Problem Formulation-Iterated Prisoner's Dilemma

Anushka

April 2024

1 Problem Requirements

The goal of this project is to develop and analyze strategies for the Iterated Prisoner's Dilemma using reinforcement learning. Unlike classical game theory solutions that focus on equilibrium points, we aim to explore dynamic strategies that could adapt based on an opponent's behavior over time.

The method required for this project is reinforcement learning, specifically focusing on the strategies that emerge when agents are trained under different conditions and assumptions.

2 Payoff Matrix

The strategic interactions within the Iterated Prisoner's Dilemma are governed by the following payoff matrix, which outlines the rewards for each agent based on their chosen actions:

Table 1: Payout scheme for the Prisoner's Dilemma

		Agent 1	
		Cooperate	Defect
Agent 2	Cooperate	(R, R)	(S, T)
	Defect	(T, S)	(P, P)

In this matrix:

- R represents the reward for mutual cooperation.
- T denotes the temptation to defect, i.e., the reward for defecting when the other cooperates.
- S is the sucker's payoff, i.e., the punishment for cooperating when the other defects.
- P stands for the punishment for mutual defection.

This matrix serves as the foundation for the strategic decisions made by the agents in the environment.

World assumptions:

1. Each game consists of multiple rounds, allowing strategies to evolve based on the history of both players' actions.

2. Players have the option to either cooperate or defect in each round, with their choices influencing the immediate reward and the future state of the game.

3. The rewards for each action pair are defined by a payoff matrix, typical for the Prisoner's Dilemma.

3 Problem Formulation

Given the setup of the Iterated Prisoner's Dilemma and the decision-making context based on the actions of the agents, we explore the problem formulation under both Markov Decision Process (MDP) and Partially Observable Markov Decision Process (POMDP) frameworks.

3.1 State Representation in Iterated Prisoner's Dilemma

Three approaches to defining the state space, based on the history length, are:

1. **Entire History:** The state incorporates all previous rounds of interaction between the two agents. This approach provides the most comprehensive view but results in a complex and potentially infinite state space, as each round adds to the history.

2. **Fixed Window:** The state is defined by the outcomes of a fixed number of recent rounds. For a fixed window of 5 rounds, the state space becomes a sequence of outcomes from those rounds. Assuming binary outcomes (Cooperate or Defect) for simplicity, the total number of unique states is 2^{10}, accounting for both agents' actions over 5 rounds. Mathematically, the state at any given time t, S_t, can be represented as a sequence of the last five outcomes:
$$S_t = \{(a_1, b_1)\}$$
where a_i and b_i are the actions taken by Player 1 and Player 2, respectively, in round i.

3. **Current State Only:** With the history length parameter set to 1, the state space in this environment is determined by the outcome of the most recent actions taken by the two agents. Each action can either be to Cooperate (0) or Defect (1), leading to four possible outcomes for the current round:

- Both cooperate (CC): Represented as (0, 0)
- Player 1 cooperates, Player 2 defects (CD): Represented as (0, 1)
- Player 1 defects, Player 2 cooperates (DC): Represented as (1, 0)
- Both defect (DD): Represented as (1, 1)

3.2 Evaluation of State Space Approaches

When modeling environments such as the Iterated Prisoner's Dilemma, the choice of state space representation is pivotal. Each approach, from considering the entire history of states to just the current or a fixed window of recent states, has its implications on the complexity and effectiveness of the learned strategies.

3.2.1 Disadvantages of Using the Entire History as State Space

Utilizing the entire history of interactions as the state space allows for a comprehensive understanding of the game dynamics up to the current point. However, this approach has significant drawbacks:

- **Complexity**: The state space becomes exponentially large as the game progresses, making it computationally infeasible to explore and learn optimal strategies effectively.
- **Non-Stationarity**: The vast state space introduces a level of non-stationarity that can be challenging to model and predict, as the significance of early interactions may diminish over time.
- **Relevance of Historical Actions**: Not all historical actions may be relevant to the current decision-making context, leading to inefficiencies in learning and strategy development.

3.2.2 Limitations of Considering Only the Current State

On the other end of the spectrum, representing the state space solely by the current state simplifies the environment but comes with its limitations:

- **Lack of Context**: This approach fails to incorporate past actions and outcomes, which are crucial for understanding the opponent's strategy and predicting future moves.
- **Strategy Evolution**: It does not capture the evolution of strategies over time, making it challenging to adapt to changes in the opponent's behavior or to exploit patterns effectively.

3.2.3 Advantages of a Fixed Window Approach

A windowed approach to state space representation, particularly with a window of 5, offers a balanced solution:

- **Manageable Complexity**: It limits the state space to a manageable size, allowing for efficient exploration and learning.
- **Stationarity**: This method facilitates modeling the environment as more stationary compared to using the entire history, as it assumes the dynamics influencing decision-making are relatively constant within the window. This assumption simplifies the learning process and can improve the effectiveness of strategies.

3.3 Observations and Memory

In the Iterated Prisoner's Dilemma, an agent's observation space can be framed as its memory of past actions, which can include its own actions, the opponent's actions, or both. The extent and content of this memory determine the agent's ability to make informed decisions.

For agent 1, the memory at time t may consist of:

- A sequence of the last k actions taken by agent 2, denoted as $b_{t-k}, b_{t-k+1}, \ldots, b_{t-1}$, representing the memory of the opponent's actions.
- A sequence of its own last k actions, denoted as $a_{t-k}, a_{t-k+1}, \ldots, a_{t-1}$, representing self-memory.
- A combination of both agents' actions over the last k steps, capturing the full interaction context.

Similarly, for agent 2, the memory can include:

- The last k actions taken by agent 1, expressed as $a_{t-k}, a_{t-k+1}, \ldots, a_{t-1}$.
- Its own last k actions, expressed as $b_{t-k}, b_{t-k+1}, \ldots, b_{t-1}$.
- A combination of both agents' actions over the last k steps.

The observation space is defined not only by the length of the memory k but also by whose actions are being observed:

- Observations of self only provide limited context, focusing exclusively on the agent's own past actions.
- Observations of the opponent only offer insight into the opponent's strategy, without self-referential context.
- Observations of both agents give the most comprehensive view, but the choice of k still determines whether the environment is fully or partially observable.

State History / Observation Type	All Time	Windowed	Most Recent
Self Only	Partial	Partial	Partial
Opponent Only	Partial	Partial	Partial
Both Agents	Complete	Partial	Complete
Decaying	Partial	Partial	Partial
Random	Partial	Partial	Partial

Table 2: Observability Based on State History and Observation Type

3.4 Actions

The agent actions in the Iterated Prisoner's Dilemma are:

- Cooperate (0)
- Defect (1)

3.5 Reward

The reward function is based on the payoff matrix, with the typical values being:

- T (Temptation to defect) - Reward for defecting when the other cooperates.
- R (Reward for mutual cooperation) - Reward for both players cooperating.
- P (Punishment for mutual defection) - Reward when both players defect.
- S (Sucker's payoff) - Reward for cooperating when the other defects.

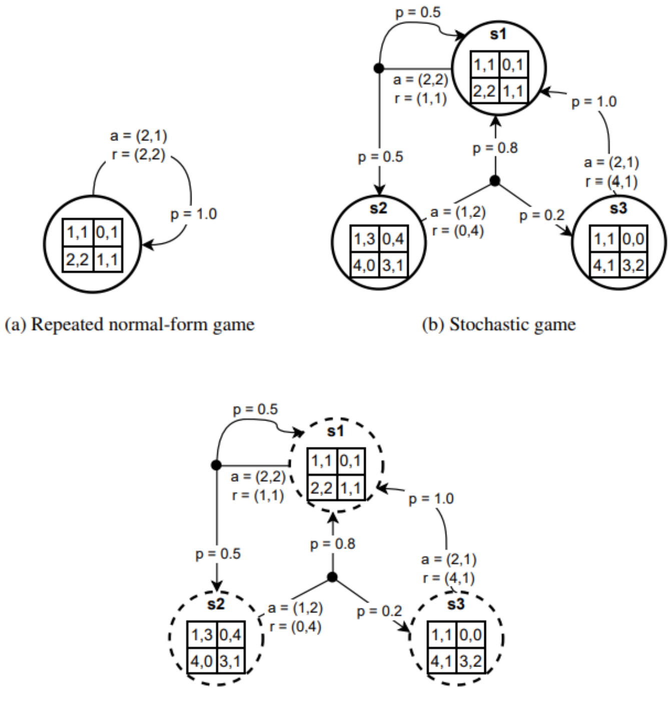
3.6 Problem Parameters

1. Number of rounds - Determines the length of each game.
2. Payoff matrix values (T, R, P, and S) - Influence the strategy dynamics.

3.7 Deciding the Model to Represent the Environment Based on State Representation

3.8 Choosing an Appropriate Model

To aid in understanding the decision-making process in the Iterated Prisoner's Dilemma, consider the following graphical representations of different game types, which illustrate the potential structures and dynamics that may influence model selection:



(a) Repeated normal-form game

(b) Stochastic game

(c) POSG

Figure 1: Graphical representations of different game types: (a) Repeated normal-form game, (b) Stochastic game, (c) Partially Observable Stochastic Game (POSG).

Below is a comparison of different models based on the window length of history considered.

Window Length	**Model**
Entire History	MDP, POMDP, Contextual Bandits
Fixed Window (e.g., Last 5 States)	MDP, POMDP, Contextual Bandits
Current State Only	MDP

Table 3: Model Selection Based on History Window Length

Model	**Mathematical Representation**	**IPD Interpretation**
MDP	(S, A, P, R, γ)	S: States represent combinations of the last 5 actions of both agents. A: Actions are to Cooperate or Defect. P: Transition probabilities are deterministic, based on actions. R: Rewards based on the IPD payoff matrix. γ: Discount factor for future rewards[LBC22]
POMDP	(S, A, P, R, O, γ)	S, A, P, and R as in MDP. O: Set of observations, which could be partial or noisy views of the opponent's last actions. Observations inform the belief about the current state.
Contextual Bandits	(C, A, R)	C: Contexts are the immediate past actions or outcomes, up to the last 5 rounds. A: Actions to Cooperate or Defect. R: Immediate reward received, without considering future states[BB22].

Table 4: Model Comparison in the Context of IPD with Fixed Window Length

References

[BB22] Peter Barnett and John Burden. Oases of cooperation: An empirical evaluation of reinforcement learning in the iterated prisoner's dilemma. 2022.

[LBC22] Baihan Lin, Djallel Bouneffouf, and Guillermo Cecchi. Online learning in iterated prisoner's dilemma to mimic human behavior. In *ICLR 2022 Workshop on Gamification and Multiagent Solutions*, 2022.