Learning Disjunctive Preferences for Negotiating Effectively *

(Extended Abstract)

Reyhan Aydoğan
Department of Computer Engineering
Boğaziçi University
Bebek, 34342, Istanbul,Turkey
reyhan.aydogan@gmail.com

Pınar Yolum
Department of Computer Engineering
Boğaziçi University
Bebek, 34342, Istanbul,Turkey
pinar.yolum@boun.edu.tr

ABSTRACT

Successful negotiation depends on understanding and responding to participants' needs. Many negotiation approaches assume identical needs (e.g., minimizing costs) and do not take into account other preferences of the participants. However, preferences play a crucial role in the outcome of negotiations. Accordingly, we propose a negotiation framework where producer agents learn the preferences of consumer preferences over time and negotiates based on this new knowledge. Our proposed approach is based on inductive learning but also incorporates the idea of revision. Thus, as the negotiation proceeds, a producer can revise its idea of the customer's preferences. This enables us to learn conjunctive as well as disjunctive preferences. Even if the consumer's preferences are specified in complex ways, such as conditional rules, our approach can learn and guide the producer to create well-targeted offers. Our experimental work shows that our proposed approach completes negotiation faster than similar approaches, especially if the producer will not be able to satisfy consumer's requests properly.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms; Experimentation

Keywords

Negotiation, Inductive Learning, Ontology, Semantic Similarity

1. INTRODUCTION

In automated service negotiation, a consumer agent that represents a real user and a provider agent that represents a business interact to reach a consensus about a service description. These agents interact by turn taking: Consumer starts the negotiation by requesting a service. If the producer cannot fulfill this need, it proposes a counter offer and so on.

*This research is supported by Boğaziçi University Research Fund under grant BAPA7A102 and Turkish Research and Technology Council CAREER Award under grant 105E073.

Cite as: Learning Disjunctive Preferences for Negotiating Effectively, (Extended Abstract), Reyhan Aydoğan, Pınar Yolum, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1201–1202

In order to negotiate successfully, participants need to consider each others' service preferences and generate offers accordingly. However, preferences of participants are almost always private and hence cannot be accessed by others. The best that can happen is that participants may learn about each others' preferences over time and through interactions. As agents learn about each others' preferences, they can provide better-targeted offers and thus enable faster negotiation.

During the learning process, consumer's requests are accepted as a positive example whereas the producer's counter-offers that are rejected by the consumer are taken as negative examples. According to the learning algorithm, producer's services are filtered so that the producer selects a service as an offer from a smaller set of possible services.

Learning consumer preferences during negotiation requires an incremental learning algorithm since the training data is gained during the negotiation process. For this purpose, we have previously extended the Candidate Elimination Algorithm (CEA) [2] to handle disjunctions [1]. In this work, we develop a different extension that benefits from service ontologies better and filters redundant offers earlier.

2. PROPOSED APPROACH

Our proposed approach can *retract* its hypothesis about what it has learned as more interactions take place. Further, it uses an underlying ontology of service features for revising hypothesis as necessary. Compared to existing approaches, our proposed approach can facilitate faster negotiation of service descriptions. If no consensus can be found, it signals this early in the negotiation.

First, according to CEA, all of the hypotheses in the most general set should cover the entire positive sample set. This rule prevents learning disjunctives since it is the union of more than one hypotheses and it cannot be covered by a single hypothesis with the condition of excluding negative samples. Therefore, in our learning algorithm we change this rule with that a positive sample should be covered by at least one of the hypotheses in the most general set. Furthermore, in some cases, a revision may be required when there is no more hypothesis in the most general set that is consistent with the incoming positive sample. In such a case, we need to add a new hypotheses covering this new positive sample while excluding all the negative samples. Hence, we require to keep the history of negative samples.

Second, when a positive sample comes, generalization of the specific set is performed in a controlled way with a threshold value, Θ. As far as disjunctive concepts are concerned, there should be more than one specific hypotheses in the most specific set. Thus, we estimate similarity of the current positive sample with respect to each hypothesis in the most specific set. The process of choosing the hypothesis that will be generalized uses this similarity information. Here, any similarity metric can be applied. During our study, we use the RP similarity [1], which uses semantic information such as subsumption relations. At the end, the approach only generalizes the selected hypothesis.

Third, different from original CEA, the generalization of a hypothesis is now controlled by another threshold value. This threshold value determines whether a generalization will be performed for each feature. If there is an ontological information such that a hierarchy exists on the values of the attribute, the generalization depends on the ratio of the covered branches in the hierarchical tree.

Our proposed approach consists of three important components:

Most specific set: It contains the most specific hypotheses that cover the positive examples. During the learning process, some hypotheses will be generalized to handle new positive samples. Since we plan to learn also disjunctives, there is no enforcement to make all specific hypotheses cover all positive samples. Instead, at least one specific hypothesis in the most specific set should cover the new positive example. Thus, when a positive sample comes, the similarity of the specific hypotheses is calculated and the most similar hypothesis to the current positive sample is found. Deciding whether the approach will generalize the most similar hypothesis to cover the current sample or add the positive sample as a separate hypothesis depends on the threshold value. In our implementation threshold value can be set as a floating number between zero and one. The similarity of the most similar hypothesis should be greater than the threshold value to generalize the hypothesis. When the approach generalizes a specific hypothesis, it will check to ensure that negative samples are not covered in this more general hypothesis.

Most general set: This set includes the most general hypotheses consistent with the positive samples. According to the original version space algorithm, the hypotheses covering the incoming negative examples are specialized such that they do not cover the negative sample any more. This rule is also valid for our proposed algorithm but there is no obligation for all hypotheses in the most general set to cover all positive samples. Since our learning algorithm allows learning disjunctives, there may be a variety of general hypotheses consistent with different positive samples. Therefore, our algorithm does not eliminate any general hypotheses not covering the current positive sample. Of course, there should be at least one hypothesis consistent with the current positive examples. In some cases, the current general set does not have any hypothesis consistent with the incoming positive examples. In such a case, our algorithm makes some revisions in which new general hypotheses are added covering the current positive example but not covering any negative examples in the negative sample set.

Negative Sample Set: It contains the collection of negative examples.

For each training sample, which is given as an input to the system, both most general and most specific set are modified. Modifications depend on the type (positive or negative) of the sample.

If the sample is positive, the algorithm updates the specific set in a way that current positive example is also covered in a specific hypothesis, but none of the negative samples are covered. In detail, for each specific hypothesis, a similarity is estimated and the hypotheses whose similarity with the current sample is greater than Θ are sorted according to their similarity value in a descending order. Starting with the first hypothesis in this list, we generalize the specific hypothesis to cover the new example. Afterward, the algorithm tests whether the extended specific hypothesis covers any negative example in the negative sample set. If this hypothesis covers a negative example, the algorithm interprets that this extension is invalid so it passes to the next specific hypothesis in the ordered list to generalize in order to cover the new positive sample. This process continues until the extended specific hypothesis does not cover any negative examples or the list is exhausted. If a hypothesis with a valid extension is found, original hypothesis is replaced with its extended version. On the other hand, if the list is exhausted or there is no hypothesis whose similarity is greater than Θ, the new positive example is added as a separate hypothesis into most specific set.

As mentioned before, the positive sample should be covered by at least one hypothesis in the general set. If none of the hypotheses cover the new positive sample, new general hypotheses covering this positive sample but not covering any negative samples are added into most general set. This revision process is a new operation for Version Space. By this operation, our algorithm supports learning disjunctive concepts.

If the sample is negative, it is first added to the negative sample set. Then, the hypotheses in the most general set is minimally specialized. Minimal specialization is crucial since it is desired that the hypothesis remains as general as possible. Therefore, the algorithm checks all the hypotheses in the most general set to see if they cover the negative sample. Each hypothesis covering the negative sample is removed from the most general set. However, the information kept in those hypothesis should not be lost. Therefore, new hypotheses are generated from the abandoned hypotheses. By only taking the hypotheses in the most specific set that are covered by the abandoned hypotheses into account, a minimal specification of the abandoned hypotheses that do not cover the negative sample are generated. To accomplish this, the algorithm compares the value of a feature of specific hypothesis with that of the negative sample. If the values are different, the value of feature of specific hypothesis is added into the abandoned hypothesis.

Next, if there are some hypotheses in most general set less general than the other hypotheses in this set, they are removed. Finally, the most specific set is checked for whether they cover the negative sample. If they do, these hypotheses covering the negative sample are removed.

3. RESULTS AND DIRECTIONS

We have implemented our algorithm and studied its performance in learning preferences. Our main observations are that our proposed algorithm can filter unnecessary offers quickly and thus speeds up the negotiation process considerably. Further, if none of the services offered by the provider can satisfy the consumer, our algorithm successfully signals a failure before trying all possible alternatives.

In this work, we assumed that the producer knows the preference model of the consumer. However, in many settings, a producer may not be aware of the preference model. As a future work, we plan to improve our learning algorithm so that it can learn the preferences of others even when their model is not known.

4. REFERENCES

[1] R. Aydoğan and P. Yolum. Learning consumer preferences using semantic similarity. In *Proceedings of Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1293–1300, May 2007.

[2] T. M. Mitchell. *Machine Learning*. McGraw Hill, New York, 1997.