AgentNyan

Ryota Arakawa*

1 Reservation Value Modeling

The reservation value modeling algorithm of our agent is based on non-linear least squares. Typical time-dependent algorithm uses a function like Eq.(1) to get the target utility value. Our algorithm predicts a and b by non-linear least squares based on opponent's offers. Then, the agent uses $u(1.0)$ as an expected reservation value. Algorithm 1 is the concrete explanation of our modeling algorithm. We use a list of tuples $O = \{(t_i, u_i)\}$, where t_i is the relative time of ith offer and u_i is the utility of the opponent of ith offer, as an input of the algorithm.

$$u(t) = b + (1.0 - b)(1.0 - t)^a \tag{1}$$

Algorithm 1 Algorithm of reservation value modeling

Require: List of tuples $O = \{(t_i, u_i)\}$
$P_t[0] \leftarrow 0.0, P_u[0] \leftarrow 1.0$
$j = 0$
for all (t_i, u_i) in O **do**
    if $P_u[j] > u_i$ **then**
        $P_t[j+1] \leftarrow t_i, P_u[j+1] \leftarrow u_i$
        $j \leftarrow j + 1$
    end if
end for
if $|P_t| < 2$ **then**
    return $P_u[j]$ (fallback to $|P_u[j]|$ since there are no enough data)
end if
Use non-linear least squares to find a, b such that the function $u(t)$ fits to $P = \{(P_t, P_u)\}$
return $u(1.0)$

* Tokyo University of Agriculture and Technology, arakawa@katfuji.lab.tuat.ac.jp

In Algorithm 1, the agent only uses offers which updated the minimum utility. If a bidding strategy of the opponent has some randomness (like our agent) and the agent uses all offers, the prediction may become unstable. Therefore, the agent filters offers to predict accurately.

Fig. 1 shows an comparison of reservation value modeling algorithm against a noisy bidding strategy. A green curve denotes actual $u(t)$, where $a = 0.5$ and $b = 0.1$. Offered utilities (blue dots) are sampled from a uniform distribution between $u(t)$ and 1.0. An orange curve denotes $u(t)$, where a and b are estimated by the algorithm described above. An blue curve also denotes $u(t)$, but all offers are used to estimate a and b. By filtering offers, our algorithm can estimate the parameters more precisely.

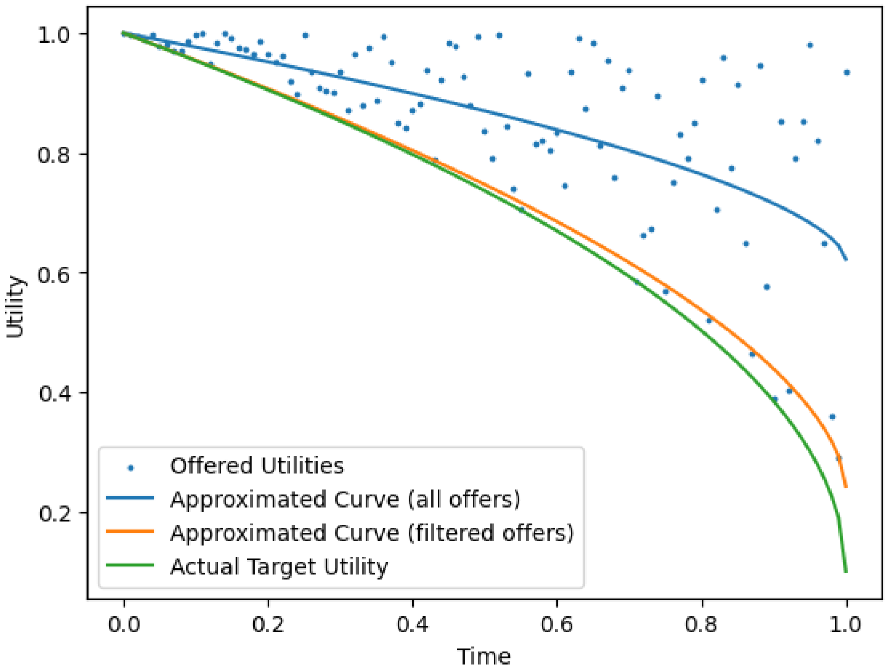

Fig. 1: An comparison of reservation value modeling algorithm

2 Acceptance Strategy

We use the following variables in the following sections.

u_m	an utility of our agent
u_o	an utility of the opponent agent
r_m	a reservation value of our agent
r'_o	an expected reservation value of the opponent agent (calculated by Algorithm 1)
t	an relative time when an offer is created ($0 < t < 1$)

If $t < 0.5$, the agent uses TimeDependent Strategy since there may be no enough data to predict r'_o accurately. Otherwise, the agent uses Advantage Strategy, which uses the expected reservation value r'_o.

2.1 TimeDependent Strategy

First, we calculate u_{th} (utility threshold) using Eq.(1) where $a = 0.5, b = u_m$. If $u_m > u_{th}$, our agent accepts the offer. Otherwise, our agent rejects the offer.

2.2 Advantage Strategy

First, the agent calculates adv_m and adv'_o by Eq.(2) and Eq.(3). These values denote one's (expected) profit compared with its reservation value.

$$adv_m = u_m - r_m \tag{2}$$

$$adv'_o = u_o - r'_o \tag{3}$$

Then, the agent calculates Δ_r (an expected gap of reservation values) and adv_{th} (advantage threshold) by Eq.(4) and Eq.(5). If $adv_m > adv_{th}$, our agent accepts the offer. Otherwise, our agent rejects the offer.

$$\Delta_r = r_m - r'_o \tag{4}$$

$$adv_{th} = \begin{cases} -\Delta_r t & (\Delta_r > 0) \\ -\Delta_r(1-t) & (\text{otherwise}) \end{cases} \tag{5}$$

The lower adv_{th}, the more compromising the agent becomes. From Eq.(5), the agent becomes more compromising over time. Also, the agent takes more aggressive strategy when Δ_r is lower. It is because there are more options that our agent can take more advantage when Δ_r is lower.

3 Bidding Strategy

First, our agent chooses candidate offers from all possible offers. An offer o is a candidate offer if all of the following conditions are met.

- $o > r_m$
- o is pareto optimum
- $t < 0.5$, or $adv_m > adv_{th}$ (calculated by Eq.(4) and Eq.(5))

Then, our agent samples an offer from candidate offers and sends it to the opponent agent. We add some randomness to the bidding strategy. It is intended to make it more difficult for the opponent agent to estimate our agent's reservation value.