Learning Collaborative Task Allocation with Human Decision-Makers

Xi Lin* **Yuchu Wang** **Haochen Wu**
Department of Mechanical Engineering
University of Michigan
Ann Arbor, MI, 48109

Abstract

Humans and autonomous agents can collaborate as a team to improve capabilities of existing human-operated systems, which motivates efficient learning for autonomous agents to collaborate with real human decision-makers. This project presents an iterative learning framework for human-autonomy teaming with two learning objectives that benefit from each other: 1) training a human-like agent from human limited demonstrations and 2) improving autonomous agents to collaborate with human. Using the developed algorithm, we quantitatively investigate the effect of human demonstrations on team performance and elucidate the benefits and limitations of learning with human decision-makers.

1 Introduction

Recent developments in human-autonomy collaboration have been focusing on either learning from demonstrations, autonomy playing an assistant role in practice, or task allocation among autonomous agents without the presence of real human decision-makers. Autonomous agents are capable of handling dangerous tasks but limited in real-time computational power, and making ethical decisions, while humans are more adaptive and creative problem-solving skills but are limited in terms of cognitive loads and fatigue. Integration of humans and autonomous agents enables teams to survive and function in complex environments by combining the efficiency of autonomy and the characteristics of humans. Therefore, the primary goal of the project is to train autonomous agents that efficiently learn collaborative task allocation decisions and understand human preference in operation scenarios with limited human demonstrations. In human-autonomy teaming or HAT, team members have equal authorities for planning and executing tasks, which consequently helps reduce managing load for human and thereafter improve the team performance. This would require the autonomous agents to make synchronized decisions along with human.

Many researches have addressed the *dynamic task allocation* within a team of agents using centralized optimization-based coordination (Choi et.al., 2010) and decentralized Markov Decision Process (Puterman, 1994)-based formulation (Omidshafiei et.al., 2017) with deep learning approach. Most of current practices do not consider team coordination scenarios with human-in-the-loop.

The field of *inverse reinforcement learning* or IRL (Ramachandran & Amir, 2007; Abbeel & Ng, 2004), demonstrations by experts are collected for imitating human risk management (Liu et.al., 2019) and learning human motion behaviors (Asfour et.al., 2006). In HAT, such trained artificial agents that imitates human behaviors can be beneficial in terms of reproducibility and efficiency without involving actual humans when interacting with autonomous agents in simulations.

*{bexilin, yuchuw, haochenw}@umich.edu

Cooperative Task Allocation Framework for HAT. We propose, therefore, an iterative learning framework for autonomous agents in human-autonomy teams to understand human behaviors from demonstrations and synergistically collaborate with human-like agents that imitates real human decision-makers. The framework has two learning objectives. The first objective is to train the human-like agent from demonstrations by playing with other autonomous agents, and the second objective is to improve the decisions of autonomous agents by interacting with the trained human-like agent. The two objectives do not stand by themselves since one-step improvement may not capture the full behavior expectation from humans. Instead, the two objectives can be achieved by learning iteratively from each other. More detailed discussion of the framework will be discussed in Section 3.2.

The rest of the report is organized as following: Section 2 provides the fundamental information on the decision-making processes for autonomy and human. Section 3 presents our approach to the problem with formal formulation and framework explanation. Our experiments studying the effect limited demonstrations on the team performance are discussed in Section 4. Section 5 concludes the project and suggests the challenges encountered and future works.

2 Related Work

The proposed framework consists of two learning algorithms: Q-Learning for autonomous agents and Bayesian Inverse Reinforcement Learning for the artificial human-like agent. The backgrounds for these two algorithms are briefly discussed in this section and the extensions for human-autonomy teaming are presented in Section 3.

Q-Learning (Watkins & Dayan, 2004). With full observation of the environment and full knowledge of the environment state transition information, the reinforcement problem on dynamic task allocation can be modeld as Markov Decision Process (Puterman, 1994) or MDP and the exact solutions can be found with model-based algorithms. However, the computational cost increases exponentially with increasing state and action spaces. In HAT, all agents have their own decision processes in a decentralized manner for reducing action space and easy maintaining reward functions. The state transition information for the whole team then becomes unknown. Therefore, Q-Learning is chosen here as the model-free algorithm to provide computationally efficient approximation of the optimal decisions.

Bayesian Inverse Reinforcement Learning or BIRL (Ramachandran & Amir, 2007). The idea of BIRL is to train an artificial agent that reflects the desired human policy from demonstrations by updating the reward function using Bayesian approach. An efficient Monte Carlo Markov Chain sampling algorithm called *PolicyWalk* is developed to infer the probabilities of the reward function conditioned on the demonstrations.

3 Methods

3.1 Problem Formulation

In this project, we address a dynamic task allocation problem in a human-autonomy teaming scenario, which could be described by the decentralized reinforcement learning setting as demonstrated in **Fig. 1**: 1) A team consisting of autonomous agents and a human is assigned several tasks to solve; 2) Each team member has a given capability value of dealing with each task, and it always makes its own decision on choosing the task; 3) Each task is described by a severity level, which is dynamically changing according to the respond of the team; 4) The whole process ends when the severity level of all tasks reach the minimum level. 5) Team members take actions and receive reward and observation information from the environment. We want to train the autonomous agents so that they could collaborate well with human decision-makers in the training scenario.

The problem is formulated as a Decentralized Markov Decision Process (Dec-MDP) model and solved via reinforcement learning. In this model, 1) A state s is described by a combination of severity levels of all tasks; 2) An action a is described by a combination of task choice of each

team member; 3) The given transition model $P_a(s, s')$ decides the probability of transition from the state s to the state s' when applying the action a; 4) The given reward function $R_a(s, s')$ determines the reward of the state s' when it's achieved by applying the action a to the state s. The goal of the problem is to maximize the overall team performance which is defined as the expectation of future culmulative discounted reward defined as $\mathbb{E}[\sum_{t=0}^{t=h} \gamma^t R_a^t(s_t, s_{t+1})]$, where γ is the discounting factor and h is time step.

Fig. 1 Demonstration of Decentralized Reinforcement Learning

Then according to Bellman's equation, the utility value of the state s, $U(s)$, and the Q value of the state-action pair (s, a), $Q(s, a)$, the quality of a state-action pair, could be expressed as equation (1) and (2), where γ is the discounting factor.

$$U(s) = \sum_{s'} P_a(s, s')(R_a(s, s') + \gamma U(s')) \tag{1}$$

$$Q(s, a) = \sum_{s'} P_a(s, s')(R_a(s, s') + \gamma max_a Q(s', a)) \tag{2}$$

The optimal policy can be easily found by taking the action with the highest Q value given a state as descried in equation (3).

$$\pi^*(s) = argmax_a Q(s, a) \tag{3}$$

We have two assumptions for the problem: 1) The operation state is fully observable, and we have full knowledge on the capability of each team member; 2) The human has more knowledge on the envrionemnt, where the transition probability model is only known to the human and not available to autonomous agents.

3.2 Cooperative Task Allocation Framework for HAT

As mentioned before, the proposed iterative framework has two learning objectives that benefit from each other. The first objective is to train the human-like agent from demonstrations, and the second objective is to improve the decisions of autonomous agents by interacting with the trained human-like agent. In such way, the human subject can provide better demonstrations that satisfies his/her expectation while the autonomous agents are gradually learning to collaborate and understand human's preference. To train agents that have good collaboration with the human, we simulate the human behavior by training a human-like agent from real human demonstration data and includes it into the training of other agents. The general framework of our method is shown in **Fig. 2** and described below.

Initial Training. In the first step, based on the problem model described in the last section, we perform an initial Q-learning on autonomous agents playing with a default agent that would be replaced by the trained human-like agent. The output is the optimal policies $\{\pi_{A_i}\}$ for autonomous agents in the teamwork manner without human.

Iterative Process. Secondly, we collect human demonstrations data $\{D_k\}$ by playing the game with the agents on our own. Then, with human demonstrations, a human-like agent is trained through inverse reinforcement learning based on the PolicyWalk method proposed by Ramachandran &

Amir, (2007), where the policy $\{\pi_H\}$ and reward function $\{R_H\}$ of the human-like agent are gradually updated to approach human performance. In the next step, we add the human-like agent into the team and perform Q-learning again and only update the policies of the autonomous agents $\{\pi_{A_i}\}$, while keeping that of the human-like agent $\{\pi_H\}$ fixed.

Stopping Criterion. Lastly, we evaluate the collaboration performance of the team and output the trained agents if the team performance has converged, otherwise we repeat the iterative process by going back to collect another set of human demonstration and improving the human-like agent and autonomous agents again.

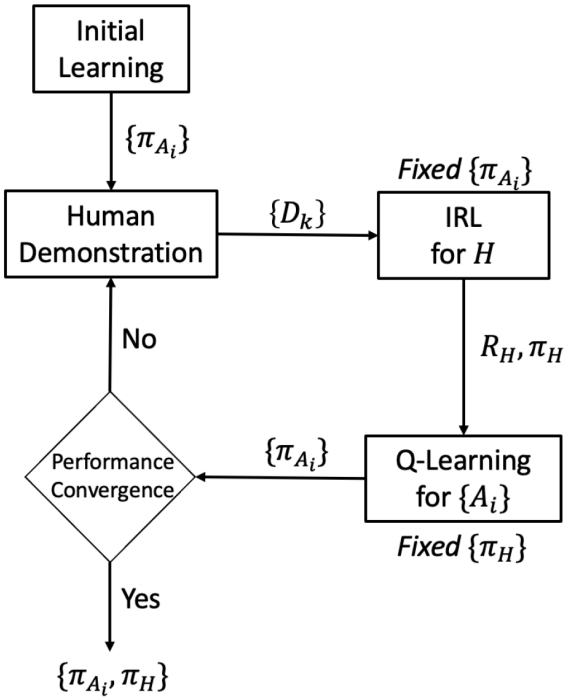

Fig. 2 Iterative Training Framework for Human-Autonomy Teaming

3.3 Decentralized Q-Learning for Autonomy

In the decentralized Q-Learning of autonomous agents, instead of training a joint policy $\{\pi_{A_1,\dots,A_n}\}$ for all agents, each agent maintains and updates its own Q-table $Q_i(s, a_i)$ and policy $\{\pi_{A_i}\}$ separately by equation (4, 5), where α is the learning rate. By this approach, the size of action space and the Q-table is significantly reduced, hence we could a have much lower computational cost.

$$Q_i(s, a_i) = Q_i(s, a_i) + \alpha(r(s, a_i) + \gamma max_a Q(s', a_i) - Q(s, a_i)) \quad (4)$$

$$\pi_{A_i} = argmax_{a_i} Q(s', a_i) \quad (5)$$

In decentralized decision-making processes, each agent does not known the overall state transition probability since it does know have the knowledge of decisions made by other. The model-free Q-Learning algorithm is still valid in this decentralized learning.

However, to train the human-like agent, the actions of autonomous agents have to be kown to provide the overall state transition model $P_{\{\pi_{A_1,\ldots,A_n},\pi_H\}}(s,s')$. The next section would discussed the method to train the human-like agent.

3.4 Bayesian Inverse Reinforcement Learning for Human-Like Agent

The objective of this part is to learn the reward function $\{R_H\}$ for the human-like agent that simulates human behavior from demonstration data. The PolicyWalk algorithm we use here is based on Markov chain Monte-Carlo method, which tries to find the probability distribution of the reward function $\{R_H\}$ that best satisfies the provided demonstration D, denoted as $\Pr(R|D)$. By the Bayes' rule, we could express the probability with equation (6), where $\Pr(R)$ is the prior distribution of the reward function, which could be uniform or beta distribution based on the scenario. The exponential term is the likehood, where (s_i, a_i) is the state-action pair in D, θ is a tuning parameter deciding the probability of choosing highest Q-value, and η is the normalizing factor.

$$\Pr(R|D) = \frac{\Pr(D|R)\Pr(R)}{\Pr(D)} = \frac{1}{\eta} e^{\theta \sum_i Q(s_i,a_i,R)} \Pr(R) \tag{6}$$

When using the PolicyWalk algorithm developed by Ramachandran & Amir, (2007), we fix the policies of other agents $\{\pi_{A_i}\}$ and treat them as part of the MDP model and only update the reward function $\{R_H\}$ as well as the policy $\{\pi_H\}$ of the human-like agent if the probability of the candidate reward function is higher than the probability of the current reward function.

Toward Efficient Sampling. Since the operation state is described by n tasks and p task levels, the number of state is p^n. The compuataional cost to update the probability on the reward function defined on the total number of state is significantly large. Instead, we defind the reward function on the sum of tasks levels or operation levels. Then the dimension of the reward function reduce to $n(p-1)+1$ from p^n. In Section 4.2, we will discuss more on the reward function by looking at the posterior distribution of the reward on operation levels.

4 Experiments

4.1 Problem Scenario

Based on the algorithm discussed before, the experiments are designed and carried out. Some details of the problem scenarios in the real experiments need to be clarified first. At present, the task allocation problem of the research interest is four tasks handled by the teamwork of two agents and one human. In this decentralized problem, all agents as well as the human are independent and identical participants. Each task is described by the severity levels ranging from 1 to 3. To add some complexity to the problem, each task also involves two aspects of operation: firefighting and rescuing people. That means each agent or the human participant can choose a task to go to and then try to put out the fire or rescue people. But if the fire level is as high as 3, the probability of successful rescue will go down to be zero. And when the people are rescued or the fire is extinguished in some task, the severity level will decrease to 1, which means the task has been finished. In this research, both aspects of capabilities of the agents or the human are all defined as 1, so this attribute matrix has all identical capability levels. That means any of the participants doesn't have a special skill in firefighting or rescuing people.

This model can be easily extended to larger scale problems without the loss of comprehensive task dynamics since tasks can be dependent on each other. Also, with different choice of capabilities, the agents become specialized and heterogenous, which could help further studies on heterogenous teaming.

4.2 Results and Discussions

Initla Training. As an initialization, reinforcement learning is used to train the initial autonomous agent policy $\{\pi_{Ai}\}$ for the collaboration between agents. As a result, in this step, three identical and independent agents work together on the tasks. And the reward function is defined as $\mathrm{R}(1) = 10$, which means that only if the severity level of operation has been decreased to 1 (all task levels are reduced to 1), a reward of 10 can be given. The results of the initial reinforcement learning are shown in Fig. 3. The Decentralized Q-Learning algorithm succeeds in finding the optimal policy for agent teamwork. As the increase of episodes, the evaluation will also go up. And in the end, it will equilibrate and vibrate around some value due to the stochastic envrionment. The reward value indicates that after reinforcement learning, the agents can collaborate well given the initial agent policy.

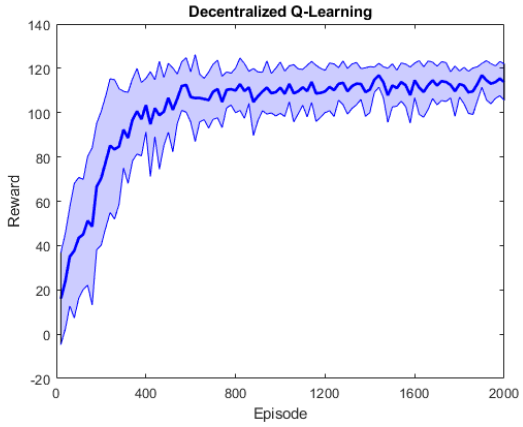

Fig. 3 Training of initial autonomous agent policy

Bayesian Inverse Reinfocement Learning for Human-Like Agent. After working with autonomous agents, enough data of human demonstration $\{D_k\}$ can be collected. And for the second step, the BIRL is applied to help train the human-like agent $\{\pi_H\}$ from demonstrations. As is shown in Fig. 4, the distributions of some representing levels of the reward weight are displayed. In Fig. 4 (a) and (b), level 2 and level 3 of the reward weight, which affect the human-like policy $\{\pi_H\}$ the most, are shown. We can find that they show a similar distribution to the normal distribution. And they have some clear tendencies. For example, the reward of level 2 has the largest probability to be 3 and the reward of level 3 tends to be 3. As a contrast, level 9 of the reward weight, which has little impact on the human-like policy, shows the bimodal distribution. This difference in the reward weight distribution indicates that many different reward samples are not affecting the behavior of the human-like agent. This method therefore provides the insight on which operation levels can be ignored for reward function to further improve the sampling efficiency.

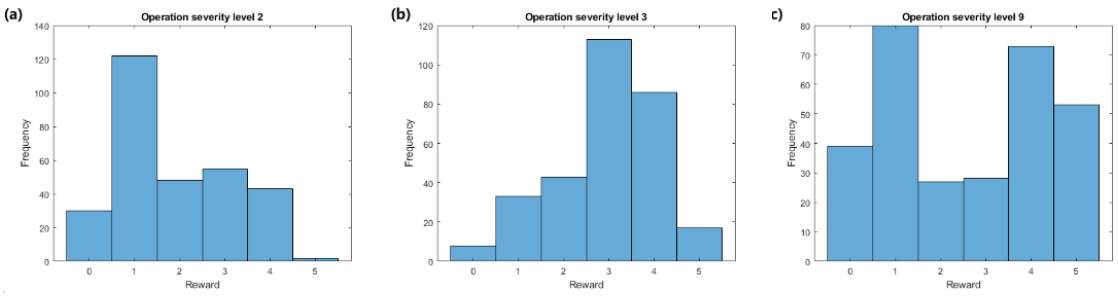

Fig. 4 Distribution of representing levels of the reward weight of the human-like agent. (a) Level 2 of the reward weight. (b) Level 3 of the reward weight. (c) Level 9 of the reward weight.

Effect of Number of Demonstrations per Iteration. After getting the human-like agent policy, Q learning is used in the third step. In this section, the human-like agent is used to help improve the performance of teamwork between the human and agents. Also, to study the impact of the size $\{D_k\}$, experiments using the different numbers of framework iterations and sets of human demonstration data are carried out by keeping the total number of demonstrations the same. Two experienments are carried out here and discussed: First, 4 demonstrations with 1 framework iteration. Second, 2 demonstrations per iteration with 2 iterations. The performance of team collaboration for both cases are shown in Fig. 5 and Fig. 6 respectively.

Fig. 5 shows the results of the experiment using four sets of human demonstration data and the number of framework iterations is one. As is shown in Fig. 5 (a), the straight line at 0 means that the evaluation of initial teamwork has zero performance reward all the time. That indicates that the initial autonomous agent policies do not work with the human at all, and the success rate on accomplishing all tasks is 0%. And as a contrast, Fig. 5 (b) in the right shows a different distribution, which has a mean of about 103.6. As a result, there is a huge improvement in the evaluation of the teamwork after one framework iteration in this experiment. The performance of the teamwork between agents and human is largely improved.

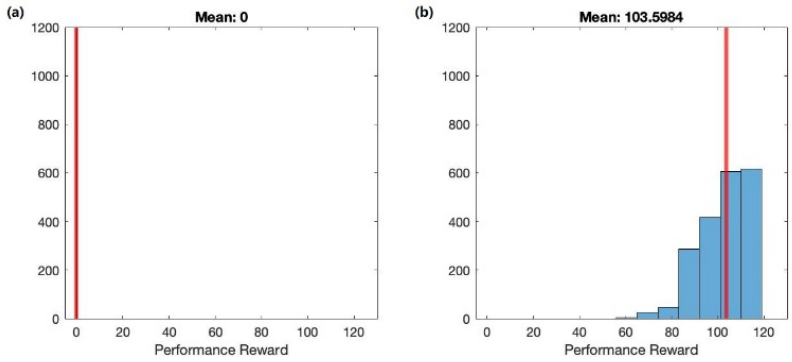

Fig. 5 Evaluation of the teamwork between two agents and one human. 4 sets of human demonstration data and one framework iteration is applied in this experiment. (a) Initial autonomous agent policy. (b) Improved policy after 4 demonstrations

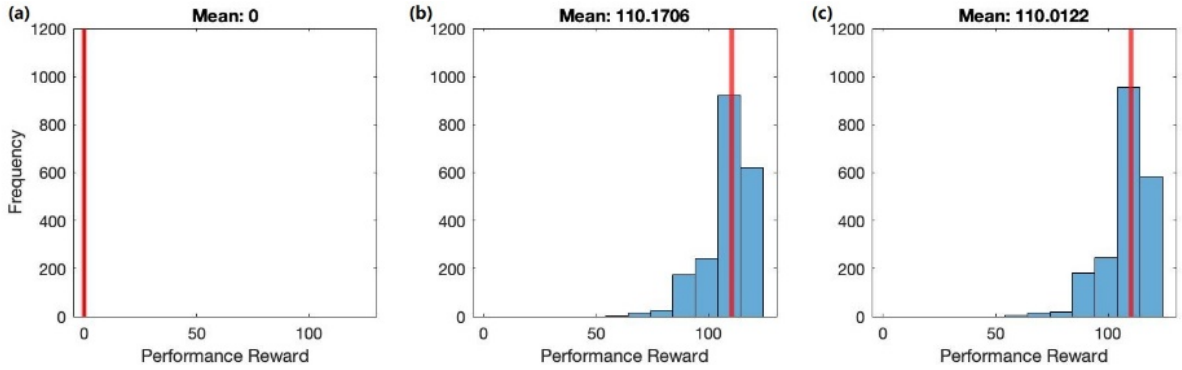

Fig. 6 Evaluation of the teamwork between two agents and one human. 2 sets of human demonstration data and 2 framework iteration are applied in this experiment. (a) Initial autonomous agent policy. (b) Improved policy after one iteration. (c) Improved policy after two iteration

Fig. 6 shows the results from another experiment using two sets of demonstration data, and these 2 datasets are used in the two framework iterations. For this 2×2 structure, the initial distribution in Fig. 6 (a) shows the same all zero result as that in Fig. 5 (a), meaning the failure of the initial collaboration. From Fig. 6 (b) (c), we can find that the mean performance reward increases to about 110.

Considering the data from the first experiment, such a performance reward implies that the collaboration of the policy from 2 sets of demonstration data is better than the one from 4 sets of data and only one framework iteration, which is about 103.6.

Additionally, it is shown that the evaluation after the first iteration is much better than expected. It is even higher than the evaluation after the second framework iteration. However, this is probably because of the subjective nature of the human demonstration. In fact, after the first iteration, the performance would be further improved after collecting data from the demonstration during the second framework iteration. As a result, for the same size of total human demonstration data {D}, we should choose a smaller size of $\{D_k\}$ and run more framework iterations to improve the performance of the teamwork between agents and human.

5 Conclusion and Future Works

In this research, we proposed an iterative framework and applied the human-like agent to help improve the agent policy designed for teamwork with human. It is revealed in this study that the initial autonomous agent policy could hardly work with the human to finish the task. As a contrast, the improved agents from the framework iteration using Q-learning show a collaborative behavior and improved performance. We also further probed the effect of size of $\{D_k\}$. Through experiments, it is observed that, given the same size of total human demonstration data, a smaller size of $\{D_k\}$ and more framework iterations will help to improve the teamwork between agents and human.

In the future, we are going to experiment with the choice of temporary human agents in the initial learning phase, because initial learning with a more exploratory agent could help in this research.

Also, there are some limits and challenges for our project. The most important challenge lies in scalability. Specifically, and firstly, our problem scale is pretty small. If the problem scale increases, the computation cost would be very high. Secondly, in this study, the capabilities of all agents and human are set to be the same. If the participants can have different skills in different aspects of the tasks, the results may be different. Thirdly, as is shown before, it is found that many different reward samples will not affect the behavior of the human-like agent. As a result, these challenges would inspire further studies in our project.

References

Abbeel P, and Ng A. Apprenticeship Learning via Inverse Reinforcement Learning. In *International Conference on Machine Learning*. 2004.

Asfour T, Gyarfas F, Azad P, and Dillmann R. Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots. In *IEEE-RAS ICHR*. pp. 40-47, 2006.

Choi H, Whitten A. K, and How J. P. Decentralized task allocation for heterogeneous teams with cooperation constraints. In *American Control Conference*, pp. 3057-3062, 2010.

Liu Q, Wu H, and Liu A. Modeling and Interpreting Real-world Human Risk Decision Making with Inverse Reinforcement Learning. In *ICML*, 2019.

Omidshafiei S, Pazis J, Amato C, How J.P, and Vian J. Deep Decentralized Multi-task Multi-Agent Reinforcement Learning under Partial Observability. In *ICML.* 70:2681-2690, 2017.

Puterman L. Markov Decision Processes: Discrete Stochastic Dynamic Programming. 1994.

Ramachandran D and Amir E. Bayesian Inverse Reinforcement Learning. In *IJCAI*. 2007.

Watkins C and Dayan P. Q-Learning. In *Machine Learning*. Vol 8, pp. 279-292, 2004.