Preference Inference from Demonstration in Multi-objective Multi-agent Decision Making

Doctoral Consortium

Junlin Lu
School of Computer Science,
University of Galway
Galway, Ireland
J.Lu5@nuigalway.ie

ABSTRACT

It is challenging to quantify numerical preferences for different objectives in a multi-objective decision-making problem. However, the demonstrations of a user are often accessible. We propose an algorithm to infer linear preference weights from either optimal or near-optimal demonstrations. The algorithm is evaluated in three environments with two baseline methods. Empirical results demonstrate significant improvements compared to the baseline algorithms, in terms of both time requirements and accuracy of the inferred preferences. In future work, we plan to evaluate the algorithm's effectiveness in a multi-agent system, where one of the agents is enabled to infer the preferences of an opponent using our preference inference algorithm.

KEYWORDS

Multi-objective Reinforcement Learning; Preference Inference; Dynamic Weight Multi-objective Agent; Multi-agent System; Opponent Modelling

ACM Reference Format:
Junlin Lu. 2023. Preference Inference from Demonstration in Multi-objective Multi-agent Decision Making: Doctoral Consortium. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

In a multi-objective decision-making process, the agent receives reward vectors for different objectives. The utility function is used to make trade-offs between competing objectives by evaluating the reward vector as a utility scalar. In the existing literature, the most frequently used approach to get utility scalar is linear scalarization [1, 2, 5, 7]. In linear scalarization, the weight over the reward is referred to as the *preference*. However, giving a precise numerical preference is not always intuitive for users. For example, consider a case where a portfolio manager selects stocks based on their weighting of minimizing risk and maximizing profits. He/she might want to give a higher weighting to maximize profits but a specific value is hard to determine. A small error in their preference can result in a significantly different policy which may lead to a sub-optimal solution. However, although it is difficult to numerically name the preference, a user can often demonstrate their preferences. Preference Inference (PI) methods that use demonstration are therefore helpful when solving such problems.

Furthermore, in a multi-agent multi-objective decision-making process, if some agent can use the PI mechanism to infer other agents' preferences, it could gain information about the other agents. The knowledge of others' preferences can provide an advantage to these "wise" agents.

2 PREFERENCE INFERENCE

PI is to infer a set of weights that are used to scalarize a reward vector during the multi-objective decision-making process. This is similar to inverse reinforcement learning (IRL), which infers the reward function by finding a set of linear parameters that scalarize state-relevant features. There are two assumptions:

- The trajectories observed are from the optimal policy or near-optimal policy. This is a widely accepted assumption in existing literature for both IRL [9, 11, 12] and PI [10].
- Based on the first assumption, given either optimal or near-optimal policy, the average reward trajectory is solely determined by preferences and environment transitions.

The PI problem happens when we are given a point assumed to be on the Pareto optimal set (POS) based on some unknown weights for the utility function, and we would like to know what exactly the weights are. For more realistic scenarios, we also consider dominated points that are close to the points on the POS, known as *sub-optimal policies* to test the PI algorithm. By adding sub-optimal noise to the data, we ensure that the inference model is robust on sub-optimal reward trajectories.

We first use the dynamic weight reinforcement learning (DWRL) agent to generate behaviour trajectories [4]. We then propose the dynamic weight-based preference inference (DWPI) algorithm to infer the preferences of the agent for different behaviour trajectories. The training process of the DWPI algorithm is presented in Figure 1. The DWRL agent takes the preference vector as part of the state so that it can change its behavior pattern during runtime by changing the preference. Given the interaction between a trained DWRL agent and the environment, a set of optimal reward trajectories are generated. The trajectory set is augmented by adding random sampled sub-optimal noise to be the training set of the inference model. The inference model is trained under the supervised learning paradigm by inputting the reward trajectory and predicting the corresponding preference.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

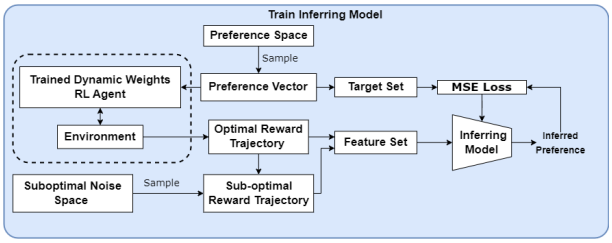

Figure 1: DWPI Training Phase

Algorithm 1 Dynamic Weight Preference Inference Algorithm

Initialize inferring model $\mathcal{I}$, sub-optimal noise space $\mathcal{SN}$, environment $\mathcal{E}$, and preference space Ω
Load the trained dynamic weights RL agent AG
Initialize feature set X, target set Y
while not enough entries in X **do**
  Sample a preference vectors $\boldsymbol{\omega}$ from Ω
  AG plays one episode with $\boldsymbol{\omega}$, generates reward trajectory $\boldsymbol{\tau_r}$
  Sample a noise vector $\boldsymbol{\delta}$ from $\mathcal{SN}$
  Store the noisy reward trajectory $\boldsymbol{\tau_r} + \boldsymbol{\delta}$ in X, store $\boldsymbol{\omega}$ in Y
end while
while $\mathcal{I}$ not converge **do**
  Sample batch from X to train the inferring model, loss $\mathcal{L} = \|\hat{\omega} - \omega\|$
end while

We evaluate our algorithm in three environments: Convex Deep Sea Treasure[8], Traffic[4], and Item Gathering[4] and compared to two benchmarks projection method (PM) [3] and multiplicative weights apprenticeship learning (MWAL) [10]. Our method outperforms the baseline methods by both time efficiency (Figure 2) and PI performance (Table 1). The experiments are implemented with Python 3.9, TensorFlow version 2.3.0, and run on a machine with 11th Gen Intel(R) Core(TM) i7-1165G7 2.80GHz CPU.

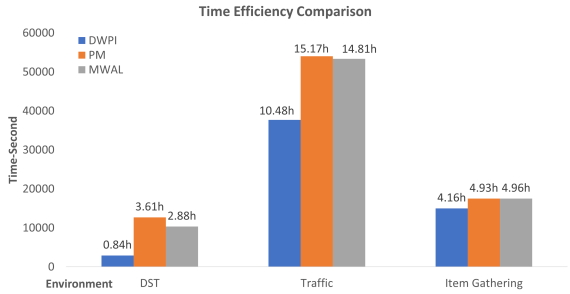

Figure 2: Time Efficiency Comparison

The results of the evaluation show that the DWPI algorithm performs well in terms of inference accuracy.

3 OPPONENT PREFERENCE MODELLING

In future work, we will utilize the DWPI algorithm in the multi-agent environment to enable an agent to gain knowledge of other agents' preferences to gain an advantage over them.

It will be evaluated in two environments. The first one is the Wolf-pack environment [6], where two predator agents try to capture randomly moving prey. If the capture happens when the Manhattan distance between the two predators $dist \leq 3$, it is determined as cooperation or competition when $dist > 3$.

The second environment is the multi-agent item gathering modified from [4]. Two RL agents move in the grid world to gather randomly distributed blocks with three different colors, i.e. green, red, and yellow. Each agent has a preference weight vector over the color of the blocks.

Two sets of experiments will be done in each environment. The first experiment is between two normal agents while the second experiment is between a normal agent and a wiser agent which is able to infer the other's preference. The performance of the agents will be analyzed to check whether the inference mechanism will help the wiser agent gain advantages during the games.

Table 1: Performance Improvement

Environment		Traffic		Item Gathering	
KL-divergence		PM	MWAL	PM	MWAL
Optimal Demo	DWPI	90.8%↑	99.56%↑	98.01%↑	99.13%↑
Sub-optimal Demo	DWPI	89.55%↑	99.53%↑	96.89%↑	99.25%↑
Mean Squared Error		PM	MWAL	PM	MWAL
Optimal Demo	DWPI	97.7%↑	99.8%↑	85.91%↑	98.1%↑
Sub-optimal Demo	DWPI	94.13%↑	99.83%↑	83.81%↑	98.9%↑
Utility		PM	MWAL	PM	MWAL
Optimal Demo	DWPI	60.56%↑	98.93%↑	90.67%↑	82.62%↑
Sub-optimal Demo	DWPI	71.87%↑	90.60%↑	99.85%↑	94.50%↑

4 CONCLUSION

We propose the DWPI algorithm to infer preference from demonstrations in the multi-objective decision-making process. We further evaluate the utility of this algorithm for enhancing an agent in the multi-agent system. With our PI model, an agent can know its opponent better and therefore achieve better performance on its target. For future work, we would like to evaluate the opponent preference modeling performance in multi-agent systems. The extension of the DWPI algorithm to infer a non-linear preference is also a direction that we are interested in.

ACKNOWLEDGMENTS

Junlin Lu's research is supported by the Government of Ireland Postgraduate Scholarship (GOIPG/2022/2140).

REFERENCES

[1] Andrea Castelletti, Francesca Pianosi, and Marcello Restelli. 2013. A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run. *Water Resources Research* 49, 6 (2013), 3476–3486.
[2] Paulo Victor R Ferreira, Randy Paffenroth, Alexander M Wyglinski, Timothy M Hackett, Sven G Bilén, Richard C Reinhart, and Dale J Mortensen. 2017. Multi-objective reinforcement learning-based deep neural networks for cognitive space communications. In *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*. IEEE, 1–8.
[3] Akiko Ikenaga and Sachiyo Arai. 2018. Inverse reinforcement learning approach for elicitation of preferences in multi-objective sequential optimization. In *2018 IEEE International Conference on Agents (ICA)*. IEEE, 117–118.

[4] Johan Källström and Fredrik Heintz. 2019. Tunable dynamics in agent-based simulation using multi-objective reinforcement learning. In *Adaptive and Learning Agents Workshop (ALA-19) at AAMAS, Montreal, Canada, May 13-14, 2019*. 1–7.
[5] Mohamed A Khamis and Walid Gomaa. 2014. Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence* 29 (2014), 134–151.
[6] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent reinforcement learning in sequential social dilemmas. *arXiv preprint arXiv:1702.03037* (2017).
[7] Junlin Lu, Patrick Mannion, and Karl Mason. 2022. A multi-objective multi-agent deep reinforcement learning approach to residential appliance scheduling. *IET Smart Grid* (2022).
[8] Patrick Mannion, Sam Devlin, Karl Mason, Jim Duggan, and Enda Howley. 2017. Policy invariance under reward transformations for multi-objective reinforcement learning. *Neurocomputing* 263 (2017), 60–73.
[9] Andrew Y Ng, Stuart J Russell, et al. 2000. Algorithms for inverse reinforcement learning.. In *Icml*, Vol. 1. 2.
[10] Naoya Takayama and Sachiyo Arai. 2022. Multi-objective deep inverse reinforcement learning for weight estimation of objectives. *Artificial Life and Robotics* (2022), 1–9.
[11] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. 2015. Deep inverse reinforcement learning. *CoRR, abs/1507.04888* (2015).
[12] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning.. In *Aaai*, Vol. 8. Chicago, IL, USA, 1433–1438.