A Bayesian Policy Reuse Approach for Bilateral Negotiation Games

Xiaoyang Gao, Siqi Chen*, Qisong Sun, Yan Zheng, Jianye Hao

College of Intelligence and Computing, Tianjin University
siqichen@tju.edu.cn

Abstract

This work studies how to play with unknown opponents in bilateral negotiation game where two parties of different interests try to reach census following the stacked alternating offer protocol. When being faced with different types of opponents using unknown strategies, it is critically essential for the negotiator to learn about opponents from observations and then find the best response in order to achieve efficient agreements. A novel approach is proposed based on deep Bayesian policy reuse+, which includes two key components, a learning module based on deep reinforcement learning to learn a new response policy when encountering an opponent using a previously unseen strategy and a policy reuse mechanism to efficiently detect the strategy of an opponent and select the optimal response policy from the policy library. The performance of our agent is evaluated against winning agents of ANAC competitions under varied negotiation scenarios. The experimental results show that the proposed agent outperforms existing state-of-the-art agents, and is also able to make efficient detection and optimal response against unknown opponents.

Introduction

As one of the most fundamental and powerful mechanisms for managing inter-agent dependencies, negotiation is central for resolving distributed conflicts between two or multiple parties (Jennings et al. 2001). Automated negotiations are a widely studied, emerging area in the field of autonomous agents and multi-agent systems. Research on agent-based negotiation not only significantly alleviates the efforts of human negotiators, but also aids humans in reaching better outcomes by compensating for the limited abilities of humans, e.g., from the computational, reasoning and cognitive perspective. At present, automated negotiation mechanism has been applied in many fields like e-commerce, laws and supply chain management. Automated negotiations may be rather complex, because there are many factors that characterize negotiations. These factors include the number of issues, dependency between issues, representation of the utility, negotiation protocol, negotiation form (e.g., bilateral or multi-party (Williams et al. 2012; Chen et al. 2013)), and time constraints (Marsa-Maestre et al. 2014; Fujita et al. 2017). Automated negotiations have been studied for a long time and there have been a large body of work concerning various settings (Hindriks and Tykhonov 2008; Chen and Weiss 2014).

In a multi-agent system, the optimal decision of autonomous negotiation agent is contingent on the behaviors of co-existing agents. Especially, when faced with different types of opponents and the opponent's strategy is unknown, the agent is required to be able to detect opponent's strategy accurately and then adapt its own policy accordingly. Though a lot of research works already existed in the field of automated negotiation, none of these works explicitly categorizes the other agent's policy and then dynamically adjust their own coping strategies.

To address the above problem, we design a novel agent – Deep BPR+ negotiating agent – which leverages Bayesian policy reuse (BPR) (Rosman, Hawasly, and Ramamoorthy 2016) for responding to an unknown opponent by selecting among a number of policies available to the agent. BPR maintains a probability distribution (Bayesian belief) over a set of known opponents capturing their similarity to the new opponent that the agent is solving. The Bayesian belief is updated with observed signals which can be any information correlated with the performance of a policy. In this work, agreement utility, number of negotiation rounds and standard deviation of the utility received from opponents' offers are used as the signal. When an unknown opponent strategy comes, identified through moving average reward as in BPR+ (Hernandez-Leal and Kaisers 2017), it switches to learning stage and starts to learn an optimal response policy using deep reinforcement learning algorithm, which learns to achieve efficient agreements by choosing a proper target utility at each step, conditioning on the timeline and offer exchange history.

The main contributions of the paper are as follows:

- We propose a general negotiation agent which supports detection of an unlabeled opponent from observed signals and then adapts its own policy accordingly. Besides, our framework can automatically switch to learn new response policy when faced with a previously unseen opponent.
- We provide a RL-based formulation for automated negotiation, and the learnt policy can adapt to different negotiation domains without retraining.

* Corresponding author.

- We validate effectiveness of the proposed agent by evaluating it against ANAC winning agents under various negotiation scenarios.

Related Work

Negotiation has been widely studied in the recent years in the field of multi-agent system (MAS). Game theory (Liang and Yuan 2008), bayesian learning (Hindriks and Tykhonov 2008) and evolutionary programming (De Jonge and Sierra 2016) have all been used in automated negotiation. Baarslag et al. (Baarslag et al. 2014) proposes an architecture named BOA architecture which separates negotiation strategy to three components, namely, bidding strategy, opponent model and acceptance strategy. A comprehensive survey on opponent models is presented by Baarslag et al. (Baarslag et al. 2016), which classified opponent models using a comprehensive taxonomy. (Baarslag, Hindriks, and Jonker 2014) proposes an simple but efficient acceptance conditions which considers both time and utility gap to determine whether to accept an offer.

In recent years, the successful application of reinforcement learning algorithms in other fields has driven its application in the field of auto-negotiation. Bakker et al. (Bakker et al. 2019) proposes an RLBOA framework based on the BOA architecture for auto-negotiation. The Tabular Q-learning algorithm is used to train the bidding strategy. So they map the offers to the utility space and discretize the utility space. But discretization can lead to information loss. Pallavi Bagga et al. (Bagga et al. 2020) first pre-trains the model through supervised learning to accelerate the learning process, and then trains the DDPG (Lillicrap et al. 2016) model. The disadvantage of this work is that it only addresses a single issue, and its RL agent's state and action are specific issue value, so it cannot work in other negotiation scenarios. (Chang 2020) is limited to specific negotiation scenarios. In (Sengupta, Mohammad, and Nakadai 2021), SAC (Haarnoja et al. 2018) algorithm is used to train the bidding strategy, whose input and output are utility values. So learned model can be used in other negotiation domains. But they do not consider the preferences of opponents.

In MAS, it is critically essential for agents to learn to cope with each other by taking the other agent's behaviors into account. But very little work has been done to explicitly categorize the other agent's policy. BPR+ algorithm (Hernandez-Leal and Kaisers 2017; Hernandez-Leal et al. 2016) can predict other agent's policy and learn a new response policy when previously unseen. But BPR+ is a tabular based algorithm that directly stores learned policies as Q-tables, which might be infeasible when handling large scale problems.

Preliminaries

Negotiation Game Settings

In this work, we consider a negotiation game in which two participants negotiate about multiple issues in a given number of rounds. The negotiation protocol used here is the stacked alternating offers protocol, a variant of (Rubinstein 1982). During the negotiation, each agent makes, in turn, an offer in form of a contract proposal until both sides agree on an offer together, or a deadline is reached (Ito et al. 2011).

Let i be an agent, j be a particular issue and k represent the choice of issue j. We define the value of issue j as v_{jk}. w_j^i denotes the weighting preference which agent i assigns to issue j. The weights of agent i over the issues are normalized summing to one (i.e., $\sum_{j=1}^{n}(w_j^i) = 1$). An offer, O, is a vector of values v_{jk} for each of issues. The utility of an offer for agent i is defined by the utility function as:

$$U^i(O) = \sum_{j=1}^{n}(w_j^i \cdot V_j^i(v_{jk})) \tag{1}$$

where V_j^i is the evaluation function of agent i, mapping the value of an issue j to a real number.

A negotiation scenario consists of a negotiation domain and preference profiles of both parties. Both parties have certain preferences prescribed by a preference profile. These preferences can be modeled by means of the utility function mapping a possible outcome ω to a real-valued number u in the range [0, 1], which indicates how satisfied the party is with an offer. The preference profiles and negotiation domain together constitute the utility space $\mathcal{U}$.

Bayes Policy Reuse

BPR(Rosman, Hawasly, and Ramamoorthy 2016) is proposed as an efficient policy reuse framework for an agent to select the best policy from a policy library when facing unknown tasks. Formally, a task $\tau \in \mathcal{T}$ is defined as a MDP and a policy $\pi(s)$ outputs an appropriate action given state s. The return, which is also known as cumulative reward, is generated by interacting with the environment in the task over an episode of k steps, $U = \sum_{i=1}^{k} r_i$, where r_i is the immediate reward received at step $i-1$. The agent is equipped with a policy library Π which contains coping policies against previously seen tasks set $\mathcal{T}$. When facing an unseen task τ^*, the agent is supposed to select the best coping policy π^* from Π within as small number of trials as possible. BPR uses the concept of $\beta(\tau)$ to measure the degree of similarity between current task $\tau^*$and tasks seen before, where β is a probability distribution over previous seen task $\mathcal{T}$. BPR uses performance model $P(U|\tau,\pi)$ to describe the performance of policy where $P(U|\tau,\pi)$ is a probability distribution over the return U using π on task τ. The belief is initialized with a prior distribution(e.g. random distribution) as $\beta^0(\tau)$. Following the Bayes rule, the belief $\beta^{n-1}(\tau)$ is updated based on $P(U|\tau,\pi)$ as below:

$$\beta^n(\tau) = \frac{P\left(U^n|\tau,\pi^n\right)\beta^{n-1}(\tau)}{\sum_{\tau'\in\mathcal{T}} P\left(U^n|\tau',\pi^n\right)\beta^{n-1}\left(\tau'\right)} \tag{2}$$

Based on the belief $\beta(\tau)$, BPR selects the policy most likely to achieve any possible improvement of return $\bar{U} < U^+ < U^{max}$ as the best coping policy π^*:

$$\pi^* = \arg\max_{\pi\in\Pi} \int_{\bar{U}}^{U^{\max}} \sum_{\tau\in\mathcal{T}} \beta(\tau) P\left(U^+ \mid \tau,\pi\right) dU^+ \tag{3}$$

BPR+ extends BPR to handle non-stationary opponent with a learning mechanism, enabling it to continuously expand its policy library as needed. Deep BPR+(Zheng et al. 2018) uses a refined belief model based on episode return and opponent behavior.

Agent Design

In this section we give the details of our proposed Deep BPR+ Negotiating Agent, as shown in Algorithm 1. This agent is capable of identifying the opponent's strategy in real time, and select the best coping policy in the policy library. Besides, when encountering a previously unseen opponent and none of the policies in the policy library can achieve good performance, it will switch to learning module to learn the new coping policy using DRL algorithm. In section 4.1, we will introduce the learning module of our proposed agent, and in section 4.2, we will explain the policy reuse mechanism.

Algorithm 1: Deep BPR+ Negotiating Agent

Require: Episodes K, performance model$P(U|\mathcal{T},\Pi)$, efficiency model $E(D|\mathcal{T},\Pi)$, behavior model $B(W|\mathcal{T},\Pi)$, policy library Π, known opponent policy set $\mathcal{T}$, window h, threshold δ

for $k = 1, 2, .., K$ **do**
  if stage is reuse **then**
    select a policy π^k based on belief model β^{k-1} and received utility U^t, agreement round D^t and standard deviation W^t (Equation 3)
    update belief model using U^t, D^t and W^t (Equation 8)
    calculated the average performance over past h episodes $\bar{U} = \frac{\sum_{i-h}^{i} U}{h}$
    if $\bar{U} < \delta$ **then**
      switch stage to learn
    end if
  else
    Optimize π using SAC
    if the policy is converged **then**
      update $P(U|\mathcal{T},\Pi)$, $E(D|\mathcal{T},\Pi)$, $B(W|\mathcal{T},\Pi)$, Π, $\mathcal{T}$
    end if
    switch stage to reuse
  end if
end for

Deep Reinforcement Learning Based Learning Module

After detecting that the opponent is using a new strategy, the agent turns to the learning stage and begins to learn the best-response policy against it. We formulate the negotiation problem as a sequential decision making problem which can be solved with a RL agent. We first describe the environment and the method used in this paper to estimate the opponent's preference information as well as acceptance conditions. We then describe the policy-based RL agent and the training procedure of our RL agent. By interacting with the environment, the agent learns to pick the optimal target utility value.

Environment - States, Actions, Transitions and Reward The classic framework of RL consists of two parts. The first part is the external environment ε which specifies the dynamics of the interaction between the agent and the opponent. It is modeled as a Markov decision process (MDP) which can be represented by a 4-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$. The second part is a policy network which maps the state vector to a stochastic policy. The neural network parameters θ are updated using stochastic gradient descent. For the sake of generalization, we design the output of the RL agent as the target utility value, which makes the action space continuous and large. Therefore, compared to value-based RL methods like Deep Q Network(DQN)(Mnih et al. 2013), policy-based RL methods turn out to be more appropriate for our negotiation problem. Before we describe the structure of our policy network, we first elaborate each component(states, actions, rewards) of the RL environment.

States In our negotiation setting, if an agreement cannot be achieved before the deadline, then the negotiation fails. So our agent's decision whether to compromise and the extent of the compromise depends in part on the timeline. Besides, the context during the negotiation process, that is, the historical bid trajectory, is crucial to the agent's decision-making. The state vector at step t is given as follows:

$$s_t = \left(\frac{t}{T_{max}}, u_o^{t-3}, u_s^{t-3}, u_o^{t-2}, u_s^{t-2}, u_o^{t-1}, u_s^{t-1} \right)$$

where T_{max} denotes the maximum rounds of each negotiation session. u_o^t denotes the utility of the bid received from the opponent at step t and u_s^t denotes the utility of the bid proposed by our own agent at step t.

Actions The set of possible actions from a state consists of all possible target utility values in the range $[u_r, 1]$, where u_r denotes the reservation value. Formally, we define the action at step t as $a_t = u_s^t$. To get the actual offer from the utility value, we define an inverse utility function $\mathcal{F} : \mathcal{U} \rightarrow \Omega$ that maps a real-valued number u to an outcome ω, where Ω denotes the outcome space . Specifically, we obtain several offers whose utility value falls within $[u, u + \Delta_u]$, then select the one that the opponent may prefer the most according to the opponent model. In this work, we use the approach proposed by Niels van Galen Last(van Galen Last 2012) for estimating the opponents interests profile, whose main idea is issues that are important to the opponent shall not be adjusted as often. Formally, the inverse utility function $\mathcal{F} : \mathcal{U} \rightarrow \Omega$ is defined as

$$\begin{aligned} \mathcal{F}(u_s) = \arg\max_{\omega} U_o^{'}(\omega), \text{ where} \\ u_s \leq U_s(\omega) \leq u_s + \Delta_u \end{aligned} \tag{4}$$

where $U_o^{'}$ denotes the opponent's utility function estimated by the opponent's historical bids, and U_s represents our utility function. Δ_u denotes the window value. In practice, we set $\Delta_u = 0.05$.

Rewards We only have a terminal reward. The agent is given a positive reward when two parties reach an agreement or reward of -1 when no agreement is achieved before the deadline. Our RL agent's acceptance condition is simple but effective. If our agent plans to propose a deal that is worse than the opponent's offer, we have reached a consensus with our opponent and we accept the offer. Formally, the reward function is defined as follows:

$$r_{t+1}(s_t, a_t) = \begin{cases} U_s(\omega), \text{if an agreement reached.} \\ -1, \text{if no agreement reached in the end.} \\ 0, \text{otherwise.} \end{cases} \tag{5}$$

Policy Network Any policy-based DRL algorithm can be used to solve the MDP modeled above. In this work, we consider Soft Actor Critic (SAC) algorithm(Haarnoja et al. 2018) for learning the optimal target utility value. SAC is a maximum entropy DRL algorithm that optimizes a stochastic policy in an off-policy way. The objective of SAC is to maximize the expected return and the entropy at the same time:

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_{\pi_\theta}}[r(s_t, a_t) + \alpha H(\pi_\theta(.|s_t))] \tag{6}$$

where $H(.)$ is the entropy measure and α controls how important the entropy term is, known as temperature parameter. The policy is trained to maximize a trade-off between expected return and entropy, a measure of randomness in the policy. This helps in improving robustness and generalization of the trained model. Soft Q-value that includes the entropy bonuses is defined as shown below:

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{(s_{t+1},a_{t+1})\sim\rho_\pi} [Q(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1} \mid s_{t+1})] \tag{7}$$

To reduce the overestimation of the value function, SAC uses two value networks. Both value networks are learned with MSBE minimization, by regressing to a single shared target. Since SAC is brittle with respect to the temperature parameter, in implementation, we use SAC with automatically adjusted temperature.

Policy Reuse Mechanism

It is difficult to find a policy that can deal with all opponents. In order to simplify the problem, we consider to reuse our existing policies in the policy library using a belief model that can match current opponent with previously seen opponents, this corresponds to the lines 2 - 8 in Algorithm 1. Every policy in our policy library is able to deal with a certain type of opponent. When encountering an unseen opponent, our policy reuse mechanism will distinguish the possibility that the previously unseen opponent belongs to a certain known negotiation style. The policy reuse mechanism is a very critical part, since higher detection accuracy can lead to more efficient strategy reuse. However, we cannot simply use the vanilla BPR+, which uses a performance model as the signal to detect different task. Since in the field of negotiation, opponents with different negotiation styles may lead to the same agreement utility. Here, we use three signals to evaluate an opponents negotiation style: agreement utility, number of negotiation rounds, the changing trend of the utility received from opponent's offer. The changing trend of the utility received from opponent's offer can be measured with different criterion, here we use the standard deviation.

Similar to Deep BPR+, we can still using Bayes' Rule to update our belief model. Now the belief $\beta(\tau)$ can be regarded as the posterior probabilities measuring the opponents policy, based on the agreement utility, number of negotiation rounds and the changing trend of the utility received from opponent's offer. Like the rectified belief model defined in Deep BPR+, we use performance model $P(U|\tau, \pi)$, efficiency model $E(D|\tau, \pi)$, behavior model $B(W|\tau, \pi)$ to describe the agreement performance, the negotiation efficiency and changes in opponent behavior where $P(U|\tau, \pi)$, $E(D|\tau, \pi)$, $B(W|\tau, \pi)$ are three probability distributions over the agreement utility U, number of negotiation rounds D and the standard deviation of the utility received from opponent's offer W using π on task τ respectively. The belief is initialized with a prior distribution(e.g. random distribution) as $\beta^0(\tau)$ and is updated as below:

$$\beta^n(\tau) = \frac{P(U^n|\tau, \pi^n) E(D^n|\tau, \pi^n) B(W^n|\tau, \pi^n) \beta^{n-1}(\tau)}{\sum_{\tau' \in \mathcal{T}} P(U^n|\tau', \pi^n) E(D^n|\tau', \pi^n) B(W^n|\tau', \pi^n) \beta^{n-1}} \tag{8}$$

Based on the belief $\beta(\tau)$, we selects the policy most likely to achieve any possible improvement of return $\bar{U} < U^+ < U^{max}$ as the best coping policy π^*, as is showed in Equation 3.

For negotiation styles that have never been seen before, this refers to a brand-new style that does not match the policies in the policy library. It usually causes the agent to be at a lower agreement utility regardless of the strategy chosen for a period of time. Specifically, our agent calculates the average agreement utility $\bar{U}$ over h episodes $\bar{U} = \frac{\sum_{i-h}^{i} U}{h}$ as the signal indicating the average performance over all policies till the current episode i. If the average agreement utility $\bar{U}$ is lower than a given threshold $\delta(\bar{U} < \delta)$, the agent will switch to learning module.

Experiments

In this section, we present experimental results of our agent based on the proposed Deep BPR+ negotiating agent. The goal of our experiments is to verify that our agent can efficiently detect the strategy of opponents and also supports the detection of previously unseen policies and learning a response policy accordingly. We first evaluate the performance of our agent against 8 ANAC winning agents. Secondly, we evaluate the performance of our agent against opponents using previously unseen strategies.

Experimental setup

We evaluate the performance of our Deep BPR+ negotiating agent against the following 8 ANAC winning agents: Ponpoko, Caduceus, ParsCat, Atlas3, ParsAgent, The Fawkes,

Table 1: Statistics of 20 domains used in experiments.

Domain	Opposition	Outcome Space	Domain	Opposition	Outcome Space
Acquisition	0.117	384	Icecream	0.148	720
Amsterdam-B	0.223	3024	Kitchen	0.057	15625
Animal	0.110	1152	Laptop	0.160	27
Barter-C	0.492	80	NiceOrDie	0.840	3
Camera	0.212	3600	Outfit	0.198	128
Coffee	0.447	112	planes	0.164	27
DefensiveCharms	0.322	36	RentalHouse-B	0.327	60
DogChoosing	0.051	270	SmartPhone	0.224	12000
FiftyFifty2013	0.707	11	Ultimatum	0.545	9
HouseKeeping	0.272	384	Wholesaler	0.308	56700

CUHKAgent and HardHeaded (Baarslag et al. 2015; Aydoğan et al. 2018; Aydogan 2016)[1]. We conduct experiments on 20 domains from ANAC. The opposition of these domains ranges from 0.051 to 0.840 and the carnality of outcome space ranges from 3 to 56700. Table 1 shows the statistics of these 20 domains we conduct our experiments on.

In the training phase, the domain is randomly selected from these 20 domains for each negotiation session and the policy that the agent learns is evaluated in all 20 domains. The maximum rounds allowed per session is 60. In all experiments the agents are trained until convergence. Moreover, for simplicity all hyperparameters of SAC algorithm are kept fixed while training against different opponents. Although we conduct our experiments on discrete domains, it is worth noting that our proposed agent works in continuous domains as well. When faced with different types of opponents and the opponent's strategy is unknown, an intuitive idea is to train a general agent, which we use as the baseline agent. Specifically, the baseline agent is trained using SAC algorithm with same hyperparameters. Both the scenarios and the opponents that the baseline agent encounters during training are randomly selected for each negotiation session. In our implementation, the baseline agent is trained for a total of 80,000 negotiation sessions.

All the experiments are conducted in our newly-developed negotiation environment. Among the negotiation settings, the reservation price is set as 0.1 and discount factor is ignored for all negotiations. Moreover, we used min-max normalisation for normalising the issue values to between 0 and 1. For performance comparisons, average utility values are calculated on negotiation data obtained in 1000 negotiation sessions between a pair of agents for each negotiation domain.

Performance against ANAC Winning Agents

In this section, we present the empirical results of our agent against 8 ANAC winning agents. We pretrain our agent against each opponent for 10,000 negotiation sessions in succession. So our agent is equipped with the corresponding pre-trained response policies and aims at selecting the most appropriate policy in hand to reuse against the opponent by detecting its behaviors. Our experiments use the following metrics:

(1) Average utility benchmark: the mean utility acquired by the agent a when negotiating with every other agent $b \in A$ in all negotiation domains D where A and D denote the set of all agents and all domains respectively.

(2) Utility against opponent benchmark: the mean utility acquired by agents $b \in A/a$ while negotiating with agent a in all negotiation scenarios.

(3) Domain utility benchmark: the mean utility obtained by all agents $a \in A$ in domain $d \in D$, while negotiating with every agent $b \in A$.

Table 2: Performance comparison based on average utility benchmark, average rounds and average agreement rate.

Agent	avg utility	avg round	agreement rate
Caduceus	0.3461±0.0013	48.98±0.12	0.37±0.00
ParsAgent	0.4570±0.0017	51.73±0.01	0.53±0.00
PonPokoAgent	0.4880±0.0011	48.65±0.23	0.55±0.00
ParsCat	0.5283±0.0003	49.97±0.05	0.64±0.00
Atlas3	0.5572±0.0026	38.16±0.11	0.84±0.00
HardHeadedAgent	0.3900±0.0020	51.75±0.15	0.47±0.00
TheFawkes	0.4369±0.0021	49.81±0.01	0.53±0.00
CUHKAgent	0.4329±0.0007	49.80±0.02	0.51±0.00
Deep BPR+ Agent	**0.6106±0.0039**	**36.92±0.06**	**0.90±0.00**

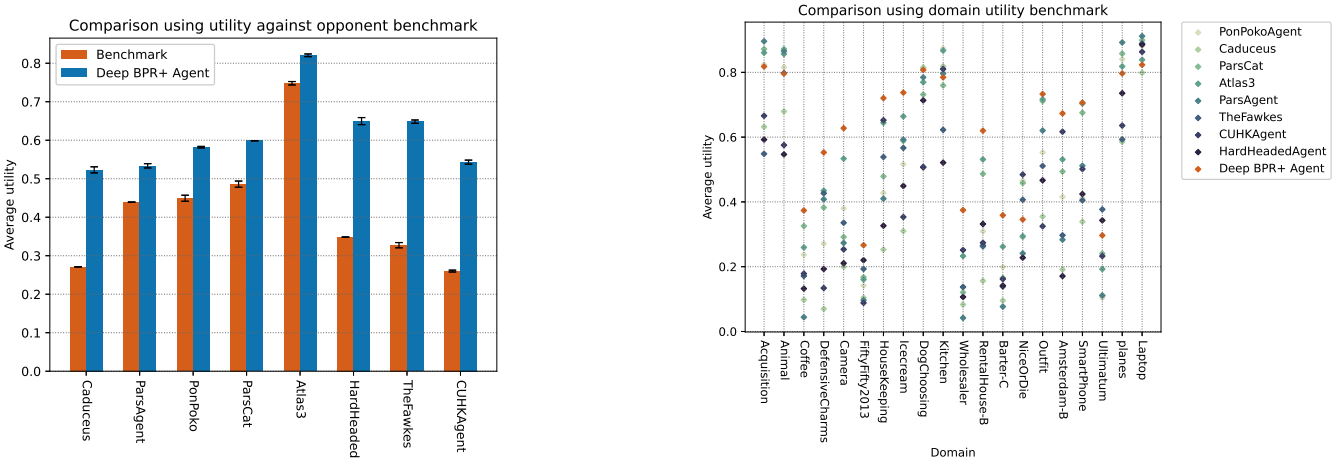
Table 2 shows the performance of our agent on the average utility benchmark, together with average rounds per negotiation session and average agreement achievement rate with standard deviation. Our Deep BPR+ negotiator outperforms all the ANAC winning agents, obtaining higher mean utility, higher agreement achievement rate and converging to an agreement in less rounds, which validates the effectiveness and efficiency of our proposed agent. On the contrary, the baseline agent fails to handle different types of opponents even though it trained with them[2]. In comparison with utility against opponent benchmark, the average utility obtained by our agent is 50% higher than the average benchmark over all ANAC winning agents as shown in Figure 1 (a). This means that when encountering each opponent, the agent is able to accurately detect the strategy of the opponent and act with the optimal policy in order to reach agreements. Figure 1 (b) compares the average utility obtained by our agent with that of 8 ANAC winning agents in each domain. It can be seen that our agent performs best in 12 out of 20 domains. Although our agent doesn't obtain the highest utility in some domains with low opposition like Acquisition, its absolute utility is still high, exceeding 0.8. Therefore in terms of average utility across all domains, our agent significantly outperforms other agents.

New Opponent Detection and Learning

In this section, we evaluate the performance of Deep BPR+ agent against opponents using previously unseen strategy. Now assume that Deep BPR+ agent is now only equipped with 4 response policies against Ponpoko, ParsCat, The Fawkes and HardHeaded. Caduceus, ParsAgent, Atlas3 and

[1] Ponpoko(2017 winner), Caduceus(2016 winner), ParsCat(2016 2^{nd} position), Atlas3(2015 winner), ParsAgent(2015 2^{nd} position), The Fawkes (2013 winner), CUHKAgent (2012 winner) and HardHeaded (2011 winner)

[2] Due to the space limitation, we only present the statistics of baseline agent in this control experiment. Mean utility, average rounds and average agreement achievement rate are 0.4573±0.0040, 49.54±0.07 and 0.57±0.00 respectively.



(a) (b)

Figure 1: (a) Comparison of our Deep BPR+ agent with utility against opponent benchmark consisting of 8 ANAC winning agents. (b) Comparison of Deep BPR+ agent with domain utility benchmark consisting of 20 domains.

CUHKAgent become unseen strategies to it. We first evaluate the performance of this agent against 8 ANAC winning agents by comparing with average utility benchmark and utility against opponent benchmark, the empirical results can be seen in Figure 2. In this evaluation, we set the opponent's strategy to CUHKAgent and keep it unchanged for 4000 negotiation sessions. Our agent may have learned a new response policy against CUHKAgent in these 4000 sessions. Then we evaluate the performance of this agent against 8 ANAC winning agents on all three metrics mentioned above, the experimental results are shown in Figure 3 [3].

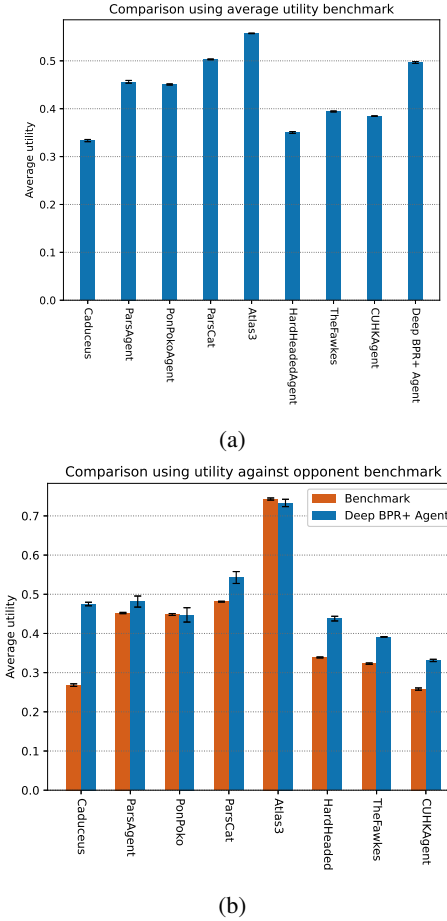
In Figure 2 (a), average utility obtained by our agent is lower than Atlas3 and perform comparably to the second place agent. In Figure 2 (b), although our agent never encounters Caduceus and CUHKAgent before, it achieves higher mean utility than utility against opponent benchmark when negotiating against Caduceus and CUHKAgent. This is because our agent can choose the optimal policy available in the policy library to act, which illustrates the importance of policy reuse mechanism.

After interacting with CUHKAgent opponent for 4000 sessions, our agent performs comparably to the Atlas3 and achieves performance improvements on all three metrics as shown in Figure 3, which means that our agent can detect an previously unseen strategy, and successfully learn a response policy accordingly.

Conclusion

This paper presents a novel Deep BPR+ negotiating agent, which responds to an unknown opponent by detecting the strategy of the opponent from received signals during negotiation and then acting with the best policy in the policy library. Besides, our agent enables online learning of new

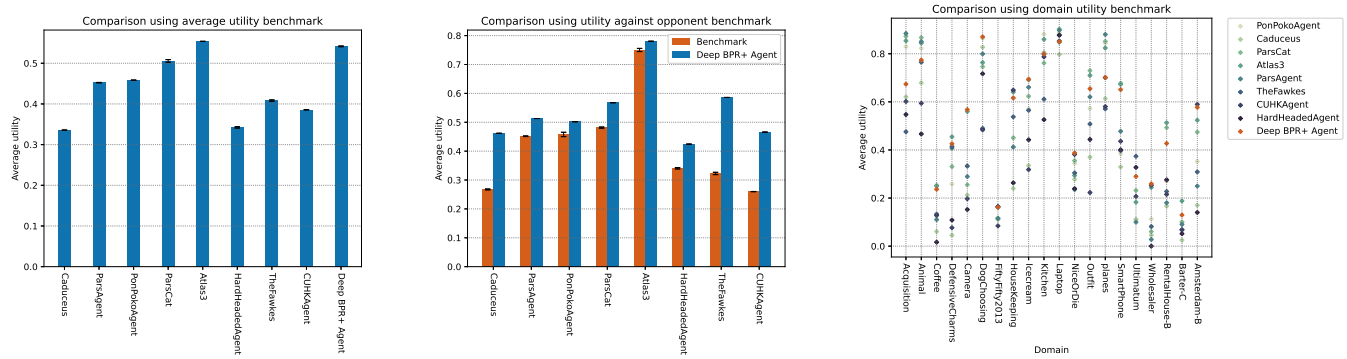
[3] We also conducted other configures and found similar results, so we only report this evaluation.



(a)

(b)

Figure 2: The performance of our agent equipped with 4 response policy against 8 ANAC winning agents by comparing it with average utility benchmark and utility against opponent benchmark.



(a) (b) (c)

Figure 3: The performance of our agent against 8 ANAC winning agents after encountering CUHKAgent opponent and learning the coping policy accordingly.

model when encountering an opponent using a new strategy and our policies available are not performing optimally. Experimental results show an efficient detection of the opponent based on observation signals, obtaining higher average utility than a baseline and ANAC winning agents.

The exceptional results justify to invest further research efforts into this deep BPR+ negotiating agent framework. As for future work, it is worth investigating how to accelerate the online new policy learning phase. Second, the extension of this framework to other negotiation settings, such as concurrent negotiation or multi-lateral negotiation, is another interesting avenue to exploit.

References

Aydogan, R. 2016. ANAC2016 - Automated Negotiating Agents Competition 2016. Website. http://web.tuat.ac.jp/~katfuji/ANAC2016/.

Aydoğan, R.; Fujita, K.; Baarslag, T.; Jonker, C. M.; and Ito, T. 2018. ANAC 2017: Repeated multilateral negotiation league. In *International Workshop on Agent-Based Complex Automated Negotiation*, 101–115. Springer.

Baarslag, T.; Aydogan, R.; Hindriks, K.; Fujita, K.; Ito, T.; and Jonker, C. 2015. The Automated Negotiating Agents Competition, 2010-2015. *AI Magazine*, 36: 115–118.

Baarslag, T.; Hendrikx, M. J.; Hindriks, K. V.; and Jonker, C. M. 2016. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems*, 30(5): 849–898.

Baarslag, T.; Hindriks, K.; and Jonker, C. 2014. Effective acceptance conditions in real-time automated negotiation. *Decision Support Systems*, 60: 68–77.

Baarslag, T.; Hindriks, K. V.; Hendrikx, M.; Dirkzwager, A.; and Jonker, C. M. 2014. Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies. In *Novel Insights in Agent-based Complex Automated Negotiation*, volume 535 of *Studies in Computational Intelligence*, 61–83. Springer.

Bagga, P.; Paoletti, N.; Alrayes, B.; and Stathis, K. 2020. A Deep Reinforcement Learning Approach to Concurrent Bilateral Negotiation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 297–303.

Bakker, J.; Hammond, A.; Bloembergen, D.; and Baarslag, T. 2019. RLBOA: A modular reinforcement learning framework for autonomous negotiating agents. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 260–268.

Chang, H.-C. H. 2020. Multi-Issue Bargaining With Deep Reinforcement Learning. *arXiv preprint arXiv:2002.07788*.

Chen, S.; Ammar, H. B.; Tuyls, K.; and Weiss, G. 2013. Using conditional restricted Boltzmann machine for highly competitive negotiation tasks. In *Proceedings of the 23th Int. Joint Conf. on Artificial Intelligence*, 69–75. AAAI Press.

Chen, S.; and Weiss, G. 2014. An Intelligent Agent for Bilateral Negotiation with Unknown Opponents in Continuous-Time Domains. *ACM Trans. Auton. Adapt. Syst.*, 9(3): 16:1–16:24.

De Jonge, D.; and Sierra, C. 2016. GANGSTER: an automated negotiator applying genetic algorithms. In *Recent advances in agent-based complex automated negotiation*, 225–234. Springer.

Fujita, K.; Bai, Q.; Ito, T.; Zhang, M.; Ren, F.; Aydoğan, R.; and Hadfi, R. 2017. *Modern approaches to agent-based complex automated negotiation*. Springer.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 1861–1870. PMLR.

Hernandez-Leal, P.; and Kaisers, M. 2017. Learning against sequential opponents in repeated stochastic games. In *The 3rd Multi-disciplinary Conference on Reinforcement Learning and Decision Making, Ann Arbor*, volume 25.

Hernandez-Leal, P.; Rosman, B.; Taylor, M. E.; Sucar, L. E.; and Munoz de Cote, E. 2016. A Bayesian approach for learning and tracking switching, non-stationary opponents.

In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 1315–1316.

Hindriks, K.; and Tykhonov, D. 2008. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, 331–338.

Ito, T.; Zhang, M.; Robu, V.; Fatima, S.; and Matsuo, T. 2011. *New trends in agent-based complex automated negotiations*, volume 383. Springer.

Jennings, N. R.; Faratin, P.; Lomuscio, A. R.; Parsons, S.; Sierra, C.; and Wooldridge, M. 2001. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2): 199–215.

Liang, Y.-q.; and Yuan, Y. 2008. Co-evolutionary stability in the alternating-offer negotiation. In *2008 IEEE conference on cybernetics and intelligent systems*, 1176–1180. IEEE.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Marsa-Maestre, I.; Lopez-Carmona, M. A.; Ito, T.; Zhang, M.; Bai, Q.; and Fujita, K. 2014. *Novel insights in agent-based complex automated negotiation*, volume 535. Springer.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Rosman, B.; Hawasly, M.; and Ramamoorthy, S. 2016. Bayesian policy reuse. *Machine Learning*, 104(1): 99–127.

Rubinstein, A. 1982. Perfect Equilibrium in a Bargaining Model. *Econometrica*, 50(1): 97–109.

Sengupta, A.; Mohammad, Y.; and Nakadai, S. 2021. An Autonomous Negotiating Agent Framework with Reinforcement Learning Based Strategies and Adaptive Strategy Switching Mechanism. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, 1163?172.

van Galen Last, N. 2012. Agent smith: Opponent model estimation in bilateral multi-issue negotiation. In *New trends in agent-based complex automated negotiations*, 167–174. Springer.

Williams, C. R.; Robu, V.; Gerding, E. H.; and Jennings, N. R. 2012. Negotiating concurrently with unkown opponents in complex, real-time domains. In *ECAI'12*, 834–839.

Zheng, Y.; Meng, Z.; Hao, J.; Zhang, Z.; Yang, T.; and Fan, C. 2018. A deep bayesian policy reuse approach against non-stationary agents. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 962–972.