Initialize replay memories $\mathcal{M}_{RL}$ (circular buffer) and $\mathcal{M}_{SL}$ (reservoir)
  Initialize average-policy network $\Pi(s, a \mid \theta^{\Pi})$ with random parameters θ^{Π}
  Initialize action-value network $Q(s, a \mid \theta^{Q})$ with random parameters θ^{Q}
  Initialize target network parameters $\theta^{Q'} \leftarrow \theta^{Q}$
  Initialize anticipatory parameter η
  for each episode **do**

Dilemma

    Set policy $\sigma \leftarrow \begin{cases} \epsilon\text{-greedy}\,(Q)\,, & \text{with probability } \eta \\ \Pi, & \text{with probability } 1-\eta \end{cases}$
    Observe initial information state s_1 and reward r_1
    for $t = 1, T$ **do**
      Sample action a_t from policy σ
      Execute action a_t in game and observe reward r_{t+1} and next information state s_{t+1}

Storing Sample

      Store transition $(s_t, a_t, r_{t+1}, s_{t+1})$ in reinforcement learning memory $\mathcal{M}_{RL}$
      if agent follows best response policy $\sigma = \epsilon$-greedy (Q) **then**
        Store behaviour tuple (s_t, a_t) in supervised learning memory $\mathcal{M}_{SL}$
      end if

Loss

      Update θ^{Π} with stochastic gradient descent on loss
        $\mathcal{L}(\theta^{\Pi}) = \mathbb{E}_{(s,a)\sim\mathcal{M}_{SL}}\left[-\log \Pi(s, a \mid \theta^{\Pi})\right]$
      Update θ^{Q} with stochastic gradient descent on loss
        $\mathcal{L}\left(\theta^{Q}\right) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{M}_{RL}}\left[\left(r + \max_{a'} Q(s', a' \mid \theta^{Q'}) - Q(s, a \mid \theta^{Q})\right)^2\right]$
      Periodically update target network parameters $\theta^{Q'} \leftarrow \theta^{Q}$
    end for
  end for
end function

Storing Sample

- For Supervised Learning
  - Learn average behavior of the agent itself
  - Stores (s, a) when the agent follows best-response policy (best-response's reservoir)

- For Reinforcement Learning(Q-learning)
  - Learn best-response toward others' average policy
  - Always stores (s, a, r, s`) (off-policy learning)

Self-Play시, Dilemma

- 모든 Agent가 Average Policy만 따른다면
  - 다른 Agent가 Average Policy 따름이 보장 (Fictitious Play 전제)
  - Off-Policy로 Q함수 update
  - 그러나 Supervised Learning을 위한 Sample이 X

- 따라서 일정확률로 best-response policy를 행함

$$\text{Set policy } \sigma \leftarrow \begin{cases} \epsilon\text{-greedy}\,(Q), & \text{with probability } \eta \\ \Pi, & \text{with probability } 1-\eta \end{cases}$$

Loss

- Loss

Update θ^{Π} with stochastic gradient descent on loss

$$\mathcal{L}(\theta^{\Pi}) = \mathbb{E}_{(s,a)\sim\mathcal{M}_{SL}}\left[-\log \Pi(s,a \mid \theta^{\Pi})\right]$$

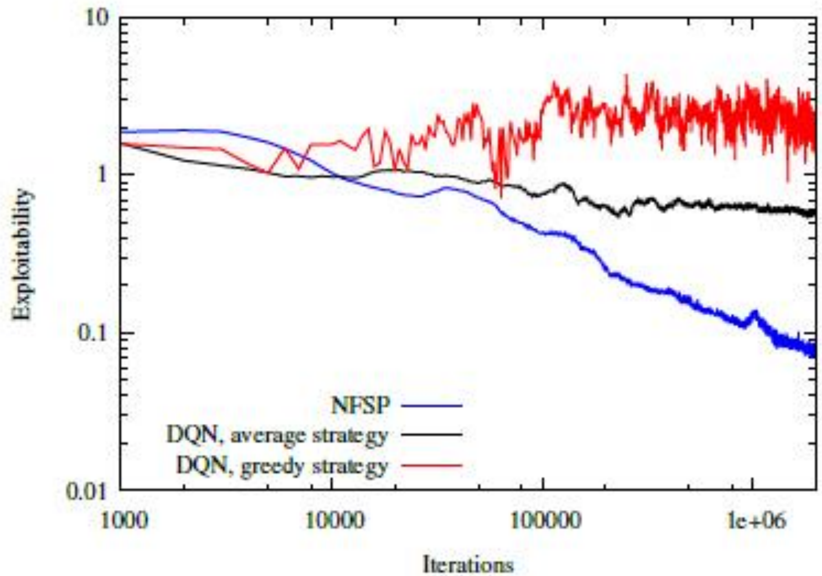
Update θ^{Q} with stochastic gradient descent on loss

$$\mathcal{L}\left(\theta^{Q}\right) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{M}_{RL}}\left[\left(r + \max_{a'} Q(s',a' \mid \theta^{Q'}) - Q(s,a \mid \theta^{Q})\right)^2\right]$$

- $-\log \Pi(s,a|\theta^{\Pi})$:
  s에서 **실제 행한 a**를 행할 확률을 높이는 방향으로 update
- Reinforcement Learning:
  전형적인 Q-Learning의 Loss

EXPERIMENT

Leduc Hold'em NFSP vs DQN



(c) Comparison to DQN

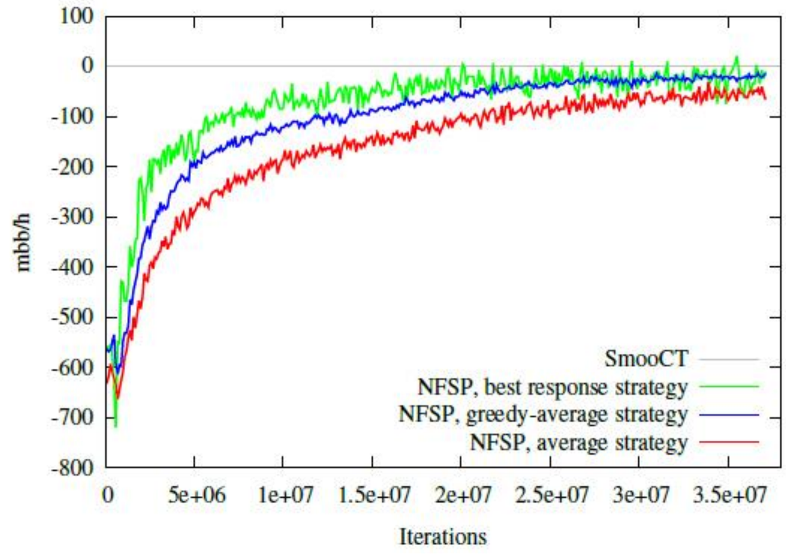
DQN result

- DQN => $\eta = 1$인 NFSP
- DQN은 Average policy도 Nash 균형 수렴하지 않음
- Why?
  - $\varepsilon - greedy$정책만으로 sample을 수집

highly correlated, focused on narrow state distribution

- NFSP는 보다 slowly changing, stable data distribution

Limit Texas Hold'em



Match-up	Win rate (mbb/h)
escabeche	-52.1 ± 8.5
SmooCT	-17.4 ± 9.0
Hyperborean	-13.6 ± 9.2

Person who always folds: 750
Expert: 40~60

끝

들어주셔서 감사합니다.