Reinforcement Learning in Multi-Agent Systems

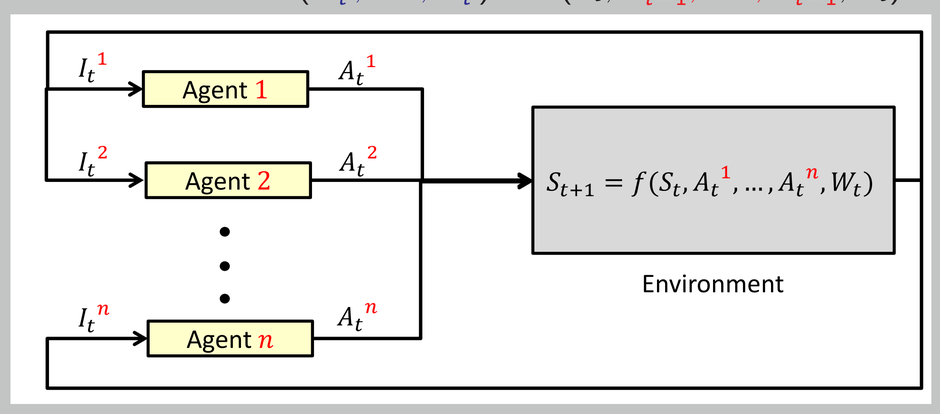
Jalal Arabneydi and Aditya Mahajan
ECE Department, McGill University

Motivation

- Multi-agent Reinforcement Learning (RL) arises in many applications ranging from **networked control systems**, **robotics**, **transportation networks**, **sensor networks**, **economics**, and **smart grids**.
- Agents have different information that creates discrepancy in perspectives that makes it conceptually challenging to establish cooperation among agents.
- Finding team-optimal solution is more challenging when agents have only **partial knowledge** or **no knowledge** of system model.

Model

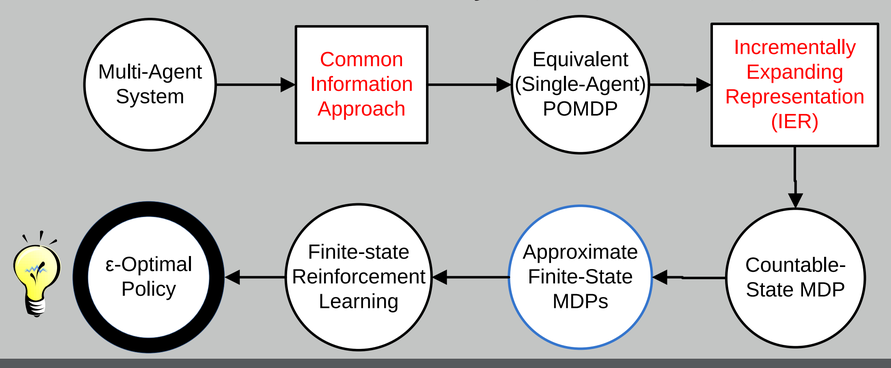
- Consider $n \in \mathbb{N}$ agents, where $t \in \mathbb{N}$ denotes time.
- State of system $S_t \in \mathcal{S}$ and action of agent i: $A_t^i \in \mathcal{A}^i$.
- Information of agent i: $I_t^i \subseteq \{O_{1:t}^1, \dots, O_{1:t}^n, A_{1:t-1}^1, \dots, A_{1:t-1}^n\}$, where observations are as follows: $(O_t^1, \dots, O_t^n) = h(S_t, A_{t-1}^1, \dots, A_{t-1}^n, V_t)$.



- **Assumption**: There exist a sequence of actions (or an action) that reset(s) the environment.
- Reward given control strategy $\mathbf{g}$: $J(\mathbf{g}) = \mathbb{E}^{\mathbf{g}}\left[\sum_{t=1}^{\infty} \gamma^t r(S_t, A_t^1, \dots, A_t^n)\right]$.
- Objective: Develop a (model-based or model-free) reinforcement learning algorithms that guarantees an ϵ-optimal strategy $\mathbf{g}^*$ i.e. $J^* - J(\mathbf{g}^*) \leq \epsilon$.

Methodology

- **Partial History Sharing**: Split $I_t^i = \{M_t^i, C_t\}$, where $C_t = \cap_{d>t} \cap_{i=1}^n I_d^i$ is common information and M_t^i is local information. Define $Z_t = C_{t+1} \backslash C_t$ as common observation, then $C_{t+1} = Z_{1:t}$.
  - The update of local information $M_{t+1}^i \subseteq \{M_t^i, A_t^i, O_{t+1}^i\} \backslash Z_t$.
  - The size of Z_t and the size of $M_t^i, \forall i$, are uniformely bounded in time t.



Main Steps

- **Common Information Approach (Step 1)**: For agent i, define prescription function β_t^i that maps local information M_t^i to action A_t^i i.e. $A_t^i = \beta_t^i(M_t^i)$.
  - Virtual coordinator observes C_t and prescribes $\beta_t := (\beta_t^1, \dots, \beta_t^n) \in \mathcal{G}$.
  - In equivalent coordinated system, $\Pi_t = \mathbb{P}(S_t, M_t^{1:n} | C_t, \beta_{1:t-1})$ is an information state and reward function is $\hat{r}(\Pi_t, \beta_t) := \mathbb{E}(r(S_t, A_t^1, \dots, A_t^n | C_t, \beta_{1:t}))$.
- **Approximate POMDP RL (Step 2):** We define a new notion called Incrementally Expanding Representation (IER) as follows.
  - IER is a 3-tuple $\langle \{\mathcal{X}\}_{k=1}^{\infty}, \tilde{f}, B \} \rangle$.
    - $\{\mathcal{X}\}_{k=1}^{\infty}$ is a sequence of finite sets such that $\mathcal{X}_1 \subsetneq \mathcal{X}_2 \subsetneq \dots \mathcal{X}_k \subsetneq \dots$, and $\mathcal{X}_1$ is singleton say $\mathcal{X}_1 = \{x^*\}$. Let $\mathcal{X} = \lim_{k\to\infty} \mathcal{X}_k$.
    - For any β and z, and $x \in \mathcal{X}_k$, we have that $\tilde{f}(x, \beta, z) \in \mathcal{X}_{k+1}$.
    - B is surjective function that maps $\mathcal{X}$ to the reachable set s.t. $\Pi_t = B(X_t)$.
  - Choose an IER whose components do not depend on unknowns . Construct countable-state MDP Δ with state space $\mathcal{X}$, action space $\mathcal{G}$, dynamics $\tilde{f}$, and reward $\tilde{r}(B(X_t), \beta_t) := \hat{r}_t(\Pi_t, \beta_t)$.
  - Approximate Δ by finite-state MDPs Δ_N with state space $\mathcal{X}_N$.
  - Apply a generic finite-state RL algorithm to learn optimal strategy of Δ_N.

Main Theorem

Let J^* be the optimal performance (reward) and $\tilde{J}$ be the performance under the learned strategy. Then,

$$J^* - \tilde{J} \leq \epsilon_N,$$

where $\epsilon_N = \frac{2\gamma^{\tau_N}}{1-\gamma}(r_{max} - r_{min}) \leq \frac{2\gamma^N}{1-\gamma}(r_{max} - r_{min})$ and τ_N is a model dependent parameter that is $N \leq \tau_N$. Note that error goes to zero **exponentially** in N.

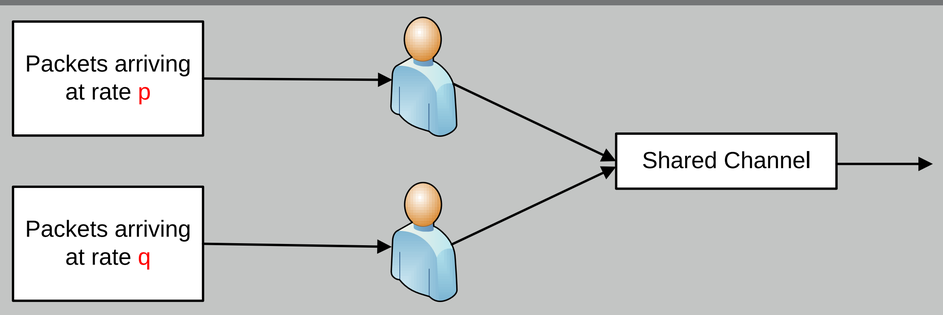
Salient Features

- This approach guarantees ϵ-**optimality** performance.
- It encompasses a large class of multi-agent systems including **delayed sharing**, **control sharing**, **mean field sharing**, etc.
- We combine **common information approach** and **POMDP RL**. Various POMDP RL algorithms may be used in step 2 to obtain different approaches.
- The methodology used in Step 2 is a **novel approach** for POMDP RL.

Multi-Agent RL Algorithm

1. Given $\epsilon > 0$, choose N such that $\frac{2\gamma^N}{1-\gamma}(r_{max} - r_{min}) \leq \epsilon$. Then, construct Δ_N with state space $\mathcal{X}_N$, action space $\mathcal{G}$, dynamics $\tilde{f}$, and reward $\tilde{r}$.
2. At iteration k, pick random prescriptions $\beta_k = (\beta_k^1, \dots, \beta_k^n)$. Agent i takes action a_k^i based on prescription β_k^i and local information m_k^i: $a_k^i = \beta_k^i(m_k^i), \forall i$.
3. Based on taken actions, system incurs a reward r_k, evolves, and generates common observation z_k that is observable to every agent. Agents consistently compute next state $x_{k+1} = \tilde{f}(x_k, \beta_k, z_k) \in \mathcal{X}_N$; otherwise, $x_{k+1} = x^*$.
4. Using a finite-state RL algorithm, learn coordinated strategy according to reward r_k by performing prescriptions β_k at state x_k and transiting to state x_{k+1}.
5. $k \leftarrow k + 1$, and got step 2 until termination.

Example: MABC



- $S_t = (S_t^1, S_t^2) \in \mathcal{S} = \{0, 1\}^2, \quad A_t^i \in \mathcal{A} = \{\text{Do not transmit,transmit}\}$
- Packets arrive at user 1 and 2 according to independent Bernoulli processes with rate p and q that are unknown, respectively.
- Each user transmits if it has a packet (i.e. $A_t^i \in \{0, 1\}$ and $A_t^i \leq S_t^i$).
- $I_t^i = \{S_t^i, A_{1:t-1}^1, A_{1:t-1}^2\}$.
- The objective is to maximize the throughput. Hence, the instantaneous reward is defined as follows: $r(S_t, A_t^1, A_t^2) = A_t^1 + A_t^2 - 2A_t^1 A_t^2$.
- Numerical result: $N = 20, p = 0.3, q = 0.6, \gamma = 0.99$.

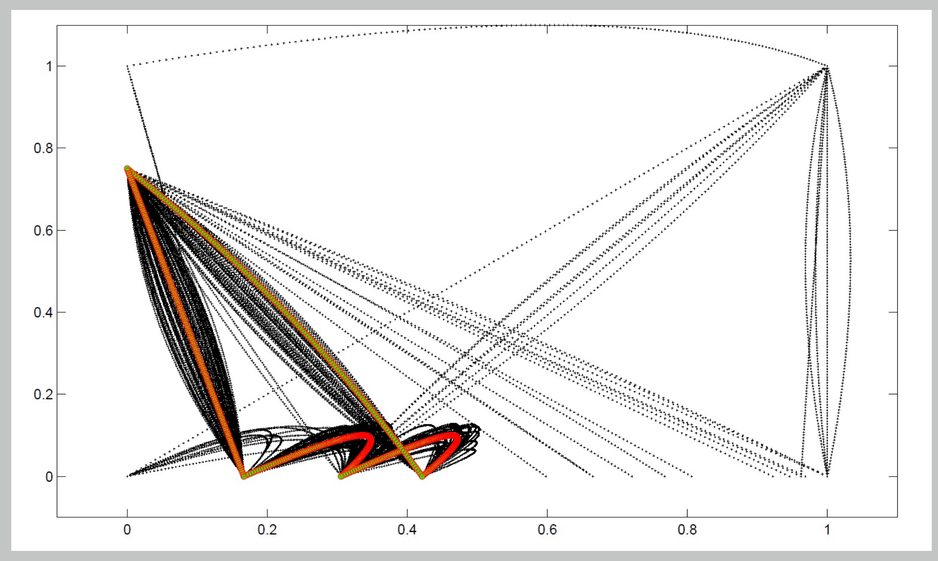


Figure 1: This figure shows the learning process of MDP Δ_N in a few snapshots.

References

[1] Jalal Arabneydi and Aditya Mahajan.
Reinforcement learning in decentralized stochastic control systems with partial history sharing.

American Control Conference (ACC), 2015.

Email:jalal.arabneydi@mail.mcgill.ca
Email:aditya.mahajan@mcgill.ca