Learning the Problem Representation for Improving Negotiation Strategies

Eddy Fledderus
Supervisor(s): Bram Renting, Pradeep Murukannaiah EEMCS,
Delft University of Technology, The Netherlands
22-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

1 Abstract

The domains of the negotiation can vary significantly. It is possible that a domain is very cooperative, where both agents can receive a high utility; the opposite is also possible, where the domain is very competitive and the agents cannot both get a high utility. In the same manner, the agents can have different strategies leading to a complicated problem with no obvious solution.

This research seeks to represent the differences in negotiation domains to improve a machine learning based agent to help the agent generalize these domains. To achieve this several ways of representing the domain have been explored. First is the shared domain information. With this representation, the agent uses information about the amount of issues, values and possible bids there are. Second is the private domain information, in this representation, the agent uses different calculations to get a view of how favorable the domain is in terms of utility. Last is the derived information, this is the representation where the agent learns about the domain by interaction with the environment or the opposing agent.

From the experiments, a conclusion could be made that a part of these representations had a positive impact on the final utility of the agent. The shared domain information had a considerable improvement over the base agent with the features having a non-negligible impact on the negotiation. The derived information also had a considerable impact on the final outcome.

2 Introduction

A negotiation problem consists of multiple agents trying to reach an agreement. The agents try to reach this agreement by sending bids and will, depending on the strategy, usually by slowly concede certain aspects of these bids. In the negotiation problem this research is dealing with, the agents seek to optimize their own utility, social welfare is a byproduct.

This negotiation is done in a certain domain, describing issues over which the agents try to make an agreement, every issue has multiple possible values, each having a certain utility value associated with it. This utility value is private, meaning that the opponent does not know the associated utility.

This research seeks to answer the following question: can we create a Representation from a Negotiation problem and use it to improve our strategy? To elaborate on this: domains in this problem can be very different from each other. It's very possible that adjusting the strategy depending on what the domain looks like will significantly improve the final results.

The agent that is used for this research is a machine learning agent, using a framework similar to the BOA architecture [1]. In this architecture the tasks of the problem are divided into 3 subproblems: Bidding strategies, Opponent Modeling and Acceptance strategies. In the used agent, the bidding and acceptance strategy are combined. The bidding and acceptance strategy is the way the agent decides what to bid and when to accept. The opponent modeling is used to figure out which parts of the solution space are important to the opponent.

This paper seeks to answer the main question by first looking at what information can be found. Then this information will be subjected to an analysis of how useful this information is in the negotiation. And finally there will be an analysis what features in the representation will improve the agent in practice.

3 Related work

Similar research has been done where features of a domain were used to select a pre-existing algorithmic method [2]. In this similar research these features were used in a selection of machine learning approaches to select a method. In this research, a similar tactic will be used. However, instead of choosing an agent, a singular NN policy will directly use this domain knowledge. Another similar research is [3]. In research the reinforcement based approach has been used in an agent, in very similar manner to the agent in this research. The agent of [3] uses a BOA based approach where a reinforcement learning policy was used in the bidding strategy.

4 Background

A bilateral negotiation problem consists of 2 agents negotiating over a domain consisting of issues, each having different possible values. These values can have a different utility for each of the agents. Each agent knows their own utility per value of an issue. Both agents are tasked with trying to get a deal while trying to optimize their own utility. During the negotiation the agents will get turns, in which

an agent can either accept a bid or send a counter-bid. The game ends after a time deadline or when either of the agents accepts a bid. For this research, a reinforcement learning agent will be used based on Proximal Policy Optimization. In this approach the agent will be given a delayed reward, for every 10 negotiation sessions a reward will be summed where after the agent will be trained with this reward. The architecture can be summed up with the picture shown in Figure 1. Where the action is a bid and an observation is a bid from the opponent. In the turn of the agent a representation of the domain gets is fed to the NN policy to calculate a target utility for itself and the opponent. In the base agent this representation is the utility of the last 3 bids from the opponent and the progress of the negotiation. Using these utilities the opponent model then gets used to find a bid close to that goal. When this bid is found, the utility of this bid is compared to the last bid of the opponent, if the opponent's bid has a higher utility, the agent accepts the bid. Else, the found bid will get send to the opponent.

5 Methodology

In this paper, the features shown in Table 1 will be tested as a representation for the problem. The Features are grouped in likeness: the common features are known by all agents, the private features are known by only the agent itself and derived features are found during a negotiation session. In order to measure the effectiveness of these different features the features have been compared to the base agent. In particular the average utility of the base agent tested against a set of opposing agents, has been compared against a set of features in addition to the base agent in the exact same setting. The base agent as mentioned before uses the utility of the last 3 opponent bids and the progress of the negotiation. First, the features grouped in similarity will be compared to the base agent. The grouped information will be almost identical to the features mentioned in Table 1 with the exception of the derived information, which will be split in exact (the first 3) and inexact (the last 2). After this there is an analysis to what extend features add to the results.

6 Results

To ensure consistency all agents are trained for 4 hours and the best achieved utility between the half hour training intervals is mentioned. (The results of training sessions can differ quite a lot, there has not been enough time to retrain the agents to see how much this can influence the results, the results might not be very reliable because of this. Retraining does however offer similar results.) In order to see the improvement of the features we first need to see the performance without them in Figure 2 the performance of the base agent can be seen the agent reached a highest average utility of 0.58 against the test set on the 3.5 hour mark.

The first group of features, displayed in figure 3, consists of: the amount of issues, the amount of all values in issues and the amount of possible proposals. This agent does better than the base agent and beats the the set of opponents while also improving over the base agent's best score with a highest average utility of 0.67 at the 3.5 hour mark. Some interesting remarks are the slow start and the sudden jump in utility. A possible conjecture on why this happens could be the that the features are static making the learning process on these features longer.

In figure 4, the performance of the private info agent is displayed. The private info agent uses the following features in addition to the base features: the average utility of all bids, the standard deviation of all values in issues and the standard deviation of the utility of all proposals. This agent does better than the base agent at every time interval with a highest utility score of 0.60 at the 3 hour mark. Just like the issue values agent these features are static, possibly disproving the previous conjecture. However, the utility of both this agent and the opponents does not seem to change much over time, possibly indicating that the NN policy found a local maximum very quickly, thus not needing to change much.

In figure 5, the performance of the Exact Derived agent is shown. The exact derived agent uses the following additional features: the average utility of the received bids, the standard deviation of the utility of the received bids and the utility of the best bid. This agent does one of the best all agents, having a maximum utility of 0.66 at the 1 hour mark. Making this one of the quickest agents to converge to a strong NN policy. This agent uses a strategy that keeps the utility of the opponent very low, which is not favorable to social welfare.

In figure 6, the performance of the inexact de-

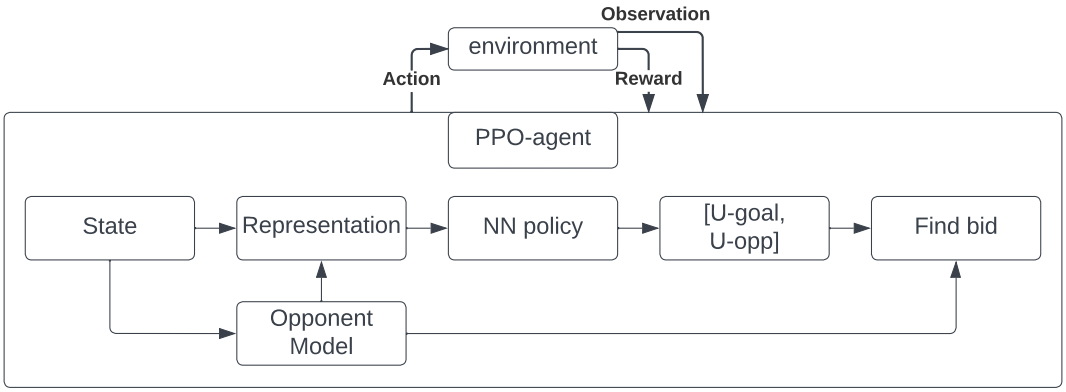

Figure 1: An overview of the ML agent's process of processing bids and learning

Features		
Common	Private	Derived
Amount of issues	Average utility of all proposals	Standard deviation of the opponent's bids
Amount of all values in issues	Standard deviation of all values in issues	Average utility of the opponent's bids
Amount of possible proposals	Standard deviation of the utility of all proposals	Utility of the best opponent bid
		The perceived Nash point
		The perceived improvement of opponent bids

Table 1: List of features grouped in columns by similarity. These features are tested as Representation. These features are mentioned in or are a variation of features in [2].

rived agent is shown. The inexact derived agent uses the following extra features: the perceived Nash point and a slope of the utility of the received bids. This agent does pretty well when compared to the base agent with a maximum utility of 0.63 at the 2.5 hour mark. This is the only agent that deals with inexact results as the perceived Nash point is also a product of the opponent modeling which is only an estimation of the opponent's preferences. In addition to that the slope only gives an idea on how much the opponent works with the agent.

In figure 7, the performance of the previously mentioned agents can be compared. The received utility of all agents is higher than the base agent at a majority of points in the time increments. Indicating that the used features does help the agents to understand the problem.

In figure 8, the performance of the combined agent of the successful agents is shown. This agent uses the combined extra features of all previously mentioned agents. This agent has a maximum utility of 0.66 at the 2.5 hour mark. The received utility is also the most stable among all the agents.

In figure 9, the final comparison of the combined agent and the base agent can be seen.

7 Analysis

In order to see how much the extra features helped improve the strategy the agents shown in Figure 3-6 have been retested. In these tests the extra features have been set to 0 in order to see how much the features added to the negotiation. In Figure 10, the

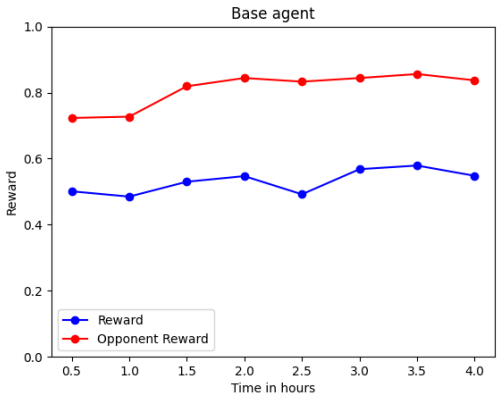

Figure 2: The average reward of the base agent (blue) against that of the test set (red) over time increments of half an hour.

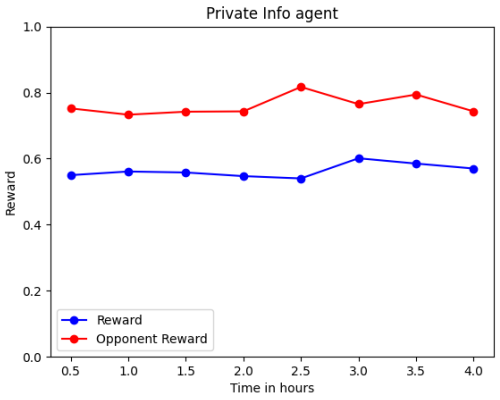

Figure 4: The average reward of the private info agent (blue) against that of the test set (red) over time increments of half an hour.

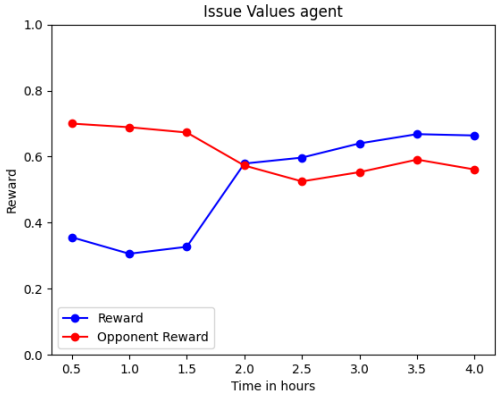

Figure 3: The average reward of the issue values agent (blue) against that of the test set (red) over time increments of half an hour.

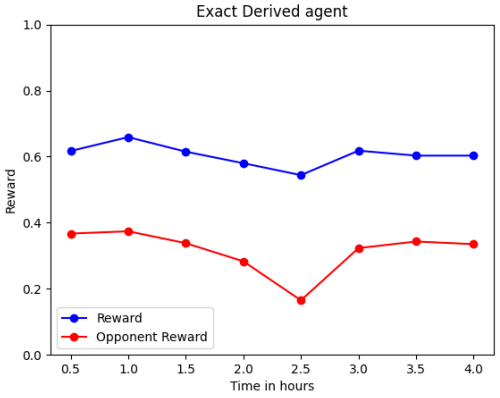

Figure 5: The average reward of the exact derived agent (blue) against that of the test set (red) over time increments of half an hour.

test of the issue values agent is shown. Setting the extra features to 0 improves the agent for the first 3 time increments. After these increments the features do give an improvement, giving the conclusion that these features do help with results.

In Figure 11, the test of the private info agent is shown. This figure brings a less optimistic conclusion with the test agent outperforming its original agent by reaching a higher maximum utility of 6.3 at the 0.5 hour mark. Also remarkable is that this test agent has the highest social welfare of all agents discussed in this paper, indicating that considering social welfare during training could be worth looking into, though not the aim of this paper.

Figure 12 shows the test of the exact derived agent. The exact derived agent still outperforms its test agent and the base agent at every time interval having a difference of maximum utility of 0.06. This test also gives a conclusion that the features do help the agent's utility.

Figure 13 shows the test of the inexact derived agent. This agent does not diverge much from its

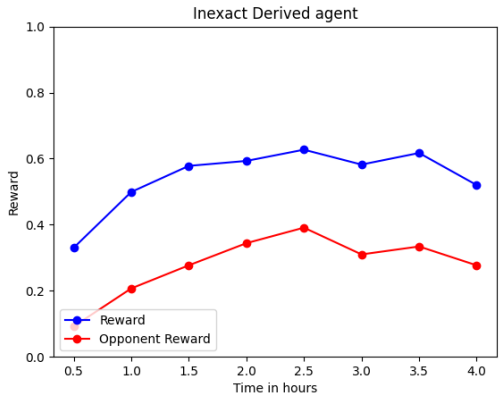

Figure 6: The average reward of the inexact derived agent (blue) against that of the test set (red) over time increments of half an hour.

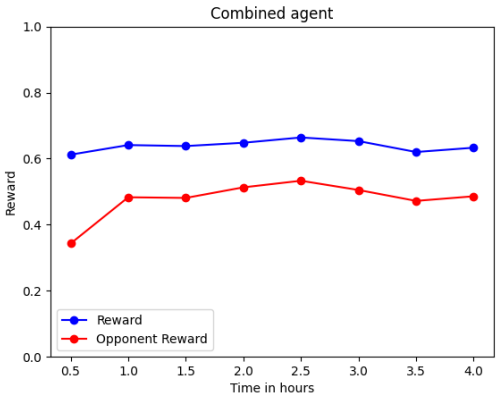

Figure 8: The average reward of the combined agent (blue) against that of the test set (red) over time increments of half an hour.

Figure 7: The average reward of the previously mentioned agents over time increments of half an hour.

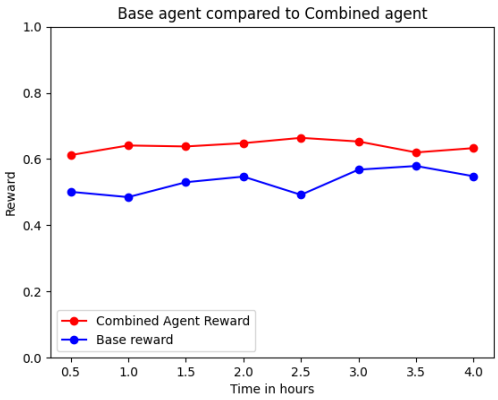

Figure 9: The average reward of the combined agent (red) against that of the base agent (blue) over time increments of half an hour.

test agent only having a difference of maximum utility of 0.02.

8 Responsible research

Multiple precautions have been made to ensure consistent and reproducible results. For one, all variants of the agents have been trained on a set of opponents and have been tested on a different set. This is to ensure that the results are directly comparable to each other. In addition to this, to ensure repeatability, the code and seed from which these results were derived can be found in the repository used for this research.

The agent has not been optimized for social welfare, which if the agent is used anywhere outside of the ANAC competition, might lead to a situation of exploitation from the agent. This exploitation could be prevented by including a reward for social welfare.

9 Conclusions and future work

The main question of this paper was: can we create a Representation from a Negotiation problem and use it to improve our strategy? Drawing a conclusion from the analysis section, it could def-

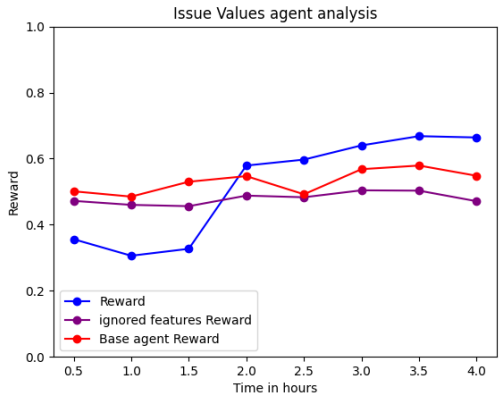

Figure 10: The average reward of the issue values agent (blue), of the same agent with the input of the extra features set to 0 (purple) and of the base agent (red) over time increments of half an hour.

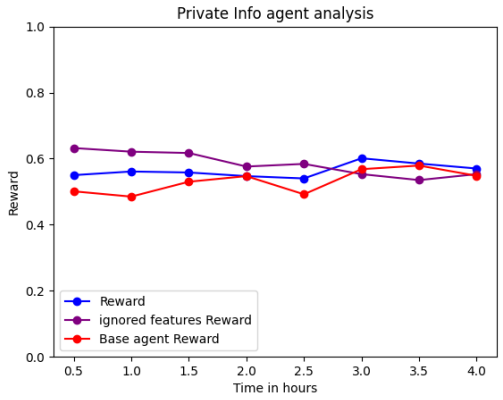

Figure 11: The average reward of the private info agent (blue), of the same agent with the input of the extra features set to 0 (purple) and of the base agent (red) over time increments of half an hour.

initely be said that a general understanding of the size of the domain does improve the strategy of the agent. Adding to that, general knowledge of the opponent's behaviour during the negotiation also improves the agent's utility by quite a bit. The unexpected behaviour from the test private info agent shown in Figure 11, might be worth investigating in the future. This agent has a quite high utility score compared to the other agents and does very

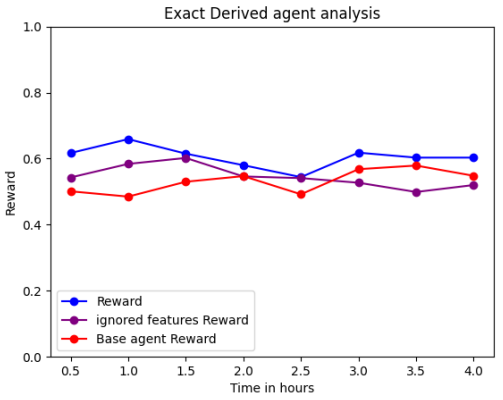

Figure 12: The average reward of the exact derived agent (blue), of the same agent with the input of the extra features set to 0 (purple) and of the base agent (red) over time increments of half an hour.

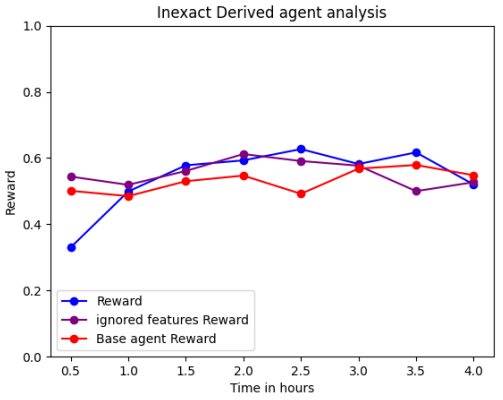

Figure 13: The average reward of the inexact derived agent (blue), of the same agent with the input of the extra features set to 0 (purple) and of the base agent (red) over time increments of half an hour.

well in social welfare. This could possibly indicate that training with social welfare in mind improves personal utility as well.

References

[1] T. Baarslag, What to Bid and When to Stop PhD Thesis, Electrical Engineering, Mathematics and Computer Science, TUDelft, 2014

[2] Ilany, L., Gal, Y. Algorithm selection in bilateral negotiation. Auton Agent Multi-Agent Syst 30, 697–723 (2016).

[3] J. Bakker, A. Hammond, D. Bloembergen, and T. Baarslag, RLBOA: A modular reinforcement learn- ing framework for autonomous negotiating agents, May 2019. [Online; accessed 19 June 2022].