Measuring of Strategies' Similarity in Automated Negotiation

L. Mashayekhy, M. A. Nematbakhsh, and B. T. Ladani, *University of Isfahan*

***Abstract*—Negotiation is a process between self-interested agents in ecommerce trying to reach an agreement on one or multi issues. The outcome of the negotiation depends on several parameters such as the agents' strategies and the knowledge one agent has about the opponents. One way for discovering opponent's strategy is to find the similarity between strategies. In this paper we present a simple model for measuring the similarity of negotiators' strategies. Our measure is based only on the history of the offers during the sessions of negotiation and we use a notion of Levenshtein distance. We implement this measure and experimentally show that the result of using this measure can improve the recognition of negotiation strategy. Also, this measure can be used for modeling behaviors of negotiators and predictive decision-making.**

***Index Terms*—Automated Negotiation, Behavior, Strategy, Similarity.**

I. Introduction

Automated negotiation is a key form of interaction in complex systems composed of autonomous agents. Negotiation is a process of making offers and counteroffers, with the aim of finding an acceptable agreement [8]. The agents (negotiators) decide for themselves what actions they should perform, at what time, and under what terms and conditions [2], [8]. The outcome of the negotiation depends on several parameters such as the agents' strategies and the knowledge one agent has about the opponents [2], [3], [4], and [10]. In recent years, the problem of modeling and predicting a negotiator behavior has become increasingly important since this can be used to improve negotiation outcome and increase satisfaction of result [2], [3], [4], and [12].

Similarity is a fundamental notion that has to be defined before one can apply various statistical, machine learning, or data mining methods [10]. Previous works have attempted to exploit the information gathered from opponent's offers during the negotiation to infer similarity between offers of the opponent to predict future offers. Bayesian classification [11] and similarity criteria work [2], [3] are examples of such efforts. When an agent has knowledge of opponent's strategy, he can use this knowledge to negotiate better deals for itself [8], [12]. However, an agent negotiates with incomplete information about opponent therefore using similarity between opponents' strategies makes this information for a negotiator [12]. The main problem is that there is not any measure for calculating similarity between negotiators' strategies.

Sequences of offers are a common form of data in negotiation that agent can use them to discover valuable knowledge in order to achieve its goal [2]. A session defined as ordered sequence of offers that an agent creates during negotiation based its strategy [3]. For finding similarity between negotiators' strategies, we use data of sessions. As data sequences, one method is to reduce sessions to points in a multi-dimensional space and use Euclidean distance in this space to measure similarity, but in negotiation, sessions do not have same lengths. One solution discussed in [5] for sequences, is to select *n* data of each sequence. The problem with this approach is which *n* offers in each session represent strategy of negotiator. Another method is to represent sessions in k-dimensional space using *k* features for each session [5]. Using the feature vector representing not only needs definition of features to model strategy of negotiator, but also the problem of sessions' similarity is transformed into the problem of finding similar feature in k-dimensional space.

In this paper we consider the problem of defining similarity or distance between strategies. We start with the idea that similarity between negotiators should somehow reflect the amount of work that has to be done to convert one negotiation session to another. We formalize this notion as Levenshtein or edit distance [5], [6] between negotiations. We apply a dynamic programming for computing the edit distances and show the resulting algorithm is efficient in practice.

In detail, the paper is organized as follows. In section 2 we present the problem in negotiations. The definition of similarity between negotiation strategies is given in section 3. In section 4 we make a review of the negotiation protocol used in our experimentation. We use some negotiation strategies in our simulation discussed in section 5. In section 6 we present some results of computing similarity measure. Section 7 is conclusions and remarks about future directions.

Manuscript received January 3, 2007. This work was completed at the University of Isfahan and supported by the Office of Graduate Studies. The authors are grateful to the Office for their support.

L. Mashayekhy is M.Sc. Student of Computer Engineering at the University of Isfahan, Isfahan, Iran (lmashayekhy@eng.ui.ac.ir).

M. A. Nematbakhsh is with the Computer Engineering Department, University of Isfahan, Isfahan, Iran (nematbakhsh@eng.ui.ac.ir).

B. T. Ladani is with the Computer Engineering Department, University of Isfahan, Isfahan, Iran (ladani@eng.ui.ac.ir).

II. Statement of Problem

One way of modeling negotiation is to consider a given set $S = (o_1, \ldots, o_m)$ of offers. S shows a negotiator exchange m offers during his negotiation session. An offer o is consists of one or multiple issues.

The basic problem we consider in this paper is how one should define a concept of similarity or distance between negotiation sessions. Such a notion is needed in any knowledge discovery application on negotiation. Exchanged offers during negotiation show strategy that the negotiator used in the session [2], [3], [4], [7] and [8]. For finding similar strategy of negotiators if one can not say when two negotiation sessions are close to each other, the possibility for making contrast is quite limited. For example, consider three buyers negotiate with a seller who wants to compare behavior of these buyers. The seller observation of these sessions (received offers) is shown in figure 1. Each of buyers has its initial offer, deadline and strategy to generate offers.

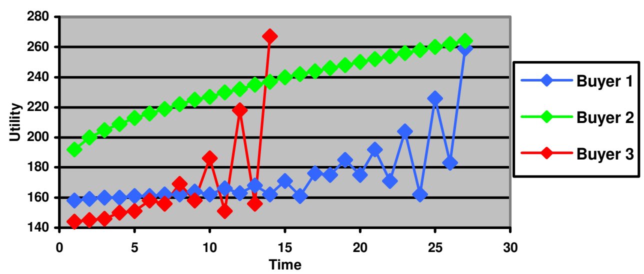

Fig. 1 Buyers' offers

Consider the problem of clustering these three buyers. When comparing two buyers to see if they are similar, we need a similarity measure. The meaning of similarity may vary depending on the domain and the purpose of using similarity. For example someone would group buyer 1 and 2 together, with buyer 3 as the out-group because of number of exchanged offers. But in this paper we want to define similarity of negotiators based on their strategy. When a seller observes received offers from different buyers are similar during their sessions, then this seller finds these buyers have similar strategies. In next section we discuss about this similarity measure.

III. Similarity Measure

In this section, we define two key concepts: first distance between two sessions and second distance between two offers.

A. *Distance between sessions*

We propose a new session similarity measure and use this measure for calculating the similarity between strategies of negotiators. Since offers are made during negotiation, we can refer them as sequences data.

The idea behind our definition of similarity, or distance, between negotiation sessions is that it should somehow reflect the amount of work needed to transform one negotiation session to another [5], [6]. The definition of similarity is formalized as edit distance $d(S, T)$ for two sessions S and T.

Operations: For counting the edit distance we need to define a set of transformation operations. We have chosen to use three operations:

- *ins(o)*: inserts an offer of the type o to the negotiation session.
- *del(o)*: deletes an offer of the type o from the negotiation session.
- *update(o, o')*: change an existing offer from o to o' in the negotiation session.

Cost of operations: instead of checking equality between two offers o_S and o_T from two sessions S and T respectively, for each operation we associate a cost *c(op)* based on distance of offers. The cost of an insertion operation is defined by (1) where o' is a previous offer of o in the negotiation session.

$$c(ins(o)) = distance(o', o), \qquad (1)$$

With this definition the cost of adding an outlying offer into the negotiation session is higher than the cost of adding in a neighboring offer. The cost of a deletion operation is defined to be the same as the cost of an insert-operation. It is proved that if the cost of insertion is equal to the cost of deletion then for each negotiation session S, T then we have $d(S, T)=d(T, S)$ [6].

The cost of an update-operation is defined as (2) where V is a constant value.

$$c(update(o, o')) = V.distance(o, o'), \qquad (2)$$

With this definition a low distance has a lower cost that a higher distance.

Definition of distance: If the cost of an operation op_i is $c(op_i)$, and k is the number of operations in the sequence Op_j, formula 3 below calculates the cost of operation sequence $Op_j = op_1, op_2, \ldots, op_k$.

$$c(Op_j) = \sum_{i=1}^{k} c(op_i) \qquad (3)$$

The distance $d(S, T)$ is defined as the sum of costs of the cheapest sequence of operations transforming S to the T as shown in (4).

$$d(S, T) = min\{c(Op_j) \mid \text{Op_j is an operation sequence transforming a session S to a session T}\} \qquad (4)$$

That is $d(S, T)$ is the minimum sum of costs of operations transforming S to T.

The problem of finding edit distance of two sessions (sequence of offers) S and T is solved using dynamic programming approach.

B. *Distance between offers*

Distance between two offers in insert, delete and update operation can be defined in a different way for each type of negotiation. Let o and o' be two offers.

In single issue negotiation that each offer has numeric value such as price, *distance*(o, o') is defined | o - o' |. For non-numeric issue distance can be calculated based on equality. In that case, the distance between any two offers is defined to be 0 if they are equal; and a positive number if they are not equal.

In multi issue negotiation, for each issue distance is calculated based on numeric or non-numeric value that discussed above. Then use Euclidean distance for calculating distance of offers. For instance if buyer and seller negotiate on price and delivery time, for calculating distance between two offers first calculate distance of price in each offer *d(p)* and then distance of delivery time in each offer *d(dt)*. Euclidean distance of *d(p)* and *d(dt)* set as distance of two offers. If issues have different importance, importance has influence on distance. Let $j \in \{1,\ldots,n\}$ be the issues under negotiation so offer o described as $(o_1, \ldots, o_n)$. The relative importance that an agent assigns to each issue under negotiation is modeled as a weight, w_j. Formula 5 shows how to calculate distance between two offers.

$$distance(o, o') = \sqrt{\sum_j w_j (o_j - o'_j)^2} \tag{5}$$

IV. Negotiation Protocol

We have created a simulation environment consisting of a set of agents equipped with a common protocol for bilateral negotiation. In this negotiation model, seller and buyer negotiate on price as a single issue. We adopt an alternating offers protocol; that is both of them can send and receive offers and decide whether accept or reject received offer until reach their own deadline [9], [10]. Each of them has incomplete information about his opponent.

Let $a \in \{b, s\}$ represent the negotiating agents and a' denote agent a's opponent. Let [min^a, max^a] denote the range of values for price that are acceptable to agent a. In this model min^b means initial price and max^b means reservation price of buyer and max^s means initial price and min^s means reservation price of seller. A value for price is acceptable to both agents if it is in the zone of agreement ([min^s, max^b]) [7]. This information is shown in figure 2.

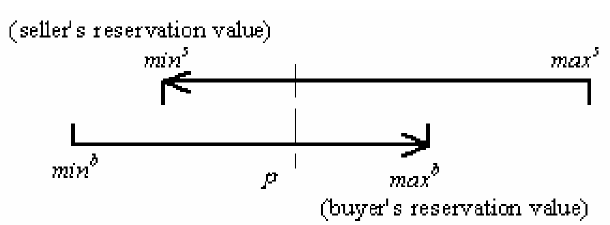

Fig. 2 zone of agreement

The agents alternately propose offers at times in $T = \{0, 1, \ldots\}$. Each agent has a deadline. T^a denotes agent a's deadline by when agent must complete the negotiation. Let $p^t_{b\rightarrow s}$ denote the price offered by agent b at time t. The agent who makes the first offer is chosen randomly. When an agent receives an offer from his opponent at time t, it rates the offer using its utility function U^a and responses that is defined as [4]:

$$Action^a(t, p^t_{a'\rightarrow a}) = \begin{cases} Quit & If\ t > T^a \\ Accept & \\ \quad If\ U^a(p^t_{a'\rightarrow a}) \geq U^a(p^{t+1}_{a\rightarrow a'}) & \\ Offer\ p^{t+1}_{a\rightarrow a'} & Otherwise \end{cases} \tag{6}$$

Offers are generated by agent's strategy which discussed in section 5.

If agent's deadline passes the agent withdraws from the negotiation. An agent accepts an offer when the value of the offered contract is higher than the offer which the agent is ready to send at that moment in time.

Agent's utility function is defined as:

$$U^a(p^t) = \begin{cases} \dfrac{max^a - p^t}{max^a - min^a} & if\ a = b \\ \dfrac{p^t - min^a}{max^a - min^a} & if\ a = s \end{cases} \tag{7}$$

A negotiation session between b and s at time t_n is a finite sequence of offers from one agent to the other ordered over time. The last element of the sequence is {*accept, reject*}.

V. Negotiation Strategies

We have implemented a number of negotiation strategies that comply with the protocol outlined. We use two types of strategies in our experiments in this paper.

The first strategy is Time dependent. This strategy is parameterized and hence it covers a large number of distinct strategies.

As time passes, the agent will concede more rapidly trying to achieve an agreement before arriving to the deadline. The offer to be uttered by agent a for a decision variable (price) at time t ($0 < t < T^a$) is computed as follows [8]:

$$p^t_{a\rightarrow a'} = \begin{cases} min^a + \varphi^a(t)(max^a - min^a) \\ \quad if\ a = b \\ min^a + (1 - \varphi^a(t))(max^a - min^a) \\ \quad if\ a = s \end{cases} \tag{8}$$

where $\varphi^a(t)$ is a function depending on time ($0 \leq \varphi^a(t) \leq 1$) and parameterized by a value β.

$$\varphi^a(t) = \left(\frac{t}{T^a}\right)^{\frac{1}{\beta}} \tag{9}$$

A wide range of time-dependent strategies can be defined by varying the way in which $\varphi^a(t)$ is computed [3]. However, depending on the value of β, three qualitatively different patterns of behavior can be identified: boulware if $\beta < 1$, linear if $\beta = 1$ and conceder if $\beta > 1$.

The second strategy we use is called Behavior dependent. The key feature of this strategy is that it offers based on opponent's behavior [4].

$$p^{t+1} = \begin{cases} min^a & If \;\; P \leq min^a \\ max^a & If \;\; P > max^a \\ P & Otherwise \end{cases} \tag{10}$$

The parameter P determines the type of imitation to be performed. We can find the following families:

Relative Tit-For-Tat: the agent reproduces, in percentage terms, the behavior that its opponent performed $\delta > 1$ steps ago.

$$P = \frac{p^{t-2\delta}}{p^{t-2\delta+2}} p^{t-1} \tag{11}$$

Absolute Tit-For-Tat: the same as before, but in absolute terms.

$$P = p^{t-1} + p^{t-2\delta} - p^{t-2\delta+2} \tag{12}$$

Averaged Tit-For-Tat: the agent applies the average of percentages of changes in a window of size $\lambda \geq 1$ of its opponent's history.

$$P = \frac{p^{t-2\lambda}}{p^{t}} p^{t-1} \tag{13}$$

We compute the values for the decision variables under negotiation according to each strategy.

VI. Experimental Results

In this section we describe how we have evaluated the effectiveness of using this measure for discovering similar negotiator strategies under different negotiation situations. In this experiment we use 50 negotiation sessions. In each session buyer and seller negotiate for price and they choose one of implemented strategies that discussed above (Conceder, Linear, Boulware, Relative TFT, Absolute TFT, and Average TFT). Buyers and sellers save information about their strategies, outcome and all exchanged offers during process of negotiation. We show how this measure finds similar strategies.

After gathering data of all sessions, we choose data of buyers with accepted result for discovering similarity of these agents. We use our measure for generating distance of these sessions. After calculating all distances we use k-medoids algorithm [1] for clustering based on these distances to evaluate our measure. This method is helpful because center of each cluster is one of existing point.

Given the strategy of a buyer in his session, this experiment shows sessions which use same strategy for negotiation form one cluster. Figure 3 shows changing offers of all sessions in one cluster.

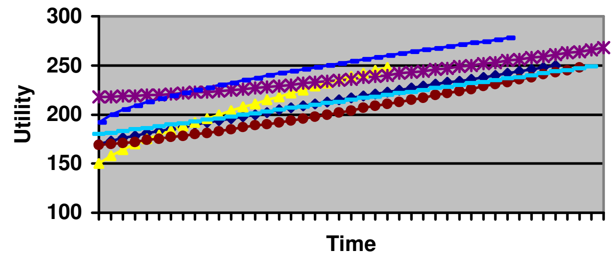

Fig. 3 Sessions in first cluster

After finding characteristics of these buyers, all of them use time dependent strategy. Figure 4 shows another cluster of sessions which all of them use behavior dependent strategy.

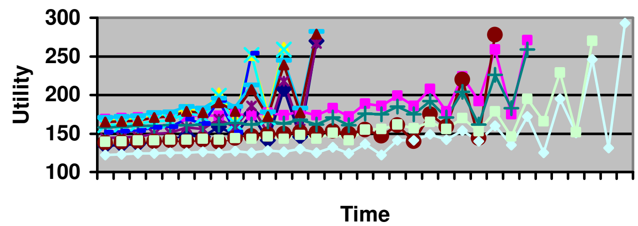

Fig. 4 Sessions in second cluster

The experiments are repeated with different numbers of clusters and with different negotiation strategies. In all experiments each cluster has buyers which use similar strategies.

As we mentioned above our experiment was based on data of buyers with accepted outcome, but for other data one can do same as this experiment.

In this paper we mainly consider a simplified model of negotiation, where each offer has only one issue. As we discussed in section 3 the model presented above can be extended for multi issue negotiation.

VII. Conclusion

The outcome of the negotiation depends on several parameters such as the agents' strategies and the knowledge one agent has about the others. Problem of modeling and predicting a negotiator behavior is important since this can be used to improve negotiation outcome and increase satisfaction of result. Finding similar behavior is one way to solve this problem. We have described a simple method for defining similarity between negotiation strategies. This method is based only on sequence of offers during negotiation. This characteristic gives the method significant practical value in negotiation for example the result can be used in knowledge discovery.

This method implemented using dynamic programming and it is tested in simple model of negotiation. The results of comparing our measure for finding similar strategies to chosen strategies are illustrated. Results show that this measure is efficient.

For the future, there are two ways in which this research can be extended. Firstly, we would like to consider the performance of our method against additional strategies. Secondly, in this work we only consider single issue negotiation model, our method could be applied to other negotiation models.

We plan to experimentally use this method for predicting opponent's strategy during negotiation.

REFERENCES

[1] Jiawei Han, Wicheline Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2000.

[2] R. M. Coehoorn, N. R. Jennings, "Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs". *In Proc. of the 6th International Conference on Electronic Commerce (ICEC2004)*, pages 113–120, Delft, The Netherlands, 2004.

[3] P. Faratin, C.Sierra, and N.R. Jennings, "Using Similarity Criteria to Make Issue Trade-Offs in Automated Negotiations". *In Artificial Intelligence 142*, pages 205-237, 2002.

[4] C. Hou, "Modelling Agents Behaviour in Automated Negotiation". *Technical Report KMI-TR-144. Knowledge Media Institute, The Open University, Milton Keynes, UK*, May 2004.

[5] Magnus Lie Hetland, "A Survey of Recent Methods for Efficient Retrieval of Similar Time Sequences". *First NTNU CSGSC*, May 2001.

[6] H. V. Jagadish, Alberto O. Mendelzon, and Tova Milo. Similarity-based queries. pp. 36–45, 1995.

[7] S.S. Fatima, M. Wooldridge, and N. R. Jennings, "An agenda-based framework for multi-issue negotiation". *Artificial Intelligence*, 152(1):1-45, 2004.

[8] P. Braun, J. Brzostowski, G. Kersten, J.B. kim, R. Kowalczyk, S. Strecker, and R. Vahidov, "e-Negotiation Systems and Software Agents: Methods, Models and Applications". *In J. Gupta; G. Forgionne; M. Mora (Hrsg.): Intelligent Decision-Making Support System: Foundation, Applications, and Challenges, Springer, Heidelberg ea., Decision Engineering Series*, 2006 (503 p., 105 illus., Hardcover. ISBN: 1-84628-228-4).

[9] Cuihong Li, Joseph Giampapa, and Katia Sycara, "A Review of Research Literature on Bilateral Negotiations". *In Tech. report CMU-RI-TR-03-41, Robotics Institute, Carnegie Mellon University*, Nov 2003.

[10] Hsiangchu Lai, Her-Sen Doong, Chi-Chung Kao, and Gregory E. Kersten, "Understanding Behavior and Perception of Negotiators from Their Strategies". *Hawaii International Conference on System Science*, 2006.

[11] Gerald Tesauro, "Efficient Search Techniques for Multi-Attribute Bilateral Negotiation Strategies". *In Proceedings of the 3rd International Symposium on Electronic Commerce*. Los Alamitos, CA, IEEE Computer Society, pp. 30-36 in 2002.

[12] Lena Mashayekhy, Mohammad A. Nematbakhsh, and Behrooz T. Ladani, "E-Negotiation Model based on Data Mining". *In Proceedings of the IADIS e-Commerce 2006 International Conference*, pp. 369-373, ISBN: 972-8924-23-2, Barcelona, Spain, December 2006.