Elicitation of User Preferences for Multi-attribute Negotiation

Yutao Guo, Jörg P. Müller
Intelligent Autonomous Systems
Siemens AG, Corporate Technology
81730 Munich, Germany
{yutao.guo.external, joerg.mueller}
@mchp.siemens.de

Christof Weinhardt
Department of Economics and
Business Engineering
University of Karlsruhe
76131 Karlsruhe, Germany
weinhardt@iw.uni-karlsruhe.de

ABSTRACT

Agents that act on behalf of users in electronic negotiations need to elicit the required information about their users' preference structures. Based on a multi-attribute utility theoretic model of user preferences, we propose an algorithm that enables an agent to learn the utility function with flexibility to accept several types of information for learning. The method combines an evolutionary learning with the application of external knowledge and local search. Empirical tests show that the algorithm provides a good learning performance.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Algorithms

Keywords

preference learning, negotiation

1. INTRODUCTION

Negotiation is a fundamental mechanism to automate business processes and to increase their flexibility. In many situations, decision processes in negotiations require consideration of multiple attributes. The challenge is to automate negotiation processes by using software agents [2], that act on behalf of humans or organizations in negotiations, and are endowed with an economic model (which can be modelled by a utility function) that guides their behavior in these negotiations.

Traditional methods for acquiring such an economic model requires users to explicitly specify their utility function, such as in the MARI framework [3]. Experiences indicate that most humans perform poorly in revealing their preferences explicitly. IBM Research developed the WORA [1] method to derive attribute weights from a user's ordinal ranking of the bids with linear programming, which usually gives multiple feasible solutions. We propose an utility elicitation framework to learn a complete multi-attribute utility function (i.e., attribute weights and attribute functions), with flexibility to accept several types of information for learning (e.g., rating, ranking, and partial knowledge). The method is based on an evolutionary framework, with the possibility to apply external knowledge, and with individual learning through simulated annealing for further refinement of the solution. The learning method reveals good performance in the simulated experiments. In particular, it shows that a substantial improvement of basic learning can be achieved by adding the steps of applying external knowledge and local search. This preference elicitation component is also implemented in the IntelliMarket agent marketplace infrastructure developed at Siemens Corporate Technology.

AAMAS'03, July 14–18, 2003, Melbourne, Australia.
ACM 1-58113-683-8/03/0007.

2. UTILITY ELICITATION

2.1 The algorithm design

To model a user's preference for multi-attribute negotiation, the widely used Multi-Attribute Utility Theory (MAUT) is adopted. The utility function U applied to a product $\mathbf{p}$ with n attributes is defined as $U(\mathbf{p}) = \sum_{i=1}^{n} w_i f_i(a_i)$. To elicit the user utility function from a complex solution space with flexibility to accept several types of user inputs, we propose a hybrid evolutionary approach, as illustrated in Figure 1. For each generation in the learning process, three operations are applied to the learning population. First, evolutionary operations, selection, crossover, and mutation are applied to the solution population for evolving the solutions. Second, the base solution is improved by integrating external knowledge into the solution population. The knowledge acquisition and integration method are detailed in Section 2.2. Third, each solution is optimized by local search, in this paper, the *Simulated Annealing* method is applied.

The solution chromosome $\mathbf{s}$ is encoded to carry information of the attribute weights and the attribute functions. In our experiment, attribute functions are assumed in the form $f_i(a_i) = a_i^{r_i}$. This allows us to encode the attribute functions by a single parameter $\mathbf{r}$ and provides variance of the functions. The fitness $g(\mathbf{s})$ of a solution $\mathbf{s}$ is a weighted sum of rating fitness $h(\mathbf{s})$ and ranking fitness $q(\mathbf{s})$. Usually the ranking information provide weak constraints [1]. In our experiments, we mainly consider the rating fitness, which can be calculated as $h(\mathbf{s}) = 1/\sum_{i=1}^{m} |u_i^{est} - u_i^{real}|$. Here m is the number of sample products; u_i^{est} and u_i^{real} are the estimated utility and user rating of the ith product. The user inputs of

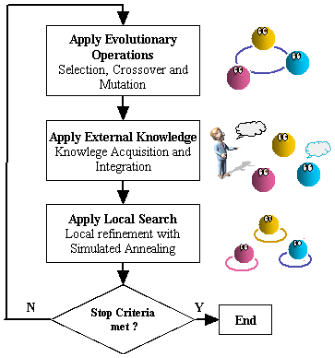
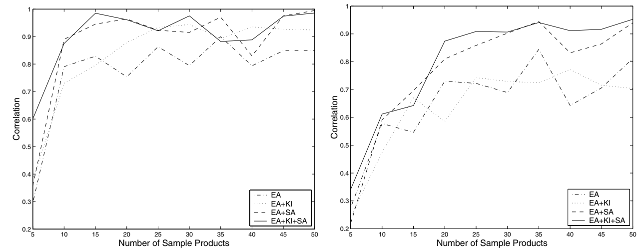


Figure 1: Utility elicitation algorithm



(a) $n = 5$ (b) $n = 10$

Figure 2: Learning result of attribute weight

product rating and ranking are applied in the fitness function. Direct knowledge about the solution is applied in the knowledge integration stage.

2.2 Applying knowledge

The knowledge to integrate into the learning population can result directly from user input or be derived from user observation. We use an information-theoretic approach [4] to estimate attribute weights directly from a user-rated product set, which is based on the mutual information $H(a_i; v)$ between the user rating v of a product and an attribute a_i of the product. $H(a_i; v)$ measures the amount of information the attribute conveys about the user valuation. In case attribute value or user valuation are continuous values, computing the mutual information between them requires to divide their ranges into k predefined intervals. The weight of attribute a_i can be calculated as $w_i^{mi} = \frac{H(v;a_i)}{\sum_{j=1}^{n} H(v;a_j)}$. Our experimental results reveal that this method is effective when the number of attribute is small, e.g. $n = 5$ or 10, whereas the estimation quality decreases with increasing number of attributes. Also, for a given number of attributes, the quality of estimation improves roughly at a linear scale as the number of rated products increase.

The knowledge integration method is designed to reflect the level of correctness of the knowledge and prevent assimilation of the solution populations. The major procedures are: First, the estimated attribute weights knowledge $\mathbf{w}^{mi}$ is applied to the solution population with probability of P_k. Then each selected solution replaces current attribute weights $\mathbf{w}^e$ with $\mathbf{w}^{mi}$, and the fitness of the new solution is evaluated. If the fitness is better, accept the knowledge that $w_i^k = w_i^e + Random(\beta, 1) * (w_i^{mi} - w_i^e)$, where β is the *acceptance rate*. Otherwise, the knowledge will be rejected. In our experiments, we used $P_k = 0.1$, and $\beta = 0.8$.

3. EMPIRICAL RESULTS

In our experiment, a virtual user is generated with his preference model in the form of a utility function. The sample products set will be randomly generated and rated with this utility function. Then the learning agents elicit the user preference model based on these rated products set. The experiments were performed with four learning algorithms on the same data set: pure evolution algorithm (EA), EA combined with application of external knowledge (EA+KI), EA combined with local search (EA+SA), and EA, KI, combined with SA (EA+KI+SA).

The learning quality of attribute weights is measured by the Bravais Pearson correlation coefficient between true and learned weights. As shown in Figure 2 on the results of attribute number $n = 5$ and 10, all the learning algorithms converge to the real attribute weights as the number of rated products increases, and the quality of the results is improved dramatically by applying external knowledge and by local search. This effect becomes stronger as the number of attributes increases. In addition, the positive effect of applying external knowledge depends very critically on the quality and correctness of that knowledge.

4. CONCLUSIONS AND FUTURE WORK

In this paper, a utility elicitation approach is proposed and implemented in an agent based negotiation system. The method combines an evolutionary learning with the application of external knowledge and local refinement. The algorithm learns a user's multi-attribute utility model with flexible user inputs. Empirical tests showed that the method can effectively elicit the attribute weights and the individual attribute utility functions.

A limitation of the work evaluated in the experiments is that only a single parameter has been used for modelling individual attribute functions. More flexible modelling will be studied either by using additional parameters or using genetic programming. In addition, symbolic attribute types will be considered by defining their domain ontologies to be accommodated in our current framework.

5. REFERENCES

[1] M. Bichler, J. Lee, C. H. Kim, and H. S. Lee. Design and implementation of an intelligent decision analysis system for e-sourcing. In *Proc. 2001 Int. Conf. on Artificial Intelligence (IC-AI'2001)*, pages 84–89, 2001.

[2] J. P. Müller. *The Design of Intelligent Agents. Lecture Notes of Artificial Intelligence, Vol. 1077.* Springer-Verlag, 1997.

[3] G. Tewari and P. Maes. Design and implementation of an agent-based intermediary infrastructure for electronic markets. In *Proc. 2nd ACM Conf. on Electronic commerce*, pages 86–94. ACM, 2000.

[4] D. Wettschereck, D. W. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11:273–314, July and Oct. 1997.