Solving Negotiation Problems Against Unknown Opponents with Wisdom of Crowds

Siqi Chen[1(✉)], Gerhard Weiss[2], and Shuang Zhou[2]

[1] School of Computer and Information Science, Southwest University, Chongqing, China
siqi.chen09@gmail.com

[2] Department of Knowledge Engineering, Maastricht University, Maastricht, The Netherlands
{gerhard.weiss,shuang.zhou}@maastrichtuniversity.nl

Abstract. For a successful automated negotiation, a vital issue is how well the agent can learn the latent preferences of opponents. Opponents however in most practical cases would be unwilling to reveal their true preferences for exploitation reasons. Existing approaches tend to resolve this issue by learning opponents through their observations during negotiation. While useful, it is hard because of the indirect way the target function can be observed as well as the limited amount of experience available to learn from. This situation becomes even worse when it comes to negotiation problems with large outcome space. In this work, a new model is proposed in which the agents can not only negotiate with others, but also provide information (e.g., labels) about whether an offer is accepted or rejected by a specific agent. In particular, we consider that there is a crowd of agents that can present labels on offers for certain payment; moreover, the collected labels are assumed to be noisy, due to the lack of expert knowledge and/or the prevalence of spammers, etc. Therefore to respond to the challenges, we introduce a novel negotiation approach that (1) adaptively sets the aspiration level on the basis of estimated opponent concession; (2) assigns labeling tasks to the crowd using online primal-dual techniques, such that the overall budget can be both minimized with sufficiently low errors; (3) decides, at every stage of the negotiation, the best possible offer to be proposed.

1 Introduction

Negotiation has traditionally been investigated in game theory [17,18], and in previous years it has also developed into a core topic of multiagent systems [1,5,15,16,20]. Generally speaking, it is a process by which parties of conflicting interests try to reach a mutually acceptable agreement [13]. In many cases, it is however expensive and low efficient mainly because humans find the activity challenging, stressful as well as time-consuming. Thus, to alleviate huge negotiation efforts of humans, autonomous agents are proposed that perform, on behalf of humans, complicated negotiation tasks in a efficient manner. Automated negotiation also provides one of the most fundamental and powerful mechanisms for

G. Friedrich et al. (Eds.): KI 2016, LNAI 9904, pp. 126–133, 2016.
DOI: 10.1007/978-3-319-46073-4_10

intelligent systems, e.g., managing inter-agent dependencies, coordination and cooperation. With the rapid development of automated negotiation in recent decades, it has successfully gained a broad spectrum of applications in industrial and commercial domains [6,10,19,20].

The driving force of negotiating agents is governed by its (hidden) preferences through its (hidden) negotiation strategy [4]. By exploiting the preferences and/or strategy of opposing agents, better final (or cumulative) agreement terms can be reached [3]. Existing literature [7–9,11] attempts to achieve that by means of learning the behavior/preferences of opponents through observations of opponent negotiation moves. Although useful, learning an opposing agent's model is not efficient, mainly because: (1) the opponent preference can only be observed indirectly through offer exchanges (e.g., our rejected offers and opponent counter offers), (2) the absence of prior information about opponent strategy/preferences, and (3) the confinement of the interaction number/time in single negotiation sessions. Apparently, this kind of learning methods somehow restrict agents' learning ability.

Thus, we consider a general negotiation model in which automated agents not only carry out negotiation with others, but also provide advice (e.g., in terms of binary label) on whether an offer is accepted or rejected by a specific agent in a ongoing negotiation, according their knowledge and experience. Each label on offers from the crowd is associated with a certain (but low) cost, while the overall budget is limited; moreover, the collected labels are assumed to be noisy, due to labeling agents' lack of expert knowledge regarding negotiation problems or the target agents, and the prevalence of spam, etc. That is to say, each agent may have different and unknown reliability. Therefore to infer the true labels from the non-expert crowd, an assignment strategy is needed to allocate tasks to the crowd, to minimize the labeling budget, while guaranteeng sufficiently low errors. With the labels in hand, the negotiating agent is more likely to make decisions toward reaching efficient agreements.

2 Negotiation Approach

The actions of the agent at each time point should take into account, (1) the aspiration level, which governs the minimum amount of expected satisfaction from negotiation at a time point, and (2) what offer to accept or reject given that. The decision-making process is decomposed into two stages – aspiration setting (AS) component and offer responding (OR) component, which are essential and vital for the agent to operate successfully. The aspiration setting (AS) component is described. It adopts a non-parametric and computationally efficient regression technique in order to approximate the opponent's negotiation strategy. This allows the agent to have accurate estimates that are used to adjust its own aspiration level. The second stage of the approach (i.e., the offer responding (OR) component) deals with how to respond to those counter-offers and determines what counter-offer to send out if not satisfied with proposals from opponents.

When selecting offers of interest, the agent adopts an adaptive assignment strategy to ask information from the crowd so that the preferences of opponents over offers could be well learnt. Next, each of the above components is detailed.

2.1 Aspiration Setting Component

As opponent strategies are unavailable to the agent, it may be beneficial to adaptively set aspiration level R(t) according to negotiation dynamics, which specifies the lowest utility expectation of the agent. Toward this end, we adopt Gaussian process to obtain opponent strategy in terms received concession, which is proved to be successful in a variety of negotiation scenarios [4,8], while we refine a simplified version here to get rid of tuning a bunch of parameters (overfitting).

The agent uses the expected received utility $E_(t)$ in its decision making. This utility, which corresponds to the expectation of how much profit can be received from an opponent at some future time $t_\star$, is defined by:

$$E(t_\star) = \frac{t_\star}{NC} \int_{-\infty}^{+\infty} u \cdot f(u; \mu_\star, \sigma_\star) du \tag{1}$$

where NC is a constant called normalizing constant, f is the probability density function of Gaussian distribution, and $\mu_\star$ and $\sigma_\star$ are the mean and standard deviation (both obtained from GPs) at $t_\star$. Unlike the approach described in [21], which truncates the probability distribution to $[0, 1]$, the agent preserves the probability distribution by introducing the normalizing constant C.

$$R(t) = u_{\text{res}} + (U_{\max} - u_{\text{res}})(1 - t)^\beta \tag{2}$$

where $u_{\text{res}} = \min(\theta, \xi)$ (with ξ the maximal received concession), and concession coefficient controlling the concession rate is given by,

$$\beta = 1 - (\frac{E(t_\star)}{U_{\max}})^2 \tag{3}$$

where $U_{\max}$ is the possible maximum utility in the scenario.

2.2 Offer Responding Component

Having obtained the aspiration level, the agent then needs to decide acceptance or rejection of opponent offers. If the opponent offer can provide a utility higher (or at least equal to) than the R(t), the agent agrees with the offer and the negotiation is finished successfully; otherwise, the agent should prepare a counter offer to continue the negotiation. The two steps are detailed next.

Negotiation Decision-Making. Given the expected utility of $R(t)$, the agent needs to examine one of two conditions in response to the opponent. In the first the agent has to validate whether the utility of the counter-offer $U(O_{opp})$ is better than u', while in the second the agent has to determine whether the opponent had already proposed this offer earlier in the negotiation process. If either one of these two conditions is satisfied, the agent accepts it and terminates the session as shown in line 12 of Algorithm 1.

Otherwise, if none of them are met, the agent proposes a new offer depending on an λ-greedy strategy. That is to select either a greedy action (i.e., exploit) with λ probability or to select a random action with a $1 - \lambda$ probability ($0 \leq \lambda \leq 1$). The greedy action is determined based on the advice of crowds, that is, labels of acceptance or rejection on offers provided by a large amount of related agents. Those agents either have negotiation experience with the agent's opponents or have certain domain knowledge. Unfortunately, the labels may be noisy due to the lacking expertise and/or different reliability among them. It usually makes labels generated by crowd suffer from low quality. Moreover, each label is produced at certain cost. In the next subsection, we will dive into details of how to adaptively assign tasks to crowded agents. With a probability λ, agent then picks the offer whose gets the best negotiation value.

In the case of the random action (probability $1 - \lambda$), the agent constructs a new offer which has an utility within some range around u'. The main motivation behind this choice is twofold: (1) it is possible, for multi-issue negotiations, to generate a number of offers whose utilities are the same or very similar to the offering agent, with granting the opposing negotiator different utilities, and (2) it is sometimes not possible to find an offer whose utility is exactly u'. Thus it is reasonable that an agent selects an offer whose utility is in the narrow range $[(1 - 0.005)u'$,$(1 + 0.005)u']$. If no such solution can be found, the agent repeats sending the latest bid in the next round.

Adaptive Assignment Strategy. Crowdsourcing services, as a remedy for noisy labels, usually resort to labeling redundancy – collecting labels from different workers for each item [14,22]. A fundamental issue for crowdsourcing in negotiation is then raised: how to make crowdsourced task assignments such that it can output desired labels with sufficiently low error, while requesting as few labels from workers as possible. Toward this end, we apply the technique proposed in [2,12] to solve crowdsourced task assignment for automated negotiation setting. Prior to task assignment, the agent first decides upon which part of the outcome space to explore via crowdsourcing. In our approach, the exploration zone first excludes offers of utility below reservation value (θ) and offers of first K highest values (by sorting the possible outcomes according to our agent's own preferences), and then selects offers randomly from the remaining offers according to the given budget. When having collected enough opponent responses (i.e., gold standard tasks) placed by Algorithm 1, the agent begins online crowdsourcing task assignment (e.g., each agent's reliability is unknown). The main steps of assignment strategy is given below.

Before diving into details of the assignment strategy, we introduce a simple model to capture workers' reliability: each worker (agent) w_j is characterised by a reliability $p_{i,j} \in [0, 1]$ for task t_i, and workers answer each question correct independently. Since errors are common among the low-paid workers, majority

Algorithm 1. Adaptive assignment strategy for negotiation tasks. s is the number of gold standard tasks. m is the number of workers, with n the number of tasks.

Require: $s, \overline{q}^*_{\min}, m, n$
while $K < s$ **do**
  collect more offer responses;
  recordOffers(t_c, O_{opp});
end while
pick up random γ percentage workers from m
calculate $\hat{q}_{i,j}$ for each agent in γ m
calculate $C_{\epsilon'}$ and obtain estimated task wight $\hat{x}^*$
for each agent j **do**
  calculate $\hat{q}_{i,j}$ using s gold standard tasks
  run primal approximation algorithm with $\hat{q}_{i,j}$ and $\hat{x}^*$
  assign agent j to tasks i if $y_{i,j} = 1$
end for
aggregate labels using weighted majority voting.
return labels

voting should be applied to their advice for a target reliability. Next, we show the error bound under this model using majority voting. Assume J_i is the set of workers assigned to task t_i, $X_{i,j}$ a random variable which represents the weighted label, with $w_{i,j}$ being the weight. Given a positive label (e.g., the value is 1), we have

$$X_{i,j} = \begin{cases} w_{i,j} & \text{with probability } p_{i,j}, \\ -w_{i,j} & \text{with probability } 1 - p_{i,j} \end{cases} \tag{4}$$

and $X_i = \Sigma_{j \in J_i} X_{i,j}$.

If $X_i \geq 0$ the task is predicted to have a label of 1, and 0 otherwise. Assume the true label is 1, bounding $\mathrm{P}(X_i \geq 0)$ would give us a bound on the probability of an error. The expectation of X_i can be expressed as below,

$$\begin{aligned} E[X_i] &= \Sigma_{j \in J_i} E[X_{i,j}] \\ &= \Sigma_{j \in J_i} (p_{i,j} w_{i,j} - (1 - p_{i,j}) w_{i,j}) \\ &= \Sigma_{j \in J_i} (w_{i,j} (2p_{i,j} - 1)) \end{aligned} \tag{5}$$

Applying Hoeffding's inequality, we have

$$\begin{aligned} P(X_i \leq 0) &\leq exp(-\frac{2(E[X_i])^2}{\Sigma_{j \in J_i} (2w_{i,j})^2}) \\ &= exp(-\frac{(\Sigma_{j \in J_i} w_{i,j} (2p_{i,j} - 1))^2}{2\Sigma_{j \in J_i} w_{i,j}^2}) \end{aligned} \tag{6}$$

Obviously, this error bound is maximized when the right side of Eq. 6 is minimized. So, we set the gradient of this expression to 0.

Then, let $y_{i,j}$ be a variable to indicate the assignment of task t_i to worker w_j, with 1 representing positive and 0 negative. The requirement can be expressed as a linear constraint of these variables. This allows us to express the optimal assignment strategy as an integer linear program,

$$\begin{aligned} \min \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j} \cdot \eta && \text{(LP1)} \\ \text{s.t.} \quad & \sum_{i=1}^{n} y_{i,j} \le M_j \\ & \sum_{j=1}^{m} q_j y_{i,j} \ge C_\epsilon \\ & y_{i,j} \in 0,1 \qquad \forall (i,j) \end{aligned}$$

where η is the cost for each task. However, solving integer linear program requires the values q_j for each worker. It will be convenient to work with the dual of the linear program,

$$\begin{aligned} \max \quad & C_\epsilon \sum_{i=1}^{n} x_i - \sum_{j=1}^{m} M_j z_j - \sum_{i=1}^{n}\sum_{j=1}^{m} t_{i,j} && \text{(LP2)} \\ \text{s.t.} \quad & 1 - q_{i,j}x_i + z_j + t_{i,j} \ge 0 \qquad \forall (i,j) \\ & x_i, z_j, t_{i,j} \ge 0 \qquad \forall (i,j) \end{aligned} \tag{7}$$

Suppose that we were given access to the task weights x_i for each task i and the values $q_{i,j}$. Then we could use the following algorithm to approximate the optimal primal solution. Then, the agent should choose the offer whose utility not smaller than the utility indicated by $R(t_c)$, and whose label is positive, When needing to propose a counter-offer. If no such an offer can be found, the offer with the minimal utility (but not smaller than the utility indicated by $R(t_c)$) is proposed for greedy offer selection.

3 Conclusions

This work introduced a novel automated negotiation approach on the basis of opponent behavior prediction and crowdsourcing services. Opponent behavior prediction is captured by Gaussian processes to estimate future received concession, thereby governing the aspiration level function in an adaptive way. The deployment of crowdsourcing mechanism provides the agent with the wisdom of the noisy crowd, and using the assignment strategy, the labeling budget can be both minimized, while guaranteeing sufficiently low errors. It is clear that the agent is more efficient than the others due to a more advanced technical framework.

Acknowledgements. This work is supported by Southwest University and Fundamental Research Funds for the Central Universities (Grant number: SWU115032, XDJK2016C042). Special thanks also go to the anonymous reviewers of this article for their valuable comments.

References

1. Baarslag, T., Fujita, K., Gerding, E.H., Hindriks, K.V., Ito, T., Jennings, N.R., Jonker, C.M., Kraus, S., Lin, R., Robu, V., Williams, C.R.: Evaluating practical negotiating agents: results and analysis of the 2011 international competition. Artif. Intell. **198**, 73–103 (2013)
2. Buchbinder, N., Naor, J.: Online primal-dual algorithms for covering and packing. Math. Oper. Res. **34**(2), 270–286 (2009)
3. Chen, S., Ammar, H.B., Tuyls, K., Weiss, G.: Optimizing complex automated negotiation using sparse pseudo-input Gaussian processes. In: Proceedings of the 12th International Joint Conference on Autonomous Agents and Multi-agent Systems, Saint Paul, Minnesota, USA, pp. 707–714. ACM (2013)
4. Chen, S., Ammar, H.B., Tuyls, K., Weiss, G.: Using conditional restricted Boltzmann machine for highly competitive negotiation tasks. In: Proceedings of the 23th International Joint Conference on Artificial Intelligence, pp. 69–75. AAAI Press (2013)
5. Chen, S., Hao, J., Zili, Z., Weiss, G., Zhou, S.: Toward efficient agreements in real-time multilateral agent-based negotiations. In: 27th IEEE International Conference on Tools with Artificial Intelligence, pp. 896–903. IEEE Computer Society (2015)
6. Chen, S., Weiss, G.: An efficient and adaptive approach to negotiation in complex environments. In: Proceedings of the 20th European Conference on Artificial Intelligence, Montpellier, France, pp. 228–233. IOS Press (2012)
7. Chen, S., Weiss, G.: An efficient automated negotiation strategy for complex environments. Eng. Appl. Artif. Intell. **26**(10), 2613–2623 (2013)
8. Chen, S., Weiss, G.: An intelligent agent for bilateral negotiation with unknown opponents in continuous-time domains. ACM Trans. Auton. Adapt. Syst. **9**(3), 16:1–16:24 (2014)
9. Chen, S., Weiss, G.: An approach to complex agent-based negotiations via effectively modeling unknown opponents. Expert Syst. Appl. **42**(5), 2287–2304 (2015)
10. Duan, L., Dogru, M.K., Ozen, U., Beck, J.: A negotiation framework for linked combinatorial optimization problems. Expert Syst. Appl. **25**(1), 158–182 (2012)
11. Hao, J., Song, S., Leung, H.-F., Ming, Z.: An efficient and robust negotiating strategy in bilateral negotiations over multiple items. Eng. Appl. Artif. Intell. **34**, 45–57 (2014)
12. Ho, C.-J., Jabbari, S., Vaughan, J.W.: Adaptive task assignment for crowdsourced classification. In: Proceedings of the 30th International Conference on Machine Learning (ICML 2013), pp. 534–542 (2013)
13. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: prospects, methods and challenges. Eng. Appl. Artif. Intell. **10**(2), 199–215 (2001)
14. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowdsourcing systems. In: Advances in Neural Information Processing Systems, pp. 1953–1961 (2011)
15. Lopes, F., Wooldridge, M., Novais, A.: Negotiation among autonomous computational agents: principles, analysis and challenges. Artif. Intell. Rev. **29**, 1–44 (2008)

16. Mor, Y., Goldman, C.V., Rosenschein, J.S.: Learn your opponent's strategy (in polynomial time)!. In: Weiss, G., Sen, S. (eds.) IJCAI-WS 1995. LNCS, vol. 1042, pp. 164–176. Springer, Heidelberg (1996)
17. Osborne, M., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (1994)
18. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press, Cambridge (1982)
19. Sanchez-Anguix, V., Julian, V., Botti, V., Garcła-Fornes, A.: Tasks for agent-based negotiation teams: analysis, review, and challenges. Eng. Appl. Artif. Intell. **26**(10), 2480–2494 (2013)
20. Weiss, G. (ed.): Multiagent Systems, 2nd edn. MIT Press, Cambridge (2013)
21. Williams, C.R., Robu, V., Gerding, E.H., Jennings, N.R.: Negotiating concurrently with unkown opponents in complex, real-time domains. In: Proceedings of the 20th European Conference on Artificial Intelligence, pp. 834–839 (2012)
22. Zhang, Y., Chen, X., Zhou, D., Jordan, M.I.: Spectral methods meet EM: a provably optimal algorithm forcrowdsourcing. In: Advances in Neural Information Processing Systems, pp. 1260–1268 (2014)