Greedy when Sure and Conservative when Uncertain about the Opponents

Haobo Fu et al.

Tencent AI Lab, Shenzhen, China

haobofu@tencent.com

ICML-2022

The Investigated Problem

The problem is competing online against unknown opponents for T episodes by sequentially deciding a policy $\pi_{1,j}$ for the main agent at each episode $j, 1 \le j \le T$, such that the regret R_T is minimized:

$$R_T = \max_{\pi_1 \in \Sigma_1} \sum_{j=1}^{T} [u_1(\pi_1, \pi_{2..n,j}) - u_1(\pi_{1,j}, \pi_{2..n,j})], \tag{1}$$

where the expected returns of the main agent when playing π_1 against other opponents $\{\pi_i\}_{i=2}^{n}$ is denoted by $u_1(\pi_1, \pi_{2..n})$. Note that we do not have control over opponent policies $\pi_{2..n,j}$ at each episode j.

Existing Methods

According to the way the main agent policy is determined at each episode during online execution, there are generally three categories of methods from the literature.

- **Playing a fixed policy,** the target of which is usually a Nash Equilibrium (NE) policy in two-player zero-sum games.
- **Opponent modelling within an episode.** The main agent conditions its policy on not only its own observation but also additional information about the opponent, which is either collected or inferred using previous interactions with the opponent within the current episode.
- **Opponent modelling across episodes,** where data from previous episodes is analysed to help decide the main agent policy for the current episode.

Our Assumptions

- We assume **full access to opponent history trajectories** (sequence of observation-action pairs) in previous episodes but *not* the current episode. This is common in human-played games, where we can look back into replays that have full visibility of opponents.
- We assume **a strong and fixed main agent policy** π_1^* is available offline, which hopefully has the best worst-case performance. The policy π_1^* can be obtained by running, e.g., regret minimization algorithms [8, 2] or competitive multiagent Reinforcement Learning (RL) algorithms [3, 6].
- We further assume that **for each opponent we have K different precomputed policies**. We denote the corresponding opponent policy set by $\Pi^{Train} = \{\pi_i^{(k)} | 2 \leq i \leq n, 1 \leq k \leq K\}$.

Greedy when Sure and Conservative when Uncertain

Greedy when Sure and Conservative when Uncertain (GSCU), a new method for competing online against unknown and nonstationary opponents, improves in four aspects:

- introduces a novel way of learning opponent policy embeddings offline.
- trains offline a single best response (conditional additionally on our opponent policy embedding) instead of a finite set of separate best responses against any opponent.
- computes online a posterior of the current opponent policy embedding, without making the discrete and ineffective decision which type the current opponent belongs to.
- selects online between a real-time *greedy* policy and a fixed *conservative* policy via an adversarial bandit algorithm, gaining a theoretically better regret than adhering to either.

An Overview of GSCU

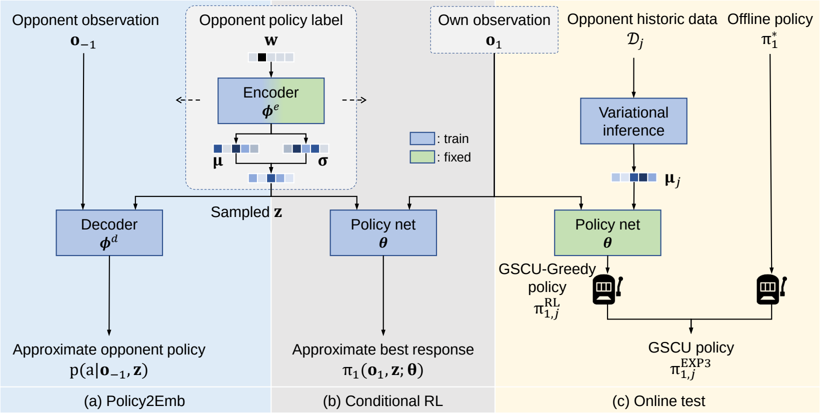


Figure: GSCU has two offline training components: (a) Policy2Emb and (b) Conditional RL. For online test, GSCU employs EXP3 [1] to select between playing greedily ($\pi_{1,j}^{RL}$) and conservatively (π_1^*) against the current opponent.

Policy2Emb: Offline Policy Embedding Learning

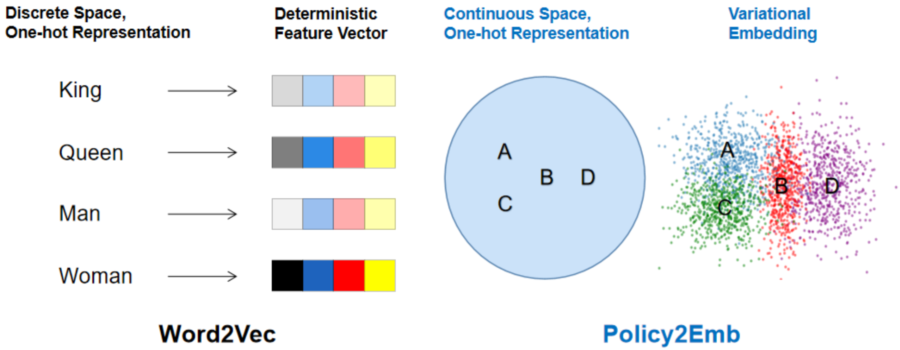


Figure: An illustration of Policy2Emb by making a comparison between it and Word2Vec [4].

Policy2Emb: Offline Policy Embedding Learning

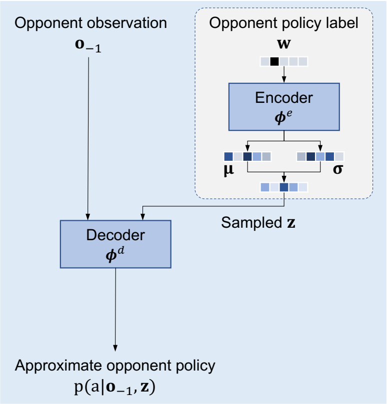


Figure: Policy2Emb employs a Conditional Variational Autoencoder (CVAE) [5] to decouple the learning of policy embedding from the representation learning of other information. The encoder depends solely on an opponent index. For the decoder, a sampled embedding together with an opponent observation produces the probability of an opponent action.

The Offline Conditional RL in GSCU

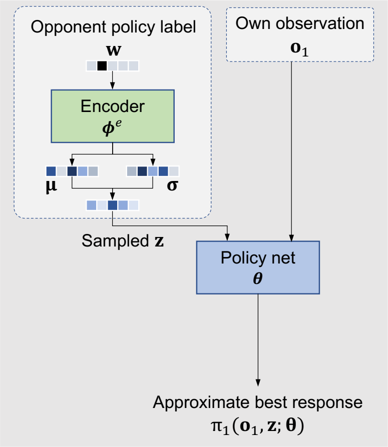


Figure: A conditional (on the opponent policy embedding learned by Policy2Emb) RL is invoked to train a single best response $\pi_1(\mathbf{o}, \mathbf{z}; \boldsymbol{\theta})$ against potential opponents in GSCU.

The Online Bayesian Inference and Policy Selection

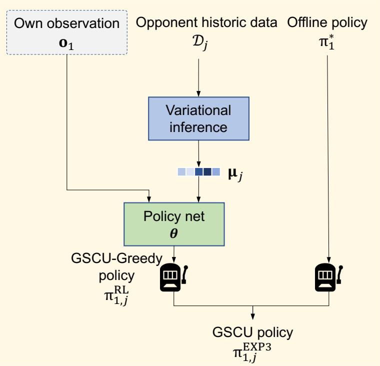


Figure: For online test, GSCU employs EXP3 to select between playing greedily and conservatively against the current opponent. The *conservative* policy is a fixed offline trained policy π_1^*, which hopefully has the best worst-case performance. The real-time *greedy* policy is the offline trained approximate best response $\pi_{1,j}^{\mathsf{RL}} = \pi_1(\mathbf{o}, \boldsymbol{\mu}_j; \boldsymbol{\theta})$, conditioning additionally on an online inferred opponent policy embedding $\boldsymbol{\mu}_j$.

Theoretical Properties of GSCU

The performance of the real-time *greedy* policy $\pi_{1,j}^{RL}$ in GSCU is lower bounded:

$$u_1(\pi_{1,j}^{\text{RL}}, \pi_{-1,j}) \geq u_1(BR(\hat{\pi}_{-1,j}), \hat{\pi}_{-1,j}) - R^{RL}(\hat{\pi}_{-1,j}) - D(\pi_{-1,j} \| \hat{\pi}_{-1,j}). \quad (2)$$

Theorem (The Regret of GSCU's Online Performance)

When $\eta = \min\left\{1, \sqrt{\frac{2\ln 2}{(e-1)\Delta T}}\right\}$, the regret of playing $\pi_{1,j}^{EXP3}$, i.e., GSCU for T episodes is upper bounded:

$$R_T(\pi_{1,j}^{EXP3}) \leq 3.1\sqrt{\Delta T} + \min\left\{R_T(\pi_1^*), R_T(\pi_{1,j}^{RL})\right\},$$

where $R_T(\pi_1^*)$ is the regret of always playing conservatively and $R_T(\pi_{1,j}^{RL})$ is the regret of always playing greedily.

Experimental Study on Kuhn Poker and Predator Prey

- **The goal of the experimental study** is to test the performance of different methods on competing online against unknown and nonstationary opponents. We also validate the effectiveness of each component in GSCU.
- **Training Protocols**. Each method has access to only the opponent policy set Π^{Train}.
- **Test Protocols**. For online test, we create four types of sequences of opponents: "seen", "unseen", "mix", and "adaptive". For the "seen" sequence, we randomly sample an opponent from Π^{Train} every M episodes. The same procedure applies to the "unseen" and "mix" sequences, except that we sample opponent policies from Π^{Test} ($\Pi^{Test} \cap \Pi^{Train} = \emptyset$) and $\Pi^{Train} \cup \Pi^{Test}$ respectively. For the "adaptive" sequence, the opponent continuously updates its own policy using PPO.

The Online Performance against Unknown Opponents

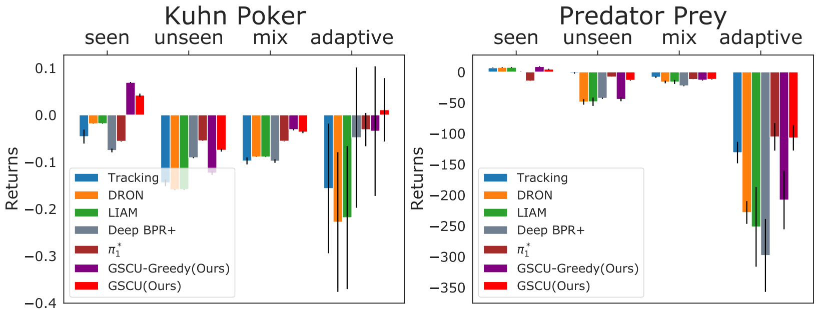


Figure: The average returns of different methods competing online against different types of sequences of opponents. GSCU demonstrates more robust performance against a wide range of unknown and nonstationary opponents.

The Online Performance against Unknown Opponents

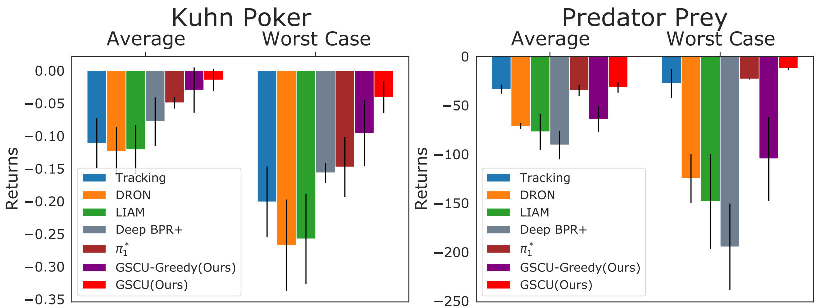
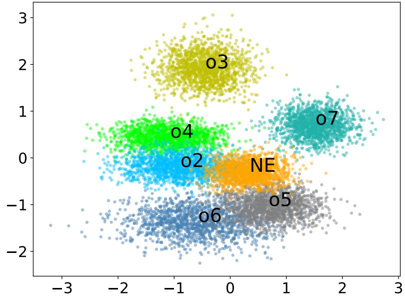


Figure: GSCU performs the best, in terms of the average and worst-case returns across the 4 settings of online opponents: "seen", "unseen", "mix", and "adaptive".

The Learned Embeddings by Policy2Emb



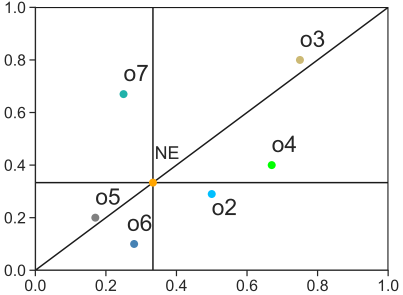


Figure: The policy embeddings learned by Policy2Emb in Kuhn poker (left) and the true policy space (right). The learned policy embedding space is well structured in the sense that it is almost a mirror image of the ground truth.

The Performance of Conditional RL in GSCU

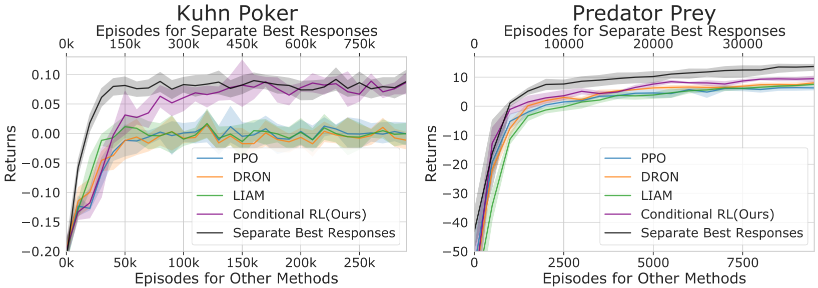


Figure: The offline RL training process of different methods. The better performance of GSCU suggests that the opponent policy embedding learned by Policy2Emb facilitates the effective learning of a single approximate best response against different opponents.

The Performance of Online Bayesian Inference in GSCU

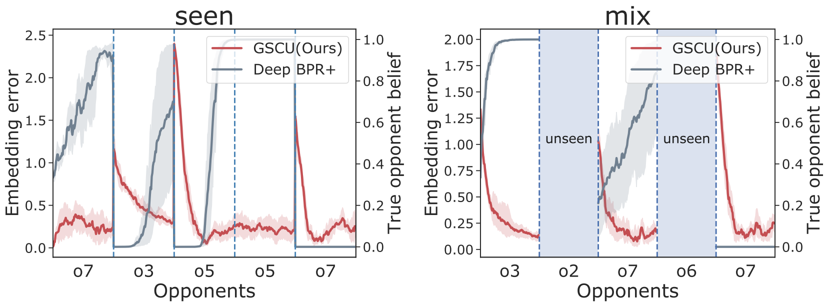


Figure: Online inference performance of GSCU and Deep BPR+ [7]. The embedding error of GSCU decreases steadily on opponents from Π^{Train} in both "seen" and "mix" sequences. Yet, Deep BPR+, which calculates a categorical distribution over Π^{Train}, sometimes fails to identify the right opponent in time.

Conclusion

- This paper develops a new approach, i.e., GSCU for competing online against unknown opponents.
- Within GSCU, we introduce Policy2Emb, a novel way of learning opponent policy embeddings offline, which is of independent interest to policy representation learning.
- GSCU trains offline a single best response, conditional on the opponent policy embedding learned by Policy2Emb.
- GSCU computes online a posterior of the current opponent policy embedding, without making the discrete and ineffective decision which type the current opponent belongs to.
- GSCU selects online between a real-time *greedy* policy and a fixed *conservative* policy via EXP3, gaining a theoretically better regret than adhering to either.

Thank you for your attention!

References I

[1] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire.
The nonstochastic multiarmed bandit problem.
SIAM Journal on Computing, 32(1):48–77, 2002.

[2] Haobo Fu, Weiming Liu, Shuang Wu, Yijia Wang, Tao Yang, Kai Li, Junliang Xing, Bin Li, Bo Ma, Qiang Fu, et al.
Actor-critic policy optimization in a large-scale imperfect-information game.
In *International Conference on Learning Representations*, 2021.

[3] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E Taylor.
A survey and critique of multiagent deep reinforcement learning.
Autonomous Agents and Multi-Agent Systems, 33(6):750–797, 2019.

References II

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.
Distributed representations of words and phrases and their compositionality.
Advances in Neural Information Processing Systems, 26, 2013.

[5] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert.
An uncertain future: Forecasting from static images using variational autoencoders.
In *European Conference on Computer Vision*, pages 835–851. Springer, 2016.

[6] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar.
Multi-agent reinforcement learning: A selective overview of theories and algorithms.
Handbook of Reinforcement Learning and Control, pages 321–384, 2021.

References III

[7] Yan Zheng, Zhaopeng Meng, Jianye Hao, Zongzhang Zhang, Tianpei Yang, and Changjie Fan.
A deep bayesian policy reuse approach against non-stationary agents.
In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 962–972, 2018.

[8] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione.
Regret minimization in games with incomplete information.
Advances in Neural Information Processing Systems, 20:1729–1736, 2007.