2019-Celli

Coordination in Adversarial Sequential Team Games via Multi-agent Deep Reinforcement Learning

---

- **Date**: 2019-12-16
- Link
- **Authors:**
  - Celli, Andrea
  - Ciccone, Marco
  - Bongo, Raffaele
  - Gatti, Nicola
- **Cites:**
  - 2018-Celli
  - 2019-Farina
- **Cited by:**
- **Keywords:** #extensive-form-game #multi-player-games #team-games #ex-ante-coordination #soft-team-actor-critic #actor-critic #MARL #team-maxim-equilibrium-w-coordination-device
- **Collections:**
- **Status:** #completed

---

0 - Abstract

- **Game setup**
  - Zero-sum games with a team of players facing an opponent.
  - Coordination can only occur before the start of the game.
    - Necessitating an *ex-ante* team strategy.
  - *Ex-ante coordination* - team members discuss and agree on strategy before the game starts, then are unable to coordinate during the game, except through publicly observed actions.
  - Ex: Bridge, collusion in poker and bidding.

- **Approach**
  - Use Soft Team Actor-Critic (STAC) to solve the team's coordination problem, without domain knowledge.
  - Team members communicate before the game using exogenous signals.

- **Results**
  - Reaches near-optimal coordinated strategies in perfectly and partially observable games.
  - Outperforms existing RL approaches.

1 - Introduction

- Finding an equilibrium with ex-ante coordination is NP-hard and inapproximable.

- Prior work:
  - First optimal coordination strategy algorithm:
    - 2018-Celli
    - Strategy representations:
      - Team members play joint normal-form actions.
      - Adversary plays a sequence-form strategy.
    - Column generation algorithm is used to compute the optimal team strategy.

  - *Fictitious Team-Play*:
    - 2019-Farina
    - Requires the solving of mixed-integer linear programs (MILP).
    - Limited scalability, can only solve games with up to 800 infosets per player.

- Problems with prior work:
  - **Biggest issue** - Need explicit representations of the sequential game.
    - Might not be exactly known to players.
    - Might be too big to be stored in a computer's memory.

  - Unable to learn in a sample-based fashion.
    - CFR and FP require models.
    - Often times, these models require domain specific knowledge to achieve good results.

- This paper investigates MARL as a solution to these problems.

- Advantages of MARL:
  - Don't require perfect environmental knowledge.

- Learn in sample-based fashion via interaction with the environment.

- Disadvantages of MARL:
  - Players are non-homogeneous.
    - Hidden information
    - Arbitrary action spaces

  - Player's policies may be conditioned only on local information.
    - Important because coordinated strategies rely on the player's ability to interpret exogeneous signals.
    - The partially observable games prevent the players from conditioning their policies on the complete game history, which would solve the problem of conditioning only on local information.

- Contributions:
  - Use STAC to learn coordinated strategies directly from experience.
  - Design an ex-ante communication framework for the team members.
  - Show strong performance on game benchmarks.

2 - Preliminaries

2.1 - Reinforcement Learning

- Basics:
  - Agent takes an action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$ according to a policy π that maps a probability distribution over the set of available actions, and receives a reward r_t from the environment as well as the next state.
  - The agent's goal is to maximize expected discounted return: $R_t := \sum_{i=0}^{\infty} \gamma^i r_{t+i}$.
  - Therefore the optimal policy is $\pi^* \in \text{argmax} E_\pi[R_0]$.

- Value-based methods:
  - Compute π^* by acting greedily wrt value estimations, either:
    - State-value function: $v_\pi(s) := E_\pi\left[R_t \mid S_t = s\right]$
    - Action-value function: $Q_\pi(s,a) := E_\pi\left[R_t \mid S_t = s,\ A_t = a\right]$

- Policy gradient methods:
  - Allow the policy π_θ to be differentiable and parameterized by θ.
  - Adjust θ via gradient ascent to improve π_θ wrt to a score function $J(\pi_\theta)$.
  - The gradient score function given by the policy gradient theorem:

$$\nabla_\theta J(\theta) = E_{(s,a)\sim\rho_\pi}\left[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)\right]$$

    where ρ_π is the state-action marginals of the trajectory distribution induced by $\pi(a_t|s_t)$.
  - Can use a cyclical process to incrementally improve the policy and value estimates:
    - Policy evaluation to learn $Q^\pi(s,a)$, called the *critic*.
    - Policy improvement to learn π_θ, called the *actor*.

2.2 - Extensive-Form Games

- Basics:
  - Extensive-form game (EFG) - models sequential interactions between a set of players, $\mathcal{P}$.
  - Exogenous stochasticity is represented by a *chance* player (a.k.a *nature*).
  - History $h \in H$ - set of all actions taken by players (including nature) up to the present.
  - Imperfect-information game - each player can only see their own information states.
  - Perfect recall - players remember all past states and actions that were observable to them.

- Information states:
  - *Information state* s_t - set of histories for a player which are consistent with the player's previous observations (i.e. the set of states which the player can not differentiate between).

  - Reasons for information states:
    1. Private information determined by the environment.
       - Ex - hands in a poker game.
    2. Limitations in the observability of other players' actions.

  - *Perfectly observable* - when the information states in the game are only created by private information from the environment, reason 1 above.

- Strategy profiles:
  - *Behavioral strategy* - policy that maps information states to probability distributions.

  - *Exploitability* $e(\pi)$ - the average incentive of a player to deviate from their strategy:

$$e(\pi) := \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} e_i(\pi)$$

    where:
    - π - is the strategy profile which defines the strategy for each player $\pi = (\pi_i)_{i\in\mathcal{P}}$
    - $e_i(\pi)$ is the incentive for player i to deviate from its strategy in π .
      Defined as

$$e_i(\pi) := \max_{\pi'_i} E_{\pi'_i,\, \pi_{-i}}\left[R_{0,i}\right] - E_{\pi}\left[R_{0,i}\right]$$

- *Nash equilibrium* - a strategy profile in which no player has an incentive to deviate form her strategy, $e(\pi) = 0$.

- Normal-form strategies:
  - *Plan* σ_i - deterministic policy for player i that selects a single action at each information state.
    - Equivalent to the normal-form representation of the EFG.
  - Σ_i - set of all plans for player i.
    - Grows exponentially as the number of infosets increases.
  - *Normal-form strategy* x_i - probability distribution over Σ_i.
    - $\mathcal{X}_i$ - set of normal form strategies for player i.

3 - Team's Coordination: A Game-Theoretic Perspective

- Team setup:
  - *Team* - set of players who share the same objectives.
  - Focus on a two-player teams $(T1,\, T2)$ playing against a single adversary (A).

- Rules:
  - Team members can only communicate before the game.
  - During the game they can only observe the actions their teammate makes.

- Intuitive understanding of team coordination:
  - Team members are at an advantage.
  - Before the game, they can coordinate each other's actions for any given state.
  - Then, by observing their teammates actions during the game, they can make an inference on their teammate's hidden information and act accordingly.

- Game theoretic understanding of team coordination:
  - *Coordination device* - used to select a pair of strategies for the two players from the set of joint plans, according to a probability distribution. This allows for correlation between the two player's strategies.
  - Notation:
    - $\Sigma_T = \Sigma_{T1} \times \Sigma_{T2}$ - set of joint plans for the team.
    - $R_{t,T}$ - return of the team from time t where $R_{t,A} = -R_{t,T}$, for all t.
  - *Definition 1 Team-maxim equilibrium with coordination device* (TMECor) - a pair $\zeta = (\pi_A, x_T)$ with $x_T \in \Delta(\Sigma_T)$ is a TMECor iff:

    $$e_A(\zeta) := \max_{\pi'_A} E_{\pi'_A,(\sigma_1,\sigma_2)\sim x_T}\left[R_{0,A}\right] - E_{\pi_A,(\sigma_1,\sigma_2)\sim x_T}\left[R_{0,A}\right] = 0$$

    and

    $$e_A(\zeta) := \max_{(\sigma'_1,\sigma'_2)\in\Sigma_T} E_{\pi_A,(\sigma'_1,\sigma'_2)}\left[R_{0,T}\right] - E_{\pi_A,(\sigma_1,\sigma_2)}\left[R_{0,T}\right] = 0$$

    - i.e. the team nor the adversary have an incentive to change their strategy.
  - *epsilon-TMECor* - approximation of TMECor where neither party can gain more than ϵ by deviating from their strategy.

- RL and Team coordination:
  - Traditional RL algorithms output behavioral strategies for single players.
  - Therefore, they're unable to coordinate their strategies amongst each other.
  - One could try to adapt the RL algorithms to work over the set of coordinated strategies Σ_T.
  - However, Σ_T is too large in practice for this to work.
  - Therefore, we must develop an RL algorithm that is capable of outputting coordinated strategies without explicitly working over Σ_T.

4 - Soft Team Actor-Critic (STAC)

- *Soft Team Actor-Critic* (STAC) - scalable sample-based technique to approximate the team's ex-ante coordinated strategies.
  - Achieved by mimicking the behavior of the coordination device through the use of an exogenous signaling scheme.
  - Teammates correlate their strategies by assigning meaning to symbols that are shared with one another.

4.1 - Soft Actor Critic

- *Soft Actor Critic* (SAC) - off-policy deep RL algorithm based on the maximum entropy (maxEnt) principle.
  - Uses an actor-critic architecture with separate policy and value function networks.
  - Actor's goal is to learn the policy that maximizes the expected reward while also maximizing its entropy at every visited state.

    Defined by the maxEnt score function:

    $$J(\pi) := \sum_t E_{(s_t,a_t)\sim\rho_\pi}\left[r_t + \alpha\mathcal{H}(\pi(\cdot|s_t))\right]$$

    where:
    - Temperature parameter, $\alpha > 0$ - weights the importance of the entropy term.

    - Entropy, $\mathcal{H}(\pi(\cdot|s_t)) = -\sum_{a \in A} \pi(a|s_t) \log \pi(a|s_t)$

      Measure of the stochasticity of the agent's policy at s_t.

      A deterministic policy has zero entropy; a uniform policy has maximum entropy.

  - The entropy term gives the agent a bias toward exploration.
  - Reuses previous samples that it collects in a replay buffer, thereby increasing sample efficiency.
  - Introduced by [2018-Haarnoja](#).

4.2 - Multi-Agent Soft Actor-Critic

- *Centralized training with decentralized execution* - extra, hidden information is shared among players at training time to aid in the emergence of cooperative behaviors, but taken away at testing time.

- Training/Testing framework for STAC:
  - **Actors** - team members are non-homogenous, play at different decision points, and, in turn, collect different sets of observations. Therefore, policy networks are needed for each player.
    - This allows for decentralized policies.

  - **Critic** - one critic for the team that has access to the complete team state (i.e. the private information of both team members).
    - This is possible because we allow team members to share observations at training time and rewards are homogenous for the team (i.e. the two players work to generate a single team reward).
    - This sharing of parameters allows for the players to learn how to coordinate with each other, during training.

4.3 - Signal Conditioning

- Introduction:
  - Ex-ante coordinated strategies are defined over the joint plan space Σ_T, this is too large.
  - Alternatively, we use an *approximate* coordination device, modeled as a fictitious player, called the *signaler*.
  - During training time, the players achieve a shared consensus on the meaning of the signals.
  - They then use this coordination to select their policies before the games starts.

- *Definition 2 Signaler* - Given a set of signals Ξ and a probability distribution $x_s \in \Delta(\Xi)$, a signaler is a non-strategic player which draws $\xi \sim x_s$ at the beginning of each episode, and subsequently communicates ξ to team members.

  Assume the number of signals is fixed and x_s is uniformly distributed.

- Shared consensus algorithm:
  - **Policy evaluation step**
    - *Value-conditioner network* - performs action-value and state-value estimates.
      - Its parameters are produced via a *hypernetwork* conditioned on the observed signal ξ.
        - This conditions the player's perception of their states/actions on the given signal.
      - Note - learning the hypernetwork's parameters degrades performance.

  - **Policy improvement step**
    - *Policy conditioner network* - given the local state, outputs a probability distribution over the set of available actions, for a team member.
      - Its parameters are produced by a fixed number of hypernetworks conditioned on the observed signal.
        - This conditions agent behavior on the given signal.
      - Hypernetworks are shared by all team members.
        - Critical to developing a shared signal meaning.
      - Updated by minimizing Kullback-Leibler divergence as in the original SAC.

  - **Algorithm**

    Algorithm 1 Soft Team Actor-Critic

    Require: $\theta_1, \theta_2, \phi, \psi$ ▷ Initial parameters
    $\bar{\psi} \leftarrow \psi$ ▷ Initialize target network weights
    $\mathcal{D} \leftarrow \varnothing$ ▷ Initialize an empty replay pool
    for each iteration **do**
     $\xi \sim x_s$ ▷ The signaler draws ξ
     for each environment step **do**
      $a_t \sim \pi_\phi(a_t|s_t;\xi)$ ▷ Sample action from the policy, conditioned on the signal
      $s_{t+1} \sim \mathcal{T}(s_t|s_t, a_t)$ ▷ Sample transition from the environment
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r_t, s_{t+1}, \xi)\}$ ▷ Store the transition in the replay pool
     end for
     for each gradient step **do**
      $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$ ▷ Update the V-function parameters
      $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$ ▷ Update the Q-function parameters
      $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$ ▷ Update policy weights
      $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$ ▷ Periodically update target network weights
     end for
    end for

5 - Experimental Evaluation

5.1 - Experimental Setting

- The team's expected payoff against a best-responding adversary (i.e. worst-case payoff) will be used as the performance metric.

- Baselines:
  - Neural Fictitious Self-Player (NFSP) - sample-based variation of fictitious play.
    - See 2016-Heinrich.
    - Two variants are used as baselines: NFSP-independent and NFSP-coordinated-payoffs.
  - SAC.
    - This allows us to measure the effect of the team-coordination.

- Game instances:
  1. Guessing game - the team members must guess the action the adversary will take. The team is only rewarded if both players guess correctly.
     - See Example 1 for more discussion.
  2. Three-player Leduc poker.
     - 3 ranks and two suits.

- Architecture details.
  - See the paper for details.

5.2 - Main Experimental Results

- Guessing game (imperfect observability):
  - Benchmark performance:
    - NFSP-independent - unable to reach the optimal worst-case payoff.
    - NFSP-coordinated - achieves optimal worst-case payoff for non-coordinating agents.
    - SAC exemplifies cyclic behavior - team members guess actions deterministically, making them easily exploitable by the adversary.

  - STAC (with $|\Xi| = 3$):
    - First two signals, the players learn to play toward the $K/2$ payoff.
    - The third signal, the players play toward the K payoff.
    - This is equivalent to the optimal TMECor EFG strategy.

  - Take away - with a sufficient number of signals, STAC is capable of achieving TMECor performance.

- Leduc Poker (perfect observability):
  - Benchmark strategies were able to achieve good results because the player's ability to observe the complete history of its teammate is enough for implicit coordination.
  - STAC was still able to achieve a modest performance improvement over the benchmarks.

6 - Related Works

- See paper for more details

7 - Discussion

- Introduced STAC as a method for approximating TMECor strategies in such a way that does not require perfect knowledge of the EFG, nor represent it explicitly.

- The key to STAC's coordination is the exogenous signal framework where team members can systematically assign meaning to shared signals, thereby correlating their individual strategies to maximize the team's reward.

- Experiments show that STAC agents are able to reach near optimal team strategies.