Nash Equilibria in Bayesian Games for Coordinating with Imperfect Humans

Shray Bansal
Georgia Institute of Technology
sbansal34@gatech.edu

Miguel Morales
Georgia Institute of Technology
mmorales34@gatech.edu

Jin Xu
The Ohio State University

Ayanna Howard
The Ohio State University

Charles Isbell
Georgia Institute of Technology

Abstract: Coordinating with humans is a challenge for AI agents, even when there are common goals because of both the heterogeneity of human behavior and an assumption of optimal decision-making. Given a large corpus of human-human interactions, we can train models to learn the best responses; however, even without such data, we can still use our knowledge of biases and limitations in humans to construct a technique that can coordinate with humans. In this paper, we present a game-theoretic method that incorporates different partner behavior in the form of different reinforcement learning models trained with suboptimal partners. We also show how to augment this approach with human data. We perform experiments in the fully-cooperative game Overcooked [1]. Our results show a 40% improvement over the baseline of a self-play agent and a 5% improvement over the baseline when using the data of human-human interactions.

1 Introduction

Multi-agent interactions are common in everyday life. As artificial agents share more of our social spaces they must be able to manage these interactions well. Even though past work in multi-agent systems focuses on competitive games, most of our interactions do not. Recent work has shown that cooperation has its own unique challenges and progress in competition does not always transfer [1].

We make two key observations: (1) human decision-making is not optimal, so assuming such can lead to a breakdown in coordination, and (2) human behavior is not homogeneous so the AI agent must be able to adapt to the the behavior of a specific agent to interact effectively. Prior work, *e.g*. Population Based Training (PBT) [2], has attempted to induce diverse behaviors by training with multiple models, but uniformly training over diverse policies may not capture human behavior and these models can converge to a single policy due to catastrophic forgetting. While others, *e.g*. [1], have shown how to utilize human data for interaction, it does not address the wide variety of applications where human data is unavailable due to costs or safety concerns.

Our main contribution then is to model interaction as a two-agent Bayesian game where the AI agent adapts to its partner by considering a small set of policies. We create different behaviors by training RL agents with intentionally slower partners to capture a human's suboptimality. We model the different policies as discrete states of a Hidden Markov Model(HMM) which is updated using observations of the human's behavior. The inferred agent type is used to sample action trajectories to create a normal-form game that is solved to find the actions to execute for the agent.

Using the fully-cooperative game of Overcooked [1], we first pair the AI agent with an agent trained by self-play, and then pair the AI agent with a proxy model of a human trained by imitation learning. Our approach is able to significantly outperform the baseline in case human data is unavailable when paired with the proxy human model. Also, the performance of our approach and the baseline increase

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

considerably when this data is made available. Next, we summarize relevant prior work (Section 2), formalize the problem and describe our solution (Section 3), present results (Section 4) and conclude with plans for future work (Section 5).

2 Related Work

There has been considerable progress in solving zero-sum games such as Backgammon [3], Go [4], Dota 2 [5], and StarCraft II [6]. Curriculum learning strategies, such as self-play and league-play, are often used to guide training [7]. In self-play, a primary agent plays against recent versions of itself. In league-play, the primary agent plays against a diverse set of continually adapting strategies.

Unfortunately, these methodologies don't transfer well to cooperative games, as shown by [1], especially when one of the decision-making agents is a human, for a number of reasons. First, equilibria in two-player zero-sum games are minimax/maximin strategies suitable for self-play, but that is not the case for cooperative or general-sum games [8]. In addition, when paired with a humans only during test time, agents trained through self-play experience out-of-distribution input data [9].

Transfer learning is a possible solution, and several methods have been proposed in recent years [10, 11, 12, 13]. In our case, we deal with a specific form of transfer learning referred to as zero-shot domain adaptation, in which agents do not have access to the target domain for training, thus zero-shot learning [14]. In addition to novel training methodologies, several deep multi-agent algorithms have been introduced in recent years. This includes methods that exploit neural network architectures and standard RL agent components [15, 16], agents that use communication mechanisms [17, 18], and methods that have an awareness of other learning agents [19, 20, 21]. However, there is space for more progress in developing agents that can adapt to human partners.

Recently, game theory has received interest in enabling coordination in general-sum multiagent scenarios [22, 23, 24, 25, 26]. The Nash equilibrium has been used to plan effectively among different agents [22, 27, 24, 28]. Others have adapted to the heterogeneity in human behavior by inferring personality [22], social compliance [24] and strategy preference [28]. We also model the human agent's adherence to different behavior types derived from different levels of task competence.

3 Method

We model multi-agent interaction as a Bayesian game represented as a tuple (N, S, A, θ, p, r) [29]. Here, N is a set of agents, S is a set of states, $A = A_1 \times ... \times A_N$ where A_i is the set of discrete actions available to player i, $\theta = \theta_1 \times ... \times \theta_n$ where θ_i is the type space for agent i, p is a prior over these types, $p(\theta_i) \geq 0, \sum p(\theta_i) = 1$, and $r = (r_1, ..., r_n)$ where $r_i : S \mapsto \mathbb{R}$ is the reward for agent i. $p \in \Delta(\theta)$

In our scenario, each action profile $a \in A$ is a sequence of control inputs for all the agents generated by sampling a policy $p(c|s) = f(s)$ where c, s refer to control input and state, respectively. The agents are modeled as having multiple types θ, each of which has an associated policy for both agents. The policies here are neural networks trained via RL or imitation learning.

3.1 Bayes-Nash

Figure 1 shows our framework. At a timestep, t, the type θ with the maximum probability is used to select the policy models for both agents. We sample the policies from state s_t to generate action sets of size k, $|A_i| = k$. Next, we compute the accumulated reward for every pair of the $k \times k$ actions by simulating them. We then represent this as a normal form game for a given type and find the corresponding Nash equilibria. A set of actions is a Nash Equilibrium, if no agent has incentive to choose a different action for themselves given that all the other agent's actions are fixed [29]. If more than one equilibrium is present, we select one uniformly at random from the set of Pareto-optimal equilibria and execute the corresponding action for the robot.

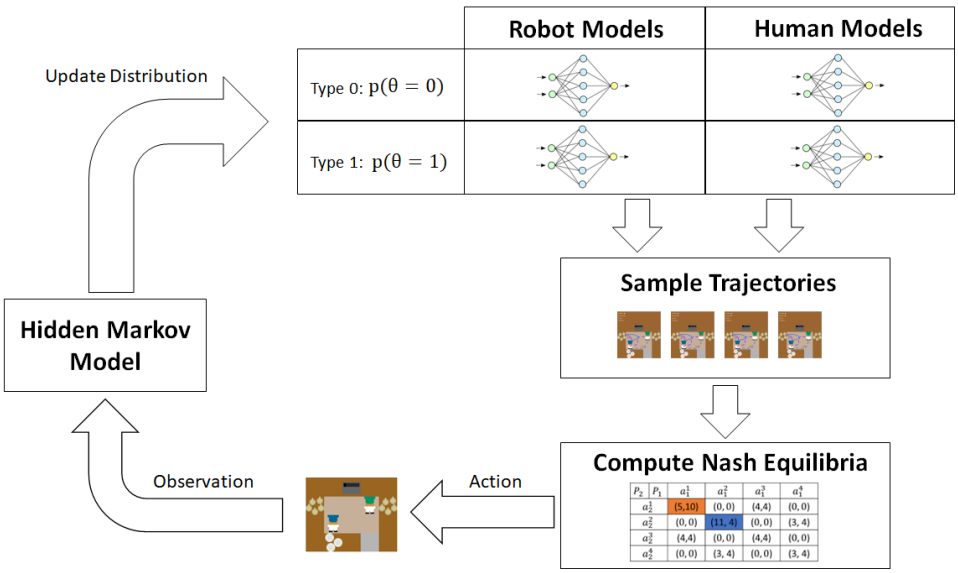

Figure 1: Bayes-Nash. Our model has two types, θ, corresponding to RL models for the robot and the human. We sample trajectories for a type and use them to construct a normal-form game. We select the robot's actions based on the Nash equilibria. We observe the human's action to update the belief of the Hidden Markov Model over human types and repeat this cycle.

The type-space of the agents is modeled by a Hidden Markov Model (HMM) with a discrete latent state that corresponds to the different policy models. The state is assumed to be full-observable and the HMM is updated at every timestep with the evidence being composed of past human states.

3.2 RL Agents

No human data. In the absence of human interaction data, we train two kinds of policies using Proximal Policy Optimization (PPO) [30]. The first uses self-play, referred to as PPO$_{SP}$, where the agent learns to play the game by interacting with a copy of itself. The second, PPO$_{noop}$, is trained by partnering with a model that samples its control from PPO$_{SP}$ with a probability $(1-p)$, and takes a noop action otherwise. This model was designed to capture the agent's response to suboptimal behavior. We trained four such models, with $p = \{0, 0.25, 0.5, 0.75, 1\}$.

Using human data. We utilize disjoint human-human interaction data to train two behavioral cloning models, BC, and H$_{Proxy}$(refer to [1] for more details). We train an RL model, PPO$_{BC}$, by interacting with BC. H$_{Proxy}$ was used only for testing the performance of these models.

4 Experiments

We use the Overcooked environment introduced by [1] because it includes both strategy and motion coordination challenges while the action space works well with deep RL algorithms. In this environment, two agents work together to cook and serve soup, with the goal being to serve as many soups as possible. We use the first layout from [1] where the agents need to put 3 onions in a pot, leave them to cook for 20 timesteps, place this soup in a dish, and serve it. The challenge comes from the agents learning to navigate the map and interact with objects while adapting to their partner's strategy. There are 6 available actions - up, down, right, left, noop, and interact. In the game, an action profile a is a fixed-length trajectory of these discrete actions. We sample $k = 4$ trajectories for each agent which leads to a 4×4 normal-form game. Rewards for the agent are accumulated over the whole trajectory. Both agents receive a joint team-reward of 20 for each dish served by either of the agents. Each agent also receives a smaller individual reward based on performing useful intermediate actions like delivering an onion or putting the soup in a dish. We use this shaping reward for the normal-form game as well as the RL training but it is decayed to 0 during the RL

(a) (b)

Figure 2: Rewards over episodes of 400 timesteps paired with (a) a self-play model and (b) a human proxy model, with standard error across 5 random seeds.

training. If we do not find a Nash equilibrium at a given timestep we randomly select a trajectory to execute, this happens very rarely in our domain($< 0.1\%$). We used the same experimental setup as [1] including the random seeds.

4.1 Results

We include three different types of agents: PPO, PPO$_{noop}$, and Bayes-Nash. If human data is unavailable, PPO refers to PPO$_{SP}$ and Bayes-Nash includes two types- PPO$_{SP}$ and PPO$_{noop}{}^{p=0.75}$. If human data is available, PPO refers to PPO$_{BC}$ and Bayes-Nash includes two types- PPO$_{BC}$ and PPO$_{noop}{}^{p=0.75}$. The parameters of the HMM in Bayes-Nash are kept constant under both conditions. We measure the accumulated joint team-reward over episodes of 400 timesteps.

Experiment 1. PPO$_{SP}$ acts as the human and we compare team performance across the different models. As shown in Figure 2a, when the agents were trained without human data, Bayes-Nash agent performed slightly better than the PPO$_{SP}$ agent, and considerably better than the PPO$_{noop}$ agent. Here, we did not observe considerable improvements when training the agents with human data.

Experiment 2. The H$_{Proxy}$ model acts as the human and we compare team performance. As shown in Figure 2b, when agents were trained without human data, Bayes-Nash outperformed both the PPO$_{SP}$ and PPO$_{noop}$ by over 40%. The Bayes-Nash agent was able to adapt to the human model while only choosing between the behaviors of the other two models without access to any human data. We believe this is due to its ability to switch to PPO$_{noop}$ when the stubbornness of PPO$_{SP}$ causes deadlocks as humans deviate from its expectations. In our experiments, Bayes-Nash mostly relies on PPO$_{SP}$ and rarely uses PPO$_{noop}$. When we include access to human data through PPO$_{BC}$, both models improve their coordination considerably, highlighting the importance of human data for enabling coordination. Bayes-Nash performs only slightly better than the baseline PPO$_{BC}$ with the addition of this data.

5 Conclusion

We present a method for interactive decision-making that uses multiple RL agents in a game-theoretic framework and infers a distribution over these models based on the history of its interaction. We also showed how to increase behavior diversity in RL agents by making their partner intentionally slow during training. While recent work has shown us how to incorporate human interaction data to achieve coordination with human partners through imitation learning, it fails to address applications where collecting a large amount of data from humans is infeasible. Our results show that incorporating knowledge of game theory and human capability can help us to develop agents that coordinate well with people even when human data in unavailable. As a next step, we plan to study mechanisms that allow us to iteratively improve these models with few human interactions. Also, to test generalizability, our plan is to perform a user-study where these algorithms can interact with humans as well as investigating the adaptibility of these ideas to other domains.

Acknowledgments

We would like to thank Amirreza Shaban, Michael Littman, Ashley Edwards, and Nitish Sontakke for helpful discussions.

References

[1] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan. On the utility of learning about humans for human-ai coordination. In *Advances in Neural Information Processing Systems*, pages 5175–5186, 2019.

[2] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

[3] G. Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219, 1994.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587): 484–489, Jan. 2016.

[5] OpenAI, :, C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. d. O. Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. Dota 2 with large scale deep reinforcement learning, 2019.

[6] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, Nov. 2019.

[7] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone. Curriculum learning for reinforcement learning domains: A framework and survey. 2020. doi:10.48550/ARXIV.2003.04960. URL https://arxiv.org/abs/2003.04960.

[8] A. Lerer and A. Peysakhovich. Learning social conventions in markov games. *CoRR*, abs/1806.10071, 2018. URL http://arxiv.org/abs/1806.10071.

[9] A. Lazaric. Transfer in reinforcement learning: A framework and a survey. In *Adaptation, Learning, and Optimization*, Adaptation, learning, and optimization, pages 143–173. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[10] I. Higgins, A. Pal, A. Rusu, L. Matthey, C. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, pages 1480–1490. PMLR, 2017.

[11] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *CoRR*, abs/1703.06907, 2017. URL http://arxiv.org/abs/1703.06907.

[12] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

[13] K. Lee, K. Lee, J. Shin, and H. Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. *arXiv preprint arXiv:1910.05396*, 2019.

[14] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018.

[15] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[16] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[17] M. Hausknecht and P. Stone. Grounded semantic networks for learning shared communication protocols. 2016.

[18] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[19] J. N. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

[20] T. Willi, A. H. Letcher, J. Treutlein, and J. Foerster. Cola: consistent learning with opponent-learning awareness. In *International Conference on Machine Learning*, pages 23804–23831. PMLR, 2022.

[21] S. Zhao, C. Lu, R. B. Grosse, and J. N. Foerster. Proximal learning with opponent-learning awareness. *arXiv preprint arXiv:2210.10125*, 2022.

[22] S. Bansal, J. Xu, A. Howard, and C. Isbell. Planning for human-robot parallel play via bayesian nash equilibrium inference. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020. doi:10.15607/RSS.2020.XVI.042.

[23] S. Bansal, A. Cosgun, A. Nakhaei, and K. Fujimura. Collaborative planning for mixed-autonomy lane merging. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[24] L. Peters, D. Fridovich-Keil, C. Tomlin, and Z. Sunberg. Inference-based strategy alignment for general-sum differential games. In *AAMAS '20*. International Foundation for Autonomous Agents and Multiagent Systems, 2020. URL https://github.com/lassepe/AAMAS2020-GameInference-Paper/blob/master/submission/ibsa-camera-ready-aamas2020.pdf.

[25] V. Gabler, T. Stahl, G. Huber, O. Oguz, and D. Wollherr. A game-theoretic approach for adaptive action selection in close proximity human-robot-collaboration. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2897–2903. IEEE, 2017.

[26] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and Systems*, 2016.

[27] A. Turnwald and D. Wollherr. Human-like motion planning based on game theoretic decision making. *International Journal of Social Robotics*, 11(1):151–170, 2019.

[28] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus. Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences*, 116(50):24972–24978, 2019.

[29] K. Leyton-Brown and Y. Shoham. Essentials of game theory: A concise multidisciplinary introduction. *Synthesis lectures on artificial intelligence and machine learning*, 2(1):1–88, 2008.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.