Learning Density-Based Correlated Equilibria for Markov Games

Libo Zhang , Yang Chen, Toru Takisaka, Bakh Khoussainov,
rsity of Electronic Science and Technology of China Chengdu, China Michael Witbrock, and Jiamou L

The University of Auckland Auckland, New Zealand

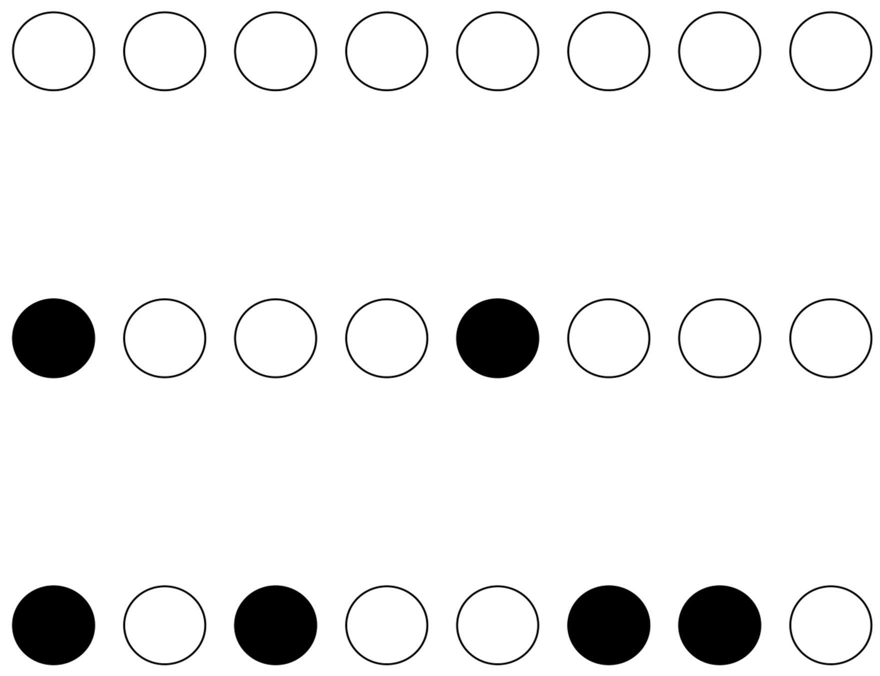
Background

Fundamental. Rather than merely considering reward signals, how do we address **non-reward requirements** such as safety in the AI systems?

Safety
Avoid visiting unsafe (black) states

Frequency
Black states are wished to be visited exactly at a 25% frequency

Fairness
Black and white states are wished to be visited at the same frequency



- ***Markov Games.*** Markov games, also known as stochastic games, are extensions of Markov decision processes to **the multi-agent setting**, where a set of agents act in a stochastic environment, each aiming to maximise its cumulative rewards.
- ***Correlated Equilibrium.*** solution to a Markov is called an equilibrium that amounts to a joint policy where no agent has an incentive to unilaterally deviate to gain rewards. Compared with Nash Equilibrium (NE), correlated equilibrium (CE) captures the **coordination among agents**.
- ***Gap.*** CE to a Markov Game forms a convex set, which is described by reward-based constraints. Existing methods either **modify the reward function**, which is not easy; or cut the CE set by **additional constraints**, which may lead to no solutions.

- ***Objective*** A new CE concept for Markov games which exploits the **state density function** to explicitly capture non-reward requirements without changing the set of all feasible CEs, **Density-based CE (DBCE)**.

Density function

- The **density function** $\boldsymbol{\rho}: \boldsymbol{S} \to \boldsymbol{R}_{\geq 0}$ measures the visitation frequency of states when navigating the environment with a policy.
- Similar to the density function, an **occupancy measure** measures the visitation frequency of state-action pairs given a stationary policy.
- $\rho^\pi(s,a) \ \coloneqq \sum_{t=0}^{\infty} \gamma^t \Pr(s^t = s, a^t = a|\pi, s^0)$ holds under the **bellman-flow constraint:**
- $\begin{cases} \sum_{a\in A} \rho^\pi(s,a) - \eta(s) - \gamma \sum_{s'\in S} \sum_{a\in A} \Pr(s|s',a)\, \rho^\pi(s',a) \\ \rho^\pi(s,a) \geq 0 \end{cases}$
- A **One-to-one correspondence** exists between a policy and an occupancy measure, $\pi(s,a) = \frac{\rho^\pi(s,a)}{\sum_{a'\in A} \rho^\pi(s,a')}$

Density-based CE as optimisation problem

PROBLEM 1. $\min_{f: \mathcal{S}\times\mathcal{A}\to\mathbb{R}} \sum_{s\in S^*} \sum_{\boldsymbol{a}\in\mathcal{A}} f(s,\boldsymbol{a})$ subject to

$$\mathrm{reg}'_f(s,i,a_i,a'_i) \leq 0, \quad \forall i \in [N], s \in \mathcal{S}, a_i, a'_i \in \mathcal{A}_i; \quad (6)$$

$$\mathrm{BFError}_f(s) = 0, \quad \forall s \in \mathcal{S}; \quad (7)$$

$$f(s,\boldsymbol{a}) \geq 0, \quad \forall s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}. \quad (8)$$

Objective function denotes the non-reward requirement.

- (6) Refers to the **CE constraint**;
- (7), (8) Refers to **Bellman Flow constraint**;

Addressing the non-reward requirements by DBCE

- ***Safety:*** $\boldsymbol{min} \sum_{s\in S^*} \boldsymbol{\rho(s)}$ for a set of states S*;
- ***Frequency:*** $\boldsymbol{min}|\sum_{s\in S^*} \boldsymbol{\rho(s) - c}|$ for some constant c;
- ***Fairness:*** $\boldsymbol{min}|\sum_{s\in S_1} \boldsymbol{\rho(s)} - \sum_{s\in S_2} \boldsymbol{\rho(s)}|$ for 2 sets of states

Algorithm and Experiment

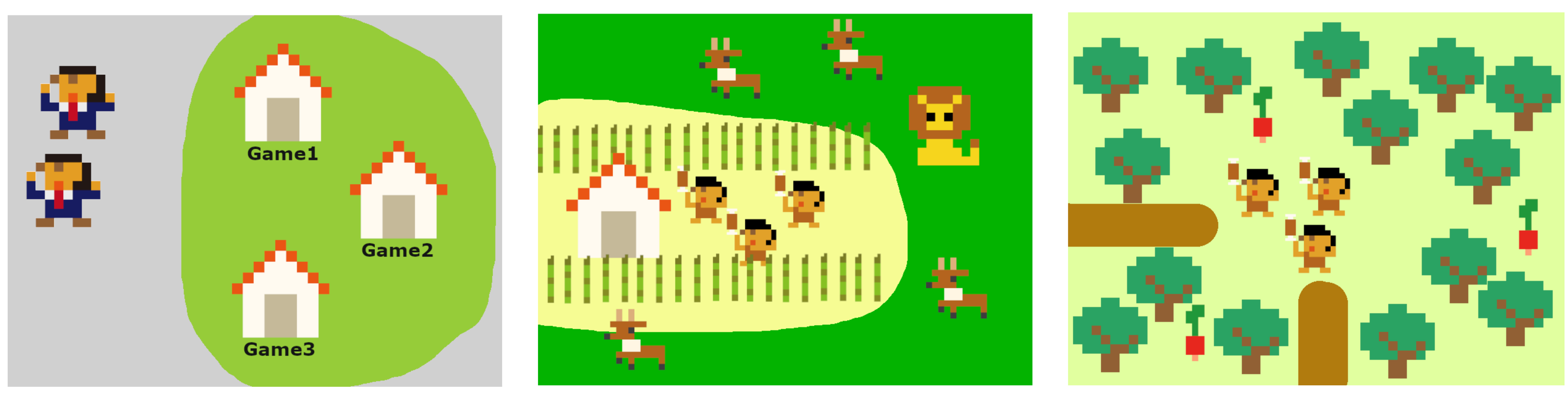
- ***Algo.*** We developed a policy-iteration-based algorithm **Density-Based Correlated Policy Iteration (DBCPI)** to calculate DBCE.
- ***Exp.*** We designed 3 games with cooperative and non-cooperative settings, all games are equipped with animation demonstrations.

We addressed 3 aforementioned non-reward requirements in these 3 games, and tested the ability of DBCPI.

Algorithm 1 Density-Based Correlated Policy Iteration

Input: A Markov game $(\mathcal{S}, \mathcal{A}, P, \{r_i\}_{i=1}^N, \eta, \gamma)$.
Initialisation: Q_i for each $i \in [N]$, learning rate α
$\pi(s,\boldsymbol{a}) \leftarrow f(s,\boldsymbol{a})/\sum_{\boldsymbol{a}'\in\mathcal{A}} f(s,\boldsymbol{a}')$
for each iteration **do**
  $f \leftarrow$ (solution to Prob. 1 with $\{Q_i\}_{i\in[N]}$)
  $\pi(s,\boldsymbol{a}) \leftarrow f(s,\boldsymbol{a})/\sum_{\boldsymbol{a}'\in\mathcal{A}} f(s,\boldsymbol{a}')$
  while Not converge **do**
    Initialise state $s \in \mathcal{S}$
    Observe transition $(s, \boldsymbol{a}, \boldsymbol{r}, s')$
    for each $i \in [N]$ **do**
      $V_i(s') \leftarrow \sum_{\boldsymbol{a}'\in\mathcal{A}} \pi(s',\boldsymbol{a}')Q_i(s',\boldsymbol{a}')$
      $Q_i(s,\boldsymbol{a}) \leftarrow (1-\alpha)Q_i(s,\boldsymbol{a}) + \alpha(r_i + \gamma V_i(s'))$
    end for
    Decay α
  end while
end for
Output: A joint policy π, and $\varphi'(f)$ as the error of π.

Game Demos:



DBCPI Performance: