Better Understanding of Humans for Cooperative AI through Clustering

Edward Su, William Raffe, Luke Mathieson, YuKai Wang

2024 IEEE Conference on Games (CoG)

s1300037 Tokuda Yuna

Background

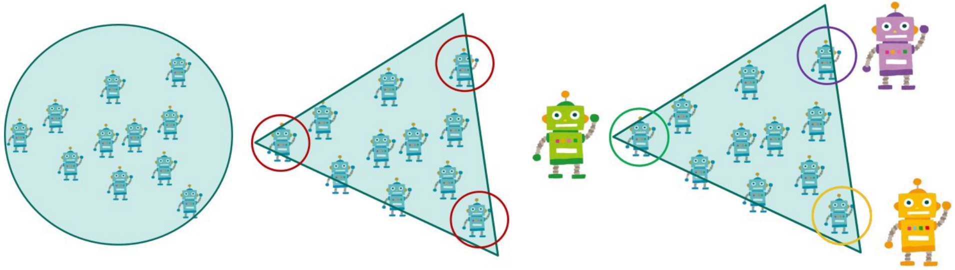
- Cooperative and alignment research has typically lagged behind other machine learning application.
- Creating a framework that leverages Archetypal Analysis.

Introduction

- archetypal analysis (AA)
  - This clusters data around outliers known as ’archetypes’ as opposed to mean data points typical of other clustering techniques.
  - These archetypes are defined as extremal points in the data
  - Create a boundary that encapsulates all other observations.
- AA agent
  - we refer to the RL model capable of cooperating with human partners as the AA agent.
  - At runtime, a linear least-squares method calculates human alignment, and the AA agent stochastically selects and performs a cooperative model action.

Introduction

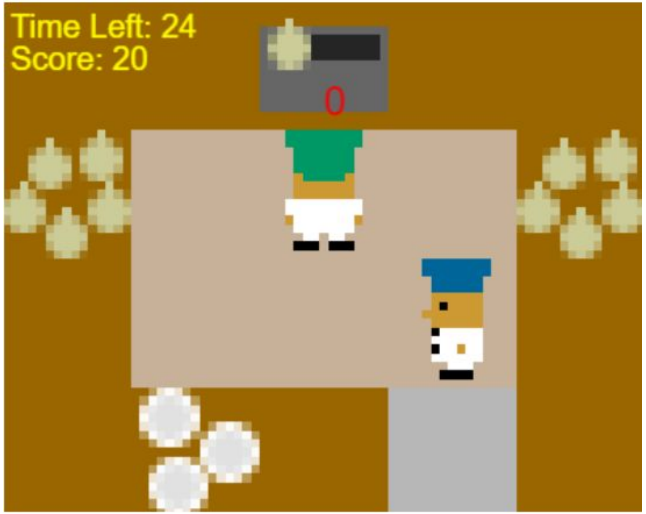
- Create a dataset of game playthroughs
- Extract ’archetypal’ playstyles using AA offline
- Train separate PPO models to cooperate with each archetype
  - called cooperative model



Implementation

Enviroment

- Use ’Overcooked’ environment developed by Carroll et. al.
- This game is complexity, with multiple tasks and others that may force coordination between agents.
- To support our custom AA agent, adding data structures for agent profile information.



Implementation

Archetypal Analysis

- Create playthrough dataset from representative players to identify archetypal playstyles using AA.
- Train RL models optimized to cooperate with each archetypal playstyle.
- Created N number of RL models for archetypal playstyles.
- Generate RL models to represent human playstyles instead of recording playthroughs.
  - This included training five self-play agents using a PPO policy, with checkpoints representing different levels of player skill.

Implementation

About five self-play agents

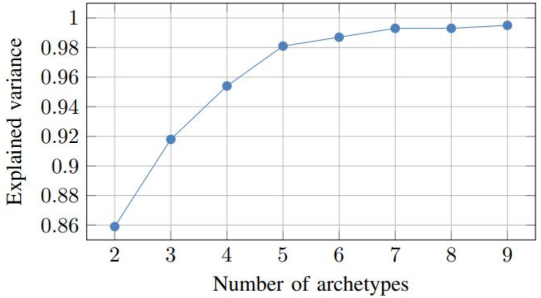
- Use 4 feature
- Initialize with a runtime seed and train for 10,000 steps.
- Create checkpoints at steps 2000, 5000, and 7500.
- perform archetypal analysis on this dataset, which provides us with K number of archetypes
- calculated explained variance to determine the optimal number of archetypes.

Implementation

- Evidently benefits of additional archetypes tapers off after 5
- Since we are using 4 features, the number of archetypes was set to 3.
- The profiles of the archetypes generated with K = 3 archetypes are seen in Table I
- data-points in the feature-set can be expressed as a convex combination of these 3 archetypes shown in Fig. 4.

TABLE I: Archetype Profiles

Arch-type	**Features**			
	objects placed	*objects boiled*	*soup delivered*	*soup placed*
A1	0.000000	0.770684	0.582924	1.000000
A2	0.000000	0.846997	0.834530	0.000000
A3	0.954288	0.000000	0.000000	0.295429



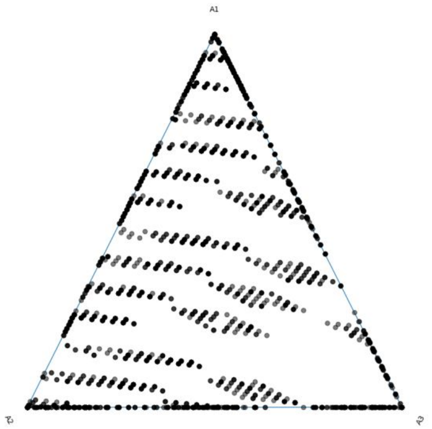


Fig. 4: Datapoints expressed as a convex combination of 3 archetypes.

Implementation

- Afterward,trained the RL model to cooperate with the model closest to each archetype.
- During runtime, the AA agent uses the player's current play data, scaled by time, to observe their alignment with each archetype through a least-squares algorithm.
- The resulting vector is used to probabilistically select the appropriate action, weighted by the player's proximity to each archetype.

Implementation

Benchmarks
this cooperative agent framework against other models

- Self-play agent
- Human-aware PPO agent
- Random action agent
- AA agent

Experimant

- Gather participants and conduct the experiment.
- The host randomly pairs AI agents with participants to play the game.
- Each game lasts 30 seconds, and the host records the score achieved.
- After the game ends, the host switches to an unselected AI agent and continues.

Result

- The standard deviation shows variability in both Human-trained and AA agents, while Random and Self-play agents have noticeably lower values in comparison.(Table3)
- These results reinforced the findings obtained from theTable5.(Table4)

TABLE III: Descriptives

Agent Type	Mean	SD	SE	CoV
AA	47.500	17.701	4.425	0.373
SelfPlay	50.000	12.649	3.162	0.253
Random	28.750	10.247	2.562	0.356
Human_Trained	56.250	15.000	3.750	0.267

TABLE IV: Bayesian Wilcoxon Signed-Rank Test

Measure 1		Measure 2	BF_{10}	W	Rhat
AA	-	SelfPlay	0.527	8.000	1.000
	-	Random	37.910	55.000	1.002
	-	Human_Trained	1.898	10.000	1.000
SelfPlay	-	Random	374.649	105.000	1.018
	-	Human_Trained	1.421	4.000	1.000
Random	-	Human_Trained	377.061	0.000	1.012

Result

- When it came to cooperative ratings, the AA and human-trained models were equivalent, with a BF10 < 3.(Table5)
- Using the same measure, the self-play model performed significantly worse than AA and human-trained models(Table 5)

TABLE V: Post Hoc Comparisons - Agent Type Rating

		$BF_{10,U}$	error %
AA	SelfPlay	0.414	8.686×10^{-7}
	Random	0.414	3.079×10^{-7}
	Human_Trained	0.414	0.019
SelfPlay	Random	0.414	7.208×10^{-7}
	Human_Trained	0.414	2.228×10^{-7}
Random	Human_Trained	0.414	2.155×10^{-7}

TABLE VI: Descriptives Rating

Agent Type	Mean	SD	SE	CoV
AA	3.750	0.856	0.214	0.228
SelfPlay	2.750	0.683	0.171	0.248
Random	1.875	0.719	0.180	0.383
Human_Trained	4.063	0.772	0.193	0.190

TABLE VII: Bayesian Wilcoxon Signed-Rank Test Rating

Measure 1		Measure 2	BF_{10}	W	Rhat
AA	-	SelfPlay	7.729	90.000	1.004
	-	Random	1450.546	120.000	1.007
	-	Human_Trained	0.521	20.000	1.000
SelfPlay	-	Random	16.759	62.000	1.006
	-	Human_Trained	60.059	0.000	1.002
Random	-	Human_Trained	173.140	0.000	1.021

Result

The AA agent matched self-play and human-trained models in scores and cooperated as well as the human-trained model, outperforming self-play and random models. It showed the highest result variance in both metrics.

Conslusion

- AA agents are suitable for AI to adapt to a variety of human partners, are generalizable, and can be easily incorporated into existing model designs.
- Future directions include facilitating smoother strategy transitions, etc.
- In doing so, we have established a promising direction for future research on improving the cooperative capabilities of deep reinforcement learning models.

Thank you for your attention