PredictRV: A Prediction Based Strategy for Negotiations with Dynamically Changing Reservation Value

Aditya Srinivas Gear(✉), Kritika Prakash(✉), Nonidh Singh(✉), and Praveen Paruchuri(✉)

IIIT - Hyderabad, Hyderabad, India
{aditya.srinivas,kritika.prakash}@research.iiit.ac.in,
nonidh.singh@students.iiit.ac.in, praveen.p@iiit.ac.in

Abstract. Negotiation is an important component of the interaction process among humans. With increasing automation, autonomous agents are expected to take over a lot of this interaction process. Much of automated negotiation literature focuses on agents having a static and known reservation value. In situations involving dynamic environments e.g., an agent negotiating on behalf of a human regarding a meeting, agents can have a reservation value (RV) that is a function of time. This leads to a different set of challenges that may need additional reasoning about the concession behavior. In this paper, we build upon Negotiation algorithms such as ONAC (Optimal Non-Adaptive Concession) and Time-Dependent Techniques such as Boulware which work on settings where the reservation value of the agent is fixed and known. Although these algorithms can encode dynamic RV, their concession behavior and hence the properties they were expected to display would be different from when the RV is static, even though the underlying negotiation algorithm remains the same. We, therefore, propose to use one of Counter, Bayesian Learning with Regression Analysis or LSTM model on top of each algorithm to develop the PredictRV strategy and show that PredictRV indeed performs better on two different metrics tested on two different domains on a variety of parameter settings.

Keywords: Automated negotiation · Dynamic reservation value · Belief update

1 Introduction

Negotiation is an important component of interaction process among humans [18,19,22]. A lot of negotiation literature assumes that we have a good amount of information about our own choices [10,15] and reservation value (**RV**), while not knowing our opponents preferences [4,5,12]. Note that RV refers to the utility of a bid in the negotiation, below which we would not be willing to accept any bid. Reasons for not accepting a bid whose utility is below RV can be due to a

D. C. Morais et al. (Eds.): GDN 2020, LNBIP 388, pp. 135–148, 2020.
https://doi.org/10.1007/978-3-030-48641-9_10

better BATNA - Best Alternative to Negotiated Agreement [6] (so RV may be set to BATNA) or that the agent receives a utility that is not good enough for the agent to accept. In settings where the environment is dynamic, there can be situations where our RV can change with time (while the preference profile is static) [16]. We may not know how the changes would pan out e.g., an agent acting on behalf of a meeting attendee may have varying estimates on when the human may arrive for the meeting [3,23]. Dynamicity of RV can, therefore, throw additional challenges when we are unaware of the nature of changes (which is different from RV changing because of a discount factor where the change is computable). Bids that simply react to the dynamicity may not be sufficient since they can change in a random fashion and result in lower utility. For example, it can be hard to agree on a meeting time if an agent acting on behalf of a human declares that the human would arrive in 30 min and then re-declares in a short period that the human would arrive in 10 min and then quickly change to say 20 min even though the agent may simply be acting based on its belief of when the human would arrive.

1.1 Related Work

Making concessions to reach an agreement is an important part of the negotiation process [8,14,20]. There are a variety of ways in which negotiating agents can concede. One such category of techniques is Time-Dependent Tactics (TDT's) [7,9] e.g., Boulware and Conceder agents. [1] presents an Optimal Non-Adaptive Concession (ONAC) algorithm with incomplete information where time pressure (amount of time to deadline) is a primary criterion to influence the concession behavior. Negotiation algorithms such as ONAC and Boulware [1] work on settings where RV of the agent is fixed and known. Although these algorithms can work with (or be modeled as a function of) a dynamic RV, their concession behaviors can have a lot more randomness or fluctuations compared to when they have a static RV. For purposes of a more stable bidding behavior, the agent should, therefore, make choices based on predicted (RV) values. While the quality of agreement is a default metric used in negotiations, popular negotiation frameworks such as the Genius platform [17] do not support the modeling of dynamic RV. We, therefore, had to develop a simple negotiation simulator that can encode dynamic RV. In addition to the quality of agreement, we use Prediction as an additional metric to evaluate the concession behavior.

We propose to use the following models on top of negotiation algorithms, to handle the effects of a dynamic RV: (a) **Counter model** [24], (b) **Bayesian learning with Regression Analysis** [25,26] and (c) **LSTM model**. All three models are present in literature and we adapt them here to work suitably with the different negotiation algorithms. While the paper builds on top of ONAC and Boulware algorithms, the procedure, in general, would be suitable to apply to algorithms that are sensitive to the dynamicity of RV (which results in fluctuations in bidding). Given that the models help to predict the RV to reduce the effect of dynamicity, we refer to the new strategy as **PredictRV**.

Rest of the paper is organized as follows: Sect. 2 presents an overview of the negotiation model and two negotiation algorithms namely ONAC and Boulware with static RV. Section 3 presents a dynamic RV version of the negotiation model and the ONAC and Boulware algorithms. In addition, it introduces the PredictRV strategy and presents three methods used to make predictions over the dynamic RV namely Counter, Bayesian Learning with Regression Analysis and LSTM based prediction. Section 4 showcases the working of the three prediction methods via an example when faced with dynamic RV. In Sect. 5, we present a variety of experiments on two different domains to evaluate the performance of the PredictRV strategy. Section 6 presents the conclusions of the paper.

2 Static RV

2.1 Negotiation Model

The negotiation model we use follows the alternating offers protocol [21] for a bilateral negotiation: Consider two agents A and B with utility functions $U_A(z)$ and $U_B(z) \in [0,1]$ where z belongs to the set of all possible negotiation outcomes for a domain D. The RV's for the agents are rv_A and $rv_B \in [0,1]$. The agents will propose offers with utility higher than their own RVs.

2.2 Utility Generation for ONAC Algorithm

The ONAC algorithm [1] aims to construct optimal concession strategies against specific classes of acceptance strategies [2]. It applies sequential decision techniques to find analytical solutions that optimize the bidders expected utility, given certain strategy sets of the opponent. The ONAC solution was found to significantly outperform state of the art approaches in terms of obtained utility. As shown in [1], the utility of the ONAC bid is computed by taking into account the probability of acceptance of the bid (x, bid of agent A) by the opponents where the agents have opposing preferences.

$$U_j = U_{j+1} + \underset{U(x) \geq rv_A}{max} (U(x) - U_{j+1})(1 - U(x))$$

where $U(x) \in [0,1]$ and U_{j+1} is the utility of the bid proposed by the agent at round $(j+1)$, U_j is the utility at round j and x is a valid bid with utility greater than RV. This is a recurrence formula that gives the utility of the bid at each round, where rv_A is the RV for agent A and N is the deadline:

$$U_N = rv_A, \quad U_j = (\frac{U_{j+1} + 1}{2})^2,\ j \in \{1, 2, 3, ..., N-1\} \tag{1}$$

2.3 Utility Generation for Boulware Algorithm

The Boulware algorithm is a TDT [7,9], which concedes considerably more as the negotiation deadline approaches. TDTs consist of a family of functions that represent an infinite number of possible. The formula for tactics, one for each

$$U_j = rv_A + (1 - rv_A) * (\frac{min(N - j, N)}{N})^{\frac{1}{\beta}} \tag{2}$$

value of β this family of functions is as follows where j is the jth round and β should be in the range $(0, 1)$ for Boulware.

3 Dynamic RV: The PredictRV Strategy

3.1 Negotiation Model

The negotiation model remains the same as for the static RV case with the following difference: Since agent A's RV is dynamic, it is represented as $rv_A(t)$ (and $rv_B(t)$ for B for generality).

3.2 ONAC for Dynamic Reservation Values

To model dynamic RV we assume that the value of RV is drawn from an unknown probability distribution, and in each round, agent A receives a signal $rv_A(t)$ drawn from that distribution. PredictRV attempts to predict this probability distribution (p.d.) and incorporate it into a negotiation algorithm (ONAC here). We assume that there is no noise in the signal $rv_A(t)$, hence it corresponds to the actual RV at time step t. The PredictRV recurrence formula would be:

$$\begin{aligned} &U_N = rv_A(t), \text{ where t = j at round j} \\ &U_j = (\frac{U_{j+1} + 1}{2})^2, \, j \in \{1, 2, 3, ...N - 1\} \end{aligned} \tag{3}$$

For a dynamic RV, the value to bid will no longer be determined using Eq. (1). Instead, we first need to assign the new RV to U_N and then re-compute for U_j as shown in Eq. (3).

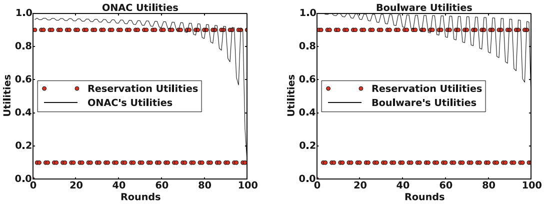
3.3 Boulware for Dynamic Reservation Values

The Boulware algorithm present in negotiation literature assumes a static RV. For Boulware that works with dynamic RV, utilities can be generated using the following function:

$$U_j = rv_A(j) + (1 - rv_A(j)) * (\frac{min(N - j, N)}{N})^{\frac{1}{\beta}}, \text{ at round j} \tag{4}$$

3.4 Illustrative Example

Consider a toy example, where the RV can be either 0.1 or 0.9 and it changes randomly every 2 rounds for a total of 100 rounds. Figure 1 shows the concession curves obtained by using the ONAC and Boulware algorithms. The x-axis of each figure shows the number of rounds from 0 to 100 while the y-axis shows the utility values ranging from 0 to 1. The utilities of the bids at each round are computed using Eq. (3). The figures show that the concession curves are not monotonic due to the dynamic nature of the RV, which results in to and fro concessions being made, where peaks correspond to RV of 0.9 and troughs correspond to 0.1.



(a) ONAC algorithm (b) Boulware algorithm

Fig. 1. Utilities obtained by using ONAC and Boulware algorithms

3.5 Steps of Strategy for PredictRV

Given a negotiation algorithm (like ONAC or Boulware):

1. Generate hypotheses about the RV and assign weights to each hypothesis. Compute the utility for each hypothesis T_{x_i} [by setting U_N as the utility of the hypothesis and plugging in Eq. (3)].
For each round j from 1 to N (no. of rounds):
2. Update weights of hypotheses based on the rv_A at that round i.e., $rv_A(j)$ [using Counter, Bayesian or LSTM approaches presented below]
3. Using the utility computed for each hypothesis in Step (1), we now compute the utility of the bid [Using one of Eqs. (6) or (13)].

End of for

To generate hypotheses (first step), we divide the range between which the RV can vary, into n number of intervals I_i for $i \in \{1, 2, 3, ...n\}$. A suitable point x_i is selected as a representative value for each interval I_i. If the RV falls within an interval, it is classified as having the utility of the point that represents the interval. We then compute negotiation algorithm utilities, $T_{x_i} = \langle U_1(x_i), U_2(x_i), ..., U_N(x_i)\rangle$ using Eq. (3). At the start, all hypotheses are equally likely, hence each hypothesis is initialized with a probability $\frac{1}{n}$ i.e., uniform distribution over hypotheses. As the negotiation progresses we may have a better prediction over the hypotheses based on the past RVs, hence the probability distribution would change. The second step of the PredictRV strategy is to update the weights of the hypotheses as the new round starts. How the weights are updated depends on the actual procedure we use namely Counter, Bayesian Learning or LSTM models presented below.

3.6 Counter Learning

In the Counter based learning procedure, the count for each hypothesis is initialized as $c_{x_i} = 0$, where $i \in \{1, 2, 3, ...n\}$. At a new round j, we obtain a new RV. As step 2 of PredictRV, using the new RV we update the counter for the hypothesis that corresponds to the new RV. We re-compute the probability for each interval as follows:

$$p_{x_i} = \frac{c_{x_i}}{\sum_{i=1}^{n} c_{x_i}}, \ i \in \{1, 2, 3, ...n\} \tag{5}$$

As step 3 of PredictRV, using the probabilities computed on different intervals we compute the utility U_j to be bid by PredictRV as:

$$U_j \ = \sum_{i=1}^{n} p_{x_i} \ * T_{x_i j}, \ j \in \ \{1, 2, 3, ...N\} \tag{6}$$

3.7 Bayesian Learning with Regression Analysis (BLRA)

In the BLRA procedure presented in [25], the learning agent i has a belief about the p.d. of its opponent's negotiation parameters (i.e., the deadline and RV). As shown in step 1 of PredictRV, we have a belief over the hypothesis of our own (dynamic) RV. By keeping track of the history of values obtained for RV so far and comparing it with fitted estimates derived from a regression analysis, the agent can revise its belief over the hypothesis by using a Bayesian updating rule and can correspondingly adapt its concession strategy.

Regression Analysis. As the negotiation proceeds [25], utility u_t for a TDT decreases according to the following decision function:

$$u_t = 1 - (\frac{t}{T})^{\beta} \tag{7}$$

where T is the deadline and β is the concession parameter. We adopt this terminology to express in terms of agent A's own dynamic RV. We assume RV to be 0 at the start of the negotiation and vary according to Eq. (7).

$$u_t = u_0 + (u_T - u_0)(\frac{t}{T})^{\beta} \tag{8}$$

where u_T is the RV at the deadline and u_0 is the RV at the start. For every round, we receive an RV for that round. We compute the regression line (fitted utilities) $\hat{RV}_{t_b} = \{\hat{u}_0, \hat{u}_1, \hat{u}_2, ..., \hat{u}_{t_b}\}$ based on the historical RVs, $RV_{t_b} = \{u_0, u_1, u_2, ..., u_{t_b}\}$ until round t_b as follows:

Step 1: Generate the hypotheses and initialize its probabilities as mentioned in Sect. 3.5 (Steps of strategy) with x_i representing the utility of each hypothesis.
Step 2: Based on Eq. (8), we use the following power regression function to calculate the regression curve:

$$\hat{u}_t = u_0 + (x_i - u_0)(\frac{t}{N})^{\beta} \tag{9}$$

where N is the deadline. Next, β is calculated using Eq. (10) (as proposed in [25]):

$$\beta = \frac{\sum_{k=1}^{t_b} t_k^* u_k^*}{\sum_{k=1}^{t_b} t_k^{*2}}, \text{where}\, u_k^* = ln(\frac{u_0 - u_k}{u_0 - x_i}), t^* = ln(\frac{t}{N}) \tag{10}$$

Step 3: Based on the calculated regression curve given by Eqs. (9) and (10), the fitted RVs, $\hat{RV}_{t_b}$ would be $= \{\hat{u}_0, \hat{u}_1, \hat{u}_2, ..., \hat{u}_{t_b}\}$ at each round (where $\hat{u}_0 = u_0$).
Step 4: We now calculate the non-linear correlation between RV_{t_b} and the fitted RVs $\hat{RV}_{t_b}$. The coefficient of non-linear correlation γ is given by Eq. (11), where $\overline{u}$ and $\overline{\hat{u}}$ are the average of all the historical and fitted RVs respectively:

$$\gamma = \frac{\sum_{k=1}^{t_b}(u_k - \overline{u})(\hat{u}_k - \overline{\hat{u}})}{\sqrt{\sum_{k=1}^{t_b}(\hat{u}_k - \overline{\hat{u}})^2 \sum_{k=1}^{t_b}(\hat{u}_k - \overline{\hat{u}})^2}}, \gamma_{new} = \frac{\gamma + 1}{2} \tag{11}$$

Step 5: Parameter γ $(-1 \leq \gamma \leq 1)$ is used for evaluating resemblance between chosen (x_i) and real RVs (u_t). To use γ as a probability to perform belief update in Bayesian Learning, we normalize it to [0,1] (γ_{new} in Eq. (11)).

Bayesian Learning

Step 1: Bayesian Learning can be used if we have a hypothesis about the prediction. Belief about p.d. of these hypotheses can be revised through a posterior probability by observing the RV. Each hypothesis H_i represents that it would be the possible RV at the end of negotiation. The prior p.d., denoted by P(H_i), i ∈ (1, 2, 3, ..., n) signifies the agent's belief about the hypothesis i.e., how likely the hypothesis matches the RV at the end of the negotiation.
Step 2: The agent can initialize the p.d. over hypotheses based on some prior information if available, otherwise a uniform distribution P(H_i) $= \frac{1}{n}$ is assigned. During each round of negotiation t_b the probability of each hypothesis would be computed using the Bayesian updating rule in Eq. (12):

$$P(H_i|RV) = \frac{P(H_i)P(RV|H_i)}{\sum_{k=1}^{n} P(RV|H_k)P(H_k)} \tag{12}$$

Step 3: The observed outcome here is historical RVs $RV_{t_b} = \{u_0, u_1, u_2, ..., u_{t_b}\}$. As presented in [25], the agent will update the prior probability P(H_i) using the posterior probability P($H_i|RV_{t_b}$), thus a more precise estimate is achieved using Eq. (12).
Step 4: As presented in [25], conditional probability P($RV_{t_b}|H_i$) is obtained by comparing the fitted points $\hat{RV}_{t_b}$ on the regression line based on each selected RV x_i, with the historical RVs RV_{t_b}. The more correlated fitted RVs are with historical RVs, the higher P($RV_{t_b}|H_i$) will be.
Step 5: Difference between the regression curve and the real RV sequence can be indicated by the non-linear correlation coefficient γ_{new}. Thus, we can use the value of γ_{new} as the conditional probability P($RV|H_i$) in Eq. (12). The learning approach will increase the probability of a hypothesis when the RV selected (x_i) is most correlated with the RV at the end of the negotiation. As mentioned in step 4 of PredictRV, using the probabilities on different intervals, we compute the utility at that round as:

$$U_j \ = \sum_{i=1}^{n} P(H_i) \ * T_{x_i j}, \ \ j \in \ \{1, 2, 3, ...N\} \tag{13}$$

3.8 LSTM Based Prediction

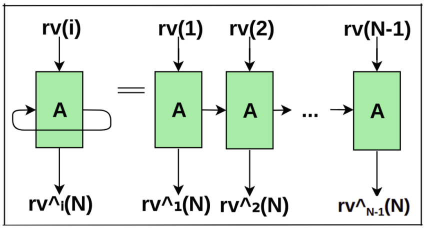


Fig. 2. LSTM architecture

LSTM (Long-Short Term Memory) [13] is a popular recurrent neural network architecture to perform deep learning tasks and is useful in time-series prediction. The negotiation problem introduced here can be modeled as a time series prediction task wherein the agent learns more information as the negotiation progresses. We, therefore, propose to use an LSTM based approach to predict the RV at the last time step n of the negotiation, using time-series forecasting. As shown in Fig. 2, the input at each time step t for LSTM is $RV(t)$ (i.e., RV provided by the environment at t). Note that there exists a single LSTM cell A to which input is fed repeatedly (one value at every time step) along with the output of the previous time step. Output at t is the predicted value for RV at the last time step n denoted by $\hat{RV}_t(n)$. The LSTM is trained using a mean squared error loss function and learns to predict better as the number of epochs increases. There are n hypotheses in our problem whose probability is updated every time step based on the predicted RV for the last time step $\hat{RV}$. This is similar to Counter model where we identify the interval the $\hat{RV}$ falls into and increase the count of that hypothesis by 1 (Eq. (5)). Using the probabilities for different hypotheses we compute the utility to be bid by PredictRV (Eq. (6)).

4 Example Continued

The rest of the example is explained using the ONAC-D algorithm (ONAC-D is ONAC strategy without any changes applied to Dynamic RV). Figure 4 shows the utility values generated by Counter, BLRA and LSTM models computed using Eqs. (6) and (13) respectively. The x-axes shows the number of rounds from 0 to 100 while the y-axes shows the utility values ranging from 0 to 1.

Figure 3 shows the belief plots for the three models. A belief plot shows how the belief in a particular hypothesis changes as the rounds progress. The figure shows two plots corresponding to the two hypotheses that the RV is 0.1 (hypothesis 0.1) and 0.9 (i.e., hypothesis 0.9). The x-axes for both the figures show the number of rounds from 0 to 100 and the y-axes show the probability of belief in the hypothesis that the figure represents e.g., a y-axis value of 0.3 in figure on left implies that an algorithm believes that the RV is 0.1 with a probability 0.3 which implies that other hypotheses are true with rest of the probability (in this case only other hypothesis is hypothesis 0.9). The belief plots show that:

(a) For hypothesis 0.1, while Counter stays close to middle (probability of 0.5), BLRA and LSTM are more clear in their belief for this hypothesis (former

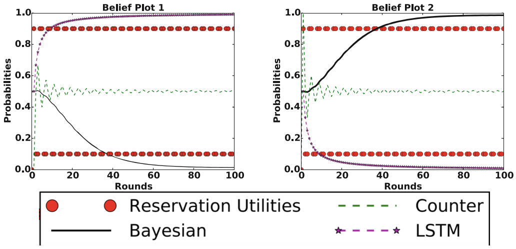


Fig. 3. Belief plots for two hypotheses

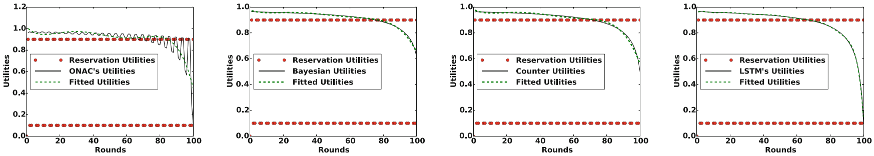


Fig. 4. Algorithms with their fitted curves

converges to close to 0 while the latter converges to close to 1 probability and stay with these probabilities once converged) showing the inherent differences between the models. (b) Counter converges quickly to a belief of 0.5 since RV alternates between the hypotheses every 2 steps, hence the count is more or less balanced. (c) For BLRA, belief in hypothesis 0.1 converges close to 0 since it is not just the count but the time when the RV changes come into play here. (e) For LSTM, belief in hypothesis 0.1 converges to close to 1 faster than other models, however to the opposite belief of BLRA for this example. The outcome utility for $\langle$ONAC-D, Counter$\rangle$ is $\langle 0.5, 0.5\rangle$, $\langle$ONAC-D, Bayesian$\rangle$ is $\langle 0.25, 0.75\rangle$ and $\langle$ONAC-D, LSTM$\rangle$ is $\langle 0.6, 0.4\rangle$.

5 Experiments

5.1 Setup for the Experiments

We have a number of hypotheses, number of rounds of negotiation N and update rate (frequency of change in RV) as the parameters of our algorithm. N is fixed to 100 for all experiments. Experiments were performed on the Fire Disaster Response and Meeting Scheduling domains. In both these domains, the agent is faced with a dynamic RV. For purposes of experimentation, we model the dynamic RV using a Markov chain model [we omit the specifics of our modeling due to space constraints]. The number of hypotheses vary across the domains. Update rate of RV is varied among the values $\{2, 5, 10, 20, 50\}$. We run each experiment for 100 iterations keeping the parameters constant. X-axis shows the (hypothesis, update rate) while y-axis shows the respective metric in each plot.

5.2 Metrics

1) **Outcome Utility Metric:** We run negotiations for agent A vs agent B, where A uses one of ONAC-D or Boulware-D and B is PredictRV strategy (Counter, BLRA or LSTM). We average the outcome over 100 iterations and compute the outcome utility for each UpdateRate and hypothesis (averaged utility represented as OD for ONAC-D, C for Counter, B for BLRA and L for LSTM). We then compute the utility of PredictRV w.r.t ONAC-D using Eq. (14) (represented in graphs as Average Percentage Utility):

$$\text{percentage utility of i} = \frac{i - OD}{OD} * 100, i \in \{C, B, L\} \tag{14}$$

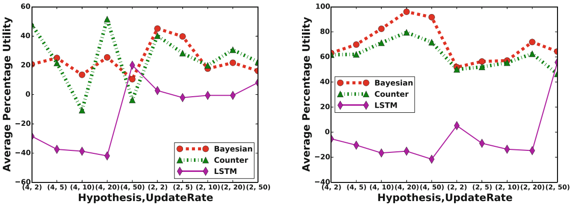
2) **Prediction Metric:** We allow each model to train until the end of the negotiation (N rounds). At the last round N, we have an RV predicted by each of the models i.e $\hat{RV}$ for round $N+1$ (which is not part of the negotiation). For each of the models, we then compute the difference between $\hat{RV}$ and the actual RV at round $N+1$ which is used to capture the quality of prediction. This value is averaged over 100 iterations where a lower difference in average value implies a better prediction.

5.3 Fire Disaster Response

Consider a forest fire where the fire can spread quickly in any of the 4 directions i.e., North, South, East or West. Assume that the forest is modeled as a grid of size $n_1{}^*n_1$ [11]. Fire fighting units (local units) are dispatched to many locations to fight the fire. The commander in charge has a global picture of the fire and wants to reduce the resources given to each local unit. The local unit leader (modeled as agent) would like to negotiate with the commander to obtain higher (than minimal needed to just put off the fire) number of resources to stop the fire quickly at the local point. Given that the direction of fire changes in different time steps, the RV is dynamic i.e. changes with time.

We operationalize the experimental parameters as follows: A negotiation is being carried out with N (=100) as the deadline. The parameters here are number of hypotheses, the update rate of the RV and grid size. The number of hypotheses are varied among the values $\{2, 4\}$ i.e., {North, South} or {North, West, East, South} directions with $\{0.75, 0.15\}$ and $\{0.75, 0.57, 0.32, 0.15\}$ (corresponding to number of resources {12, 10, 7, 4}) as the values for RV. Experiments were performed with the local location start point, as a random point around the center of the grid (up to a radius of 4 units from the center).

Figure 5 shows two plots corresponding to ONAC and Boulware with a random start point for fire with grid size 100. Both plots show that the values for outcome utility metric (Sect. 5.2) for PredictRV are higher than for ONAC-D or Boulware-D respectively e.g., plot (b) of Fig. 5 shows that the Average Percentage Utility for BLRA varies from 50% at the lowest to 95% at the highest. The plot also shows that the overall Average Percentage Utility across all the intervals and update rates for BLRA is 71% while it is 61% for Counter and −4.4% for LSTM.



(a) ONAC Strategy (b) Boulware Strategy

Fig. 5. Outcome utility in fire domain with random start points

To showcase the statistical significance of the outcome utility results presented in Fig. 5 of the paper, we performed a paired valued t-tests for the following settings (where O: ONAC, Bo: Boulware, Ba: Bayesian, C: Counter, L: LSTM): In PredictRV experiments for O: Ba vs. O, C vs. O, L vs. O, Ba vs. C, Ba vs. L, C vs. L. In PredictRV experiments for Bo: Ba vs. Bo, Co vs. Bo, L vs. Bo, Ba vs. C, Ba vs. L, C vs. L. If the calculated P-value is less than 0.05, it means that statistically the mean difference (in outcome utility as shown in Fig. 5) between the paired observations is different. Our testing showed that mean values of outcome utility for Bayesian vs. Counter does not have significant difference statistically (both for ONAC and Boulware i.e., 2 tests). All the other (10) tests, showed that the differences in (the averaged) outcome utility are statistically different.

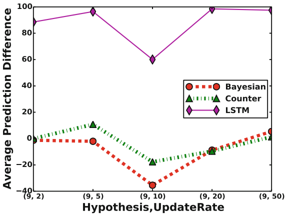


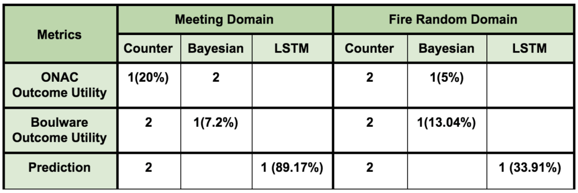
Fig. 6. Prediction in the meeting domain

5.4 Meeting Scheduling Domain

For brevity purposes, we present the gist of the domain here: We operationalized parameters for this domain from the E-Elves [23] application. The parameters for the algorithm are the update rate of the RV, delay intervals and number of hypotheses. There are 9 possible delay intervals we consider here i.e $\{5, 10, 15, 20, 25, 30, 35, 40, 45\}$ min. A delay interval of 10 min means that a meeting supposed to start at 10 am is now rescheduled to start at 10:10 am. For this domain each hypothesis corresponds to a delay interval, hence 9 hypotheses correspond to 9 delay intervals. The overall value of the meet is computed as below:

$$\begin{aligned} Delaycost &= (delay^{alpha}) * 2, Value\, of\, the\, meet = 200 \\ Overall\, value &= Value\, of\, meet - Delaycost \end{aligned} \tag{15}$$

where *delay* is delay w.r.t. the scheduled starting time and $alpha \in \{1.0, 1.2, 1.4, 1.6\}$. Utility of the hypothesis is calculated by normalizing the reward obtained using Eq. (15). The prediction measurement for the meeting domain is shown in Fig. 6. Experiments for measuring prediction for the meeting domain were performed with the following summary (we skip graphs due to



Metrics	Meeting Domain			Fire Random Domain		
	Counter	Bayesian	LSTM	Counter	Bayesian	LSTM
ONAC Outcome Utility	1(20%)	2		2	1(5%)	
Boulware Outcome Utility	2	1(7.2%)		2	1(13.04%)	
Prediction	2		1 (89.17%)	2		1 (33.91%)

Fig. 7. Relative performance table (ranks)

space issues): The overall average percentage prediction across all the intervals and the update rates for Counter, BLRA and LSTM are -3.05, -8.27 and 88.12 respectively.

5.5 Summary of the Experiments

In Fig. 7, 1 signifies the best performing model and 2 signifies the second-best performing model for a given metric and domain, $x\%$: how much better the best model is relative to the second-best model. Formulation: $a =$ metric value of best model, $b =$ metric value of 2nd best model. For the outcome utility relative performance $= 100 * \frac{(a-b)}{a}$, $(a > b)$. For the prediction metric, relative performance $= 100 * \frac{(b-a)}{b}$, $(b > a)$.

Explanation: For each of the metrics, we measure the relative value of the best performing model w.r.t the second-best performing model for each domain. For example, in the Fire Random domain for the ONAC algorithm, BLRA is 5% better than Counter on outcome utility metric and LSTM is 33.91% better than Counter on the prediction metric.

6 Conclusions

We introduced the PredictRV strategy which uses one of Counter, BLRA or LSTM learning models that predict over the dynamic RV to perform a better negotiation. Our results show that: a) For Outcome Utility: the BLRA model performs slightly better than Counter although the difference is not statistically significant. b) For Prediction metric: LSTM is the best performing model while Counter performs next best. c) Outcome utility is the standard metric that is used to evaluate negotiations. Given that both BLRA and Counter methods perform well on this metric, they can be tested for the specific use case needed and one of them picked based on the insights obtained. In summary, the key novelty of our work is that we enhance the ability of current negotiation algorithms to handle dynamic RV. The problem can be more general where only an indicator function for the RV is available rather than the actual value at each update step as assumed here. Popular negotiation platform such as Genius allows us to encode static RV currently – we believe this work takes a significant step towards dealing with challenges in handling dynamic RV.

References

1. Baarslag, T., Hadfi, R., Hindriks, K., Ito, T., Jonker, C.: Optimal non-adaptive concession strategies with incomplete information. In: Fukuta, N., Ito, T., Zhang, M., Fujita, K., Robu, V. (eds.) Recent Advances in Agent-based Complex Automated Negotiation. SCI, vol. 638, pp. 39–54. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30307-9_3
2. Baarslag, T., Hindriks, K., Jonker, C.: Effective acceptance conditions in real-time automated negotiation. Decis. Support Syst. **60**, 68–77 (2014)
3. Chalupsky, H., et al.: Electric elves: agent technology for supporting human organizations. AI Mag. **23**(2), 11 (2002)
4. Choi, S.P., Liu, J., Chan, S.P.: A genetic agent-based negotiation system. Comput. Netw. **37**(2), 195–204 (2001)
5. Coehoorn, R.M., Jennings, N.R.: Learning on opponent's preferences to make effective multi-issue negotiation trade-offs. In: Proceedings of the 6th International Conference On Electronic Commerce, pp. 59–68. ACM (2004)
6. Crump, L., Moon, D.: Precedents in negotiated decisions: Korea-Australia free trade agreement negotiations. Negot. J. **33**(2), 101–127 (2017)
7. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. Robot. Auton. Syst. **24**(3–4), 159–182 (1998)
8. Fatima, S., Kraus, S., Wooldridge, M.: Principles of Automated Negotiation. Cambridge University Press, Cambridge (2014)
9. Fatima, S., Wooldridge, M., Jennings, N.R.: Optimal negotiation of multiple issues in incomplete information settings. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 3, pp. 1080–1087. IEEE Computer Society (2004)
10. Fatima, S.S., Wooldridge, M., Jennings, N.R.: An agenda-based framework for multi-issue negotiation. Artif. Intell. **152**(1), 1–45 (2004)
11. Heyman, D.P., Sobel, M.J.: Stochastic Models in Operations Research: Stochastic Optimization, vol. 2. Dover Publication, New York (1982). Courier Corporation
12. Hindriks, K., Tykhonov, D.: Opponent modelling in automated multi-issue negotiation using bayesian learning. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 331–338. International Foundation for Autonomous Agents and Multiagent Systems (2008)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Wooldridge, M.J., Sierra, C.: Automated negotiation: prospects, methods and challenges. Group Decis. Negot. **10**(2), 199–215 (2001)
15. Kraus, S.: Strategic Negotiation in Multiagent Environments. MIT Press, Cambridge (2001)
16. Li, M., Vo, Q.B., Kowalczyk, R., Ossowski, S., Kersten, G.: Automated negotiation in open and distributed environments. Expert Syst. Appl. **40**(15), 6195–6212 (2013)
17. Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K., Jonker, C.M.: Genius: an integrated environment for supporting the design of generic automated negotiators. Comput. Intell. **30**(1), 48–70 (2014)
18. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press, Cambridge (1982)
19. Rosenfeld, A., Kraus, S.: Predicting human decision-making: from prediction to action. Synth. Lect. Artif. Intell. Mach. Learn. **12**(1), 1–150 (2018)

20. Rosenschein, J.S., Zlotkin, G.: Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. MIT Press, Cambridge (1994)
21. Rubinstein, A.: Perfect equilibrium in a bargaining model. Econom.: J. Econom. Soc. **50**, 97–109 (1982)
22. Sandholm, T., Lesser, V.R., et al.: Issues in automated negotiation and electronic commerce: extending the contract net framework. ICMAS **95**, 12–14 (1995)
23. Scerri, P., Pynadath, D.V., Tambe, M.: Towards adjustable autonomy for the real world. J. Artif. Intell. Res. **17**(1), 171–228 (2002)
24. Scheaffer, R.L., Young, L.: Introduction to Probability and its Applications. Cengage Learning, Boston (2009)
25. Yu, C., Ren, F., Zhang, M.: An adaptive bilateral negotiation model based on bayesian learning (2013)
26. Zeng, D., Sycara, K.: Bayesian learning in negotiation. Int. J. Hum.-Comput. Stud. **48**(1), 125–141 (1998)