

Learning the Structure of Utility Graphs Used in Multi-Issue Negotiation through Collaborative Filtering

Preliminary version*

Valentin Robu, Han La Poutré

CWI, Dutch National Research Center for Mathematics and Computer Science
Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands
{robu, hlp}@cwi.nl

Abstract. Graphical utility models represent powerful formalisms for modeling complex agent decisions involving multiple issues [2]. In the context of negotiation, it has been shown [10] that using utility graphs enables reaching Pareto-efficient agreements with a limited number of negotiation steps, even for high-dimensional negotiations involving complex complementarity/ substitutability dependencies between multiple issues. This paper considerably extends the results of [10], by proposing a method for constructing the utility graphs of buyers automatically, based on previous negotiation data. Our method is based on techniques inspired from item-based collaborative filtering, used in online recommendation algorithms. Experimental results show that our approach is able to retrieve the structure of utility graphs online, with a high degree of accuracy, even for highly non-linear settings and even if a relatively small amount of data about concluded negotiations is available.

1 Introduction

Negotiation represents a key form of interaction between providers and consumers in electronic markets. One of the main benefits of negotiation in e-commerce is that it enables greater customization to individual customer preferences, and it supports buyer decisions in settings which require agreements over complex contracts. Automating the negotiation process, through the use of intelligent agents which negotiate on behalf of their owners, enables electronic merchants to go beyond price competition by providing flexible contracts, tailored to the needs of individual buyers.

Multi-issue (or multi-item) negotiation models are particularly useful for this task, since with multi-issue negotiations mutually beneficial (“win-win”) contracts can be found [4, 13, 5, 8, 7]. However, most existing approaches to automated negotiations only

* This is a preliminary version of this work, as it resulted from a presentation at the PRIMA’05 workshop in September 2005. At the time of the publication of these post-proceedings (2009), however, a more definitive version of this work has already appeared as a book chapter in “Rational, Robust, and Secure Negotiations in Multi-Agent Systems”, Ito, T.; Hattori, H.; Zhang, M.; Matsuo, T. (Eds.), Studies in Computational Intelligence Series, vol. 89., Springer-Verlag, 2008. Interested readers may consult either version.

deal with linearly additive utility functions, and do not consider high-dimensional negotiations and in particular, the problem of interdependencies between evaluations for different issues or items. This is a significant problem, since identifying and exploiting substitutability/complementarity effects between different items can be crucial in reaching mutually profitable deals.

In our previous work (to appear as full paper at the AAMAS 2005 conference - [10]), we have introduced the concept of utility graphs. Intuitively defined, a utility graph (UG) is a structural model of a buyer, representing a buyer's perception of dependencies between two items (i.e. whether the buyer perceives two items to be as complementary or substitutable). An estimation of the buyer's utility graph can be used by the seller to efficiently compute the buyer's utility for a "bundle" of items, and propose a bundle and price based on this utility. The main result presented in [10] is that Pareto-efficient agreements can be reached, even for high dimensional negotiations with a limited number of negotiation steps, but provided that the seller starts the negotiation with a reasonable approximation of the *structure* of the true utility graph of the type of buyer he is negotiating with (i.e. he has a reasonable idea which issues or items may be complementary or substitutable in the evaluation of buyers in his domain).

The seller agent can then use this graph to negotiate with a specific buyer. During this negotiation, the seller will adapt the weights and potentials in the graph, based on the buyer's past bids. However, this assumes the seller knows a super-graph of the utility graphs of the class of buyers he is negotiating with (i.e. a graph which subsumes the types of dependencies likely to be encountered in a given domain - c.f. Sec. 2.2).

An important issue left open in [10] is how does the seller acquire this initial graph information. One method would be to elicit it from human experts (i.e. an e-commerce merchant is likely to know which items are usually sold together or complimentary in value for the average buyer and which items are not). For example, if the electronic merchant is selling pay-per-item music tunes, the tunes from the same composer or performer can be potentially related.

In this paper, we show this can also be retrieved automatically, by using information from completed negotiations data. The implicit assumption we use here is that buyer preferences are in some way clustered, i.e. by looking at buyers that have shown interest for the same combinations of items in the past, we can make a prediction about future buying patterns of the current customer. Note that this assumption is not uncommon: it is a building block of most recommendation mechanisms deployed in Internet today [12]. In order to generate this initial structure of our utility graph, in this paper we propose a technique inspired by collaborative filtering.

1.1 Collaborative filtering

Collaborative filtering [12] is the main underlying technique used to enable personalization and buyer decision aid in today's e-commerce, and has proven very successful both in research and practice.

The main idea of collaborative filtering is to output recommendations to buyers, based on the buying patterns detected from buyers in previous buy instances. There are two approaches to this problem. The first of these is use of the preference database to discover, for each buyer, a neighborhood of other buyers who, historically, had similar

preferences to the current one. This method has the disadvantage that it requires storing a lot of personalized information and is not scalable (see [12]). The second method, of more relevance to our approach, is **item-based collaborative filtering**. Item based techniques first analyze the user-item matrix (i.e. a matrix which relates the users to the items they have expressed interest in buying), in order to identify relationships between different items, and then use these to compute recommendations to the users [12].

In our case, of course, the recommendation step is completely replaced by negotiation. What negotiation can add to such techniques is that enables a much higher degree of customization, also taking into account the preferences of a specific customer. For example, a customer expressing an interest to buy a book on Amazon is sometimes offered a "special deal" discount on a set (bundle) of books, including the one he initially asked for. The potential problem with such a recommendation mechanism is that it's static: the customer can only take it, leave it or stick to his initial buy, it cannot change slightly the content of the suggested bundle or try to negotiate a better discount. By using negotiation a greater degree of flexibility is possible, because the customer can critique the merchant's sub-optimal offers through her own counter-offers, so the space of mutually profitable deals can be better explored.

1.2 Paper structure and relationship to previous work

The paper is organized as follows. In Section 2 we briefly present the general setting of our negotiation problem, define the utility graph formalism and the way it can be used in negotiations. Section 3 describes the main result of this paper, namely how the structure of utility graphs can be elicited from existing negotiation data. Section 4 presents the experimental results from our model, while Section 5 concludes with a discussion.

An issue which is important to define is the relationship of this paper with our previous work [10]. The two papers are related, since they present solutions to complementary problems. For lack of space, we cannot present the model from [10] in this paper, except at a very general level, since we prefer to concentrate on describing the new results. However, the interested reader is asked to consult [10] for further details.

2 The multi-issue negotiation setting

In this section we give some background information of the set-up of our model. First we give a formal definition of the concept of utility graphs. Next we describe (very briefly) how this formalism can be used in negotiation (a issue fully discussed in [10]). Finally we discuss how the learning of the structure from past data is integrated with the negotiation part.

2.1 Utility Graphs: Definition and Example

We consider the problem of a buyer who negotiates with a seller over a bundle of n items, denoted by $B = \{I_1, \dots, I_n\}$. Each item I_i takes on either the value 0 or 1: 1 (0) means that the item is (not) purchased. The utility function $u : Dom(B) \mapsto \mathbb{R}$ specifies the monetary value a buyer assigns to the 2^n possible bundles ($Dom(B) = \{0, 1\}^n$).

In traditional multi-attribute utility theory, u would be decomposable as the sum of utilities over the individual issues (items) [9]. However, in this paper we follow the previous work of [2] by relaxing this assumption; they consider the case where u is decomposable in *sub-clusters* of individual items such that u is equal to the sum of the *sub-utilities* of different clusters.

Definition 1. Let C be a set of (not necessarily disjoint) clusters of items C_1, \dots, C_r (with $C_i \subseteq B$). We say that a utility function is factored according to C if there exists functions $u_i : \text{Dom}(C_i) \mapsto \mathbb{R}$ ($i = 1, \dots, r$ and $\text{Dom}(C_i) = \{0, 1\}^{|C_i|}$) such that $u(\mathbf{b}) = \sum_i u_i(\mathbf{c}_i)$ where \mathbf{b} is the assignment to the variables in B and \mathbf{c}_i is the corresponding assignment to variables in C_i . We call the functions u_i sub-utility functions.

We use the following factorization, which is a relatively natural choice within the context of negotiation. Single-item clusters ($|C_i| = 1$) represent the individual value of purchasing an item, regardless of whether other items are present in the same bundle. Clusters with more than one element ($|C_i| > 1$) represent the *synergy effect* of buying two or more items; these synergy effects are positive for complementary items and negative for substitutable ones. In this paper, we restrict our attention to clusters of size 1 and 2 ($|C_i| \in \{1, 2\}, \forall i$). This means we only consider binary item-item complementarity/substitutability relationships, though the case of retrieving larger clusters could form the object of future research. The factorization defined above can be represented as an undirected graph $G = (V, E)$, where the vertexes V represent the set of items I under negotiation. An arc between two vertexes (items) $i, j \in V$ is present in this graph if and only if there is some cluster C_k that contains both I_i and I_j . We will henceforth call such a graph G a *utility graph*.

Example 1. Let $B = \{I_1, I_2, I_3, I_4\}$ and $C = \{\{I_1\}, \{I_2\}, \{I_1, I_2\}, \{I_2, I_3\}, \{I_2, I_4\}\}$ such that u_i is the sub-utility function associated with cluster i ($i = 1, \dots, 5$). Then the utility of purchasing, for instance, items I_1, I_2 , and I_3 (i.e., $\mathbf{b} = (1, 1, 1, 0)$) can be computed as follows: $u((1, 1, 1, 0)) = u_1(1) + u_2(1) + u_3((1, 1)) + u_4((1, 1))$, where we use the fact that $u_5(1, 0) = u_5(0, 1) = 0$ (synergy effect only occur when two or more items are purchased). The utility graph of this factorization is depicted in Fig. 1.

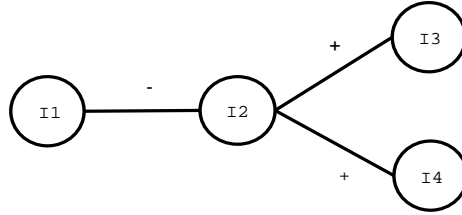


Fig. 1. The utility graph that corresponds to the factorization according to C in Example 1. The + and - signs on the edges indicate whether the synergy represents a complementarity, respectively substitutability effect.

2.2 Minimal super-graph for a class of buyers

The definition of utility graphs given in Section 2.1 corresponds to the modeling the utility function of an individual buyer. In this paper, we call the utility graph of an individual buyer the *underlying* or *true* graph (to distinguish it from the *retrieved* or *learned* graph, reconstructed through our method). The underlying graph of any buyer remains hidden from the seller throughout the negotiation.

We do assume, however, that the buyers which negotiate with a given electronic merchant belong to a certain class or population of buyers. This means the utility buyers assign to different bundles of items follow a certain structure, specific to a buying domain (an assumption also used indirectly in [13, 4, 12]). Buyers from the same population are expected to have largely overlapping graphs, though not all buyers will have all interdependencies specific to the class.

Definition 2. Let $A = \{A_1, \dots, A_n\}$ be a set (class, population) of n buyers. Each buyer $i = 1..n$ has a utility function u_i , which can be factored according to a set of clusters $C_i = \{C_{i,1}, C_{i,2}, \dots, C_{i,r(i)}\}$. We define the super-set of clusters for the class of buyers $A = \{A_1, \dots, A_n\}$ as: $C_A = C_1 \cup C_2 \cup \dots \cup C_n$.

In graph-theoretic terms (as shown in Section 2.1), the set of clusters C_i according to which the utility a buyer A_i is structured is represented by a utility graph G_i , where each binary cluster from $\{C_{i,1}, \dots, C_{i,r(i)}\}$ represents a dependency or an edge in the graph. The super-set of buyer clusters C_A can also be represented by a graph G_A , which is the **minimal super-graph** of graphs G_i , $i = 1..n$. This graph is called **minimal** because it contains no other edges than those corresponding to a dependency in the graph of at least one buyer agent from this class. We illustrate this concept by a very simple example, which also relies on Fig. 1.

Example 2. Suppose we have 2 buyer agents A_1 and A_2 (obviously, this is a simplification, since a class would normally contain many more buyer graphs). Suppose the utility function of buyer A_1 can be factored according to the clusters $C_1 = \{\{I_1\}, \{I_2\}, \{I_2, I_3\}, \{I_2, I_4\}\}$, while the utility of A_2 is factored according to $C_2 = \{\{I_1, I_2\}, \{I_2, I_3\}, \{I_3\}\}$. Then the minimal utility super-graph for class A is given by: $C_A = \{\{I_1\}, \{I_2\}, \{I_3\}, \{I_1, I_2\}, \{I_2, I_3\}, \{I_2, I_4\}\}$. This super-graph is minimal, because if we were to add the dependency $\{I_1, I_3\}$ to C_A we would also obtain a super-graph, though not the minimal one.

It is important to note that the above definition for the utility super-graph for a class of buyer refers only to the **structure** (i.e. clusters C_i) and makes no assumption about the sub-utility **values** (i.e. functions u_i) in these clusters. To illustrate the difference, suppose that at a structural level, there is a complementarity effect between two items. However, for some buyers in the population, the utility value corresponding to this dependency may be very high (i.e. it is important for the agent to get both items), while for others it is much more moderate (or even close to zero).

2.3 Application to negotiation

The negotiation, in our model, follows an alternating offers protocol. At each negotiation step each party (buyer/seller) makes an offer which contains an instantiation with

0/1 for all items in the negotiation set (denoting whether they are/are not included in the proposed bundle), as well as a price for that bundle. The decision process of the seller agent, at each negotiation step, is composed of 3 inter-related parts: (1) take into account the previous offer made by the other party, by updating his estimated utility graph of the preferences of the other party, (2) compute the contents (i.e. item configuration) of the next bundle to be proposed, and (3) compute the price to be proposed for this bundle.

An important part of our model is that the burden of exploring the exponentially large bundle space and recommending profitable solutions is passed to the seller, who must solve it by modeling the preferences of his buyer (this is a reasonable model in e-commerce domains, where electronic merchants typically are more knowledgeable than individual buyers [4, 13]). The model the seller maintains of his buyer is represented by a utility graph, and tailors this graph towards the preferences of a given buyer, based on his/her previous offers.

The seller does not know, at any stage, the values in the actual utility graph of the buyer, he only has an approximation learned after a number of negotiation steps. However, the seller does have some prior information to guide his opponent modeling. He starts the negotiation by knowing a *super-graph of possible inter-dependencies* between the issues (items) which can be present for the class of buyers he may encounter. The utility graphs of buyers form subgraphs of this graph. Note that this assumption says nothing about values of the sub-utility functions, so the negotiation is still with double-sided incomplete information (i.e. neither party has full information about the preferences of the other).

In [10] we show how the presence of this graph helps to greatly reduce the complexity of the search space on the side of the seller. In [10] we argued that the structure of the minimal super-graph of the class of buyers likely to be encountered during negotiations can be obtained either from human experts or automatically, from a history of past negotiations, but in [10] we proposed no concrete mechanism how can this be achieved. It is this open problem that forms the subject of this paper.

2.4 Overview of our approach

There are two main stages of our approach(see also Figure 2):

1. Using information from previously concluded negotiations to construct the structure of the utility super-graph. In this phase the information used (past negotiation data) refers to a class of buyers and is not traceable to individuals.
2. The actual negotiation, in which the seller, starting from a super-graph for a class (population) of buyers, will negotiate with an individual buyer, drawn at random from the buyer population above. In this case, learning occurs based on the buyer's previous bids during the negotiation, so information is buyer-specific. However, this learning at this stage is guided by the structure of the super-graph extracted in the first phase.

Phase 2 is described in our previous work [10]. The rest of this paper will focus on describing the first phase of our model, namely retrieving the structure of the utility super-graph from previous data.

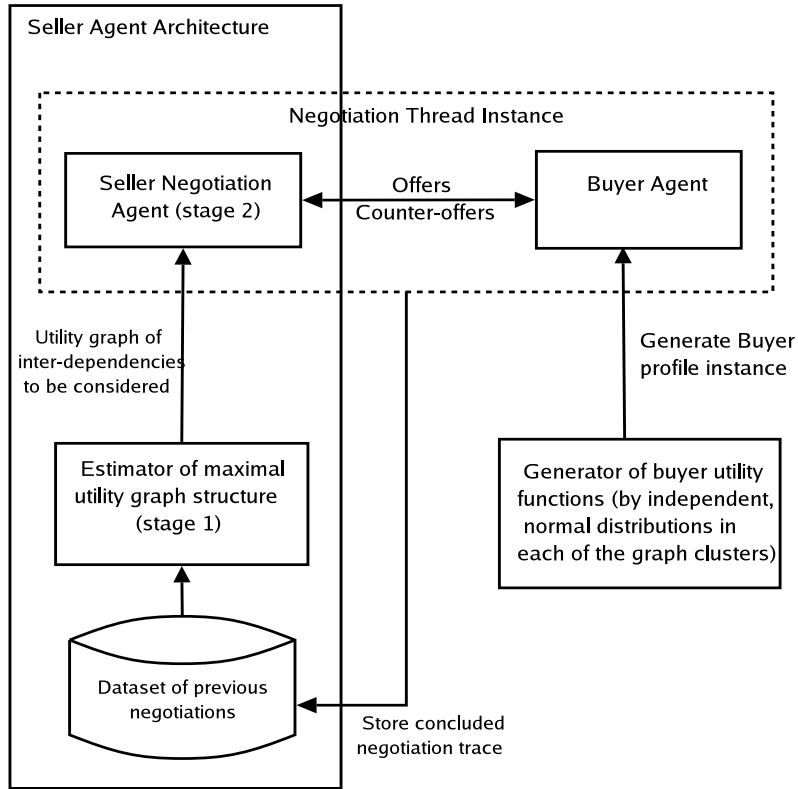


Fig. 2. Top-level view of our agent architecture and simulation model

3 Constructing the Structure of Utility Graphs Using Concluded Negotiation Data

Suppose the seller starts by having a dataset with information about previous concluded negotiations. This dataset may contain complete negotiation traces for different buyers, or we may choose, in order to minimize bias due to uneven-length negotiations, to consider only one record per negotiation. This can be either the first bid of the buyer or the bundle representing the outcome of the negotiation (for details regarding how this negotiation data is generated and buyer profiles for the simulated negotiations are generated, please see the experimental set-up description in Section 4).

The considered dataset is not personalised, i.e. the data which is collected online cannot be traced back to individual customers (this is a reasonable assumption in e-commerce, where storing a large amount of personalised information may harm customer privacy). However, in constructing of the minimal utility graph which the customers use, we implicitly assume that customers' preference functions are related - i.e. their corresponding utility graphs, have a (partially) overlapping structure.

Our goal is to retrieve the **minimal super-graph** of utility interdependences which can be present for the class or population of buyers from which the negotiation data was generated.

We assume that past data can be represented as a $N * n$ matrix, where N is the number of previous negotiation instances considered (e.g. up to 3000 in the tests reported in this paper) and n is the number of issues (e.g. 50 for our tests). All the data is binary (i.e. with values of "1" in the case the buyer asked for this item or "0" if he does not). Item-based collaborative filtering [12] works by computing "similarity measures" between all pairs of items in the negotiation set. The steps used are:

1. Compute raw item-item statistics (i.e. from existing negotiation data)
2. Compute item-item similarity matrices (from the raw statistics)
3. Compute qualitative utility graph, by selecting which dependencies to consider from the similarity matrices.

In the following, we will examine each of these separately.

3.1 Computing the "raw" statistic matrices

Since what we need to compute is item-item similarity measures, we extract from this data some much smaller ($n*n$ tables) which are sufficient to compute the required measures. We use the following notations throughout this paper:

- N for the total number of previous negotiation outcomes considered
- For each item $i = 1..n$ $N_i(1)$ and $N_i(0)$ represent the number of times the item was (respectively was not) asked by the buyer, from the total of N previous negotiations
- For each pair of issues $i, j = 1..n$ we denote by $N_{i,j}(0,0)$, $N_{i,j}(0,1)$, $N_{i,j}(1,0)$ and $N_{i,j}(1,1)$ all possibilities of joint acquisition (or non acquisition) of items i and j .

From the above definitions, the following property results immediately: $N_{i,j}(0,0) + N_{i,j}(0,1) + N_{i,j}(1,0) + N_{i,j}(1,1) = N_i(0) + N_i(1) = N_j(0) + N_j(1) = N$, for all items $i, j = 1..n$.

3.2 Computing the similarity matrices

The literature on item-based collaborative filtering defines two main criteria that could be used to compute the similarity between pairs of items. We discuss them in separate sub-sections. Since their mathematical definitions (as presented in [12]) are for real-valued preference ratings, we needed to derive the appropriate form for the binary values case. In the following we present just the resulting formulas, and abstract away from how they were deduced from the model presented in [12].

Cosine-based similarity For cosine based similarity, two separate matrices are computed: one for likely complementarity between a pair of items and one for likely substitutability. Both of these have positive entries (between 0 and 1), a higher number denoting a stronger potential similarity.

The formula to compute the entries in the complementarity matrix is:

$$Sim_{compl}(i, j) = \frac{N_{i,j}(1, 1)}{\sqrt{N_i(1) * N_j(1)}} \quad (1)$$

The formula for entries in the substitutability matrix is:

$$Sim_{subst}(i, j) = \frac{N_{i,j}(0, 1) + N_{i,j}(1, 0)}{\sqrt{N_i(1) * N_j(1)}} \quad (2)$$

Correlation-based similarity For correlation-based similarity, just one similarity matrix is computed containing both positive and negative values (to be more precise between -1 and 1). We first we define for each item $i = 1..n$, the average buy rate:

$$Av_i = \frac{N_i(1)}{N} \quad (3)$$

The following two terms are defined:

$$\begin{aligned} \psi_1 = & N_{i,j}(0, 0) * Av_i * Av_j - N_{i,j}(0, 1) * Av_i * (1 - Av_j) \\ & - N_{i,j}(1, 0) * (1 - Av_i) * Av_j + N_{i,j}(1, 1) * (1 - Av_i) * (1 - Av_j) \end{aligned}$$

and the normalization factor:

$$\psi_2 = \sqrt{\frac{N_i(0) * N_i(1)}{N}} * \sqrt{\frac{N_j(0) * N_j(1)}{N}}$$

The values in the correlation-based similarity matrix are then computed as:

$$Sim(i, j) = \frac{\psi_1}{\psi_2} \quad (4)$$

3.3 Building the super-graph of buyer utilities

After constructing the similarity matrices, the next step is to use this information to build the utility super-graph for the class of buyers likely to be encountered in future negotiations. This amounts to deciding which of the item-item relationships from the similarity matrixes should be included in this graph. For both similarity measures, higher values (i.e. closer to 1) represent stronger potential complementarity. For substitutability detection, the cosine similarity uses a different matrix, while the correlation-based it is enough to select values closer to -1.

Ideally, all the inter-dependencies corresponding to the arcs in the graph representing the true underlying preferences of the buyer should feature among the highest (respectively the lowest) values in the retrieved correlation tables. When an interdependency is returned that was not actually in the true graph, we call this is spurious (or erroneous) arc or interdependency. Due to noise in the data, it is unavoidable that a number of such spurious arcs are returned. For example, if item I_1 has a complimentary value with I_2 and I_2 is substitutable with I_3 , it may be that items I_1 and I_3 often do not

appear together, so the algorithm detects a substitutability relationship between them, which is in fact spurious.

The question on the part of the seller is: how many dependencies should be considered, given that the seller does not know the number of dependencies which should be actually present in the minimal super-graph which it needs to retrieve? Setting a value threshold for the strength of interdependencies to be considered has so far proved error-prone, since it may include too many or too few dependencies. If we choose too few dependencies, there is a high probability that dependencies actually present in the utility graph may be missed. If we choose too many, then surely a large number of spurious (i.e. dependencies not in the real graph of any buyer) will be included. Both of these cases can negatively affect the Pareto-efficiency and negotiation speed in the negotiation stage (stage 2 of our set-up), though missing dependencies can typically have a larger negative effect.

One way to tackle this problem is to set a formal limit on the maximum complexity (i.e. density) of the class of utility graphs which can be handled by our approach. More precisely stated, we restrict our attention to graphs where the number of edges (i.e. dependencies) is a linear function of the number of vertexes (i.e. items in the negotiation set). Formally, if n is the number of items under negotiation, then we assume the number of dependencies in the utility graph of the buyer should not exceed a factor of $k_{max} * n$, where k_{max} is a small positive number.

This restriction is reasonable considering our application domain, since we are modeling preferences of human buyers in e-commerce scenarios. From a human cognition perspective, it is reasonable to assume that the number of utility inter-dependencies (i.e. complementarity and substitutability effects) between a set of items is linear in the number of items and not exponential, for most realistic products (e.g. books, music tunes etc.) sold on the internet. From a graph-theoretic perspective, this restriction also makes sense, since our negotiation algorithm (reported in [10]) explicitly relies on the assumption that utility graphs used in the negotiation have a tree-width of maximum 2 (which means that their maximum number of edges is restricted to $2 * n$). We stress, however, that we make no assumption on number of issues which can be handled in our negotiation set, only on the maximum degree of interdependencies between these issues (density of the utility graph). Further work will focus, using results from random graph theory [1], on better identifying and defining cut-off values of k_{max} for different types of graph structures complexity.

For the experimental results presented in this paper, in building the minimal super-graph we approximate $k_{max} = 1.5$, which means we restrict our attention to graphs that have a number of edges relatively close to the number of vertexes. We found that this ratio provides a relatively good balance between including too many spurious dependencies and the chance of missing some of them.

4 Experimental results

The model above was tested for a setting involving 50 binary-valued issues (items). Two sets of tests were performed: one for the cosine-based similarity criteria, one for

the correlation-based similarity. There are two dimensions across which the two criteria need to be tested:

- **The strength of the interdependencies in the generated buyer profiles.** This is measured as a ratio of the average strength of the inter-dependency over the average utilities of an individual item. To explain, each buyer profile is generated as follows: First, for each item, an individual value is generated by drawing from identical, independent normal distributions (i.i.d.) of center $C_{individual-item} = 1$ and variance 0.5. Next, the substitutability/complementarity effects for each binary issue dependency (i.e. each cluster containing two items) are generated by drawing from a normal i.i.d-s with a centers $C_{non-linearity}$ and the same spread 0.5. The strength of the interdependency is then taken to be $\frac{C_{non-linearity}}{C_{individual-item}}$. The smaller this ratio is, the more difficult it will be to detect non-linearity (i.e. complementarity and substitutability effects between items). In fact, if this ratio takes the value 0, there are no effects to detect (which explains the performance at this point), at 0.1 the effects are very weak, but they become stronger as it approaches 1 and 2.
- **Number of previous negotiations from which information (i.e. negotiation trace) is available.**

The performance measure used is computed as follows. Each run of an algorithm (for a given history of negotiations, and a certain probability distribution for generating that history) returns an estimation of the utility graph of the buyer. Our performance measure is the recall, i.e. the percentage of the dependencies from the underlying utility graph of the buyer (from which buyer profiles are generated) which are found in the graph retrieved by the seller. Due to noise and/or insufficient data, we cannot expect this graph retrieval process to always have 100% accuracy. We therefore studied what is the effect of an imprecise graph on the part of the seller on the negotiation process itself (stage 2 of our approach). This is discussed in Section 4.2.

The setting presented above was tested for both cosine-based and correlation based similarity. Figure 3 gives the resulting graphs for the cosine-based case, while Fig. 4 gives the results for the correlation-based one. Each of the points plotted and resulting dispersions was computed by averaging over 50 different tests. In all these tests, to make them independent as possible, a new data set was generated. For each set of tests, the structure of the graph was also generated at random, by selecting at random the items (vertexes) connected by each edge representing a utility inter-dependency. For 50 issues, 75 random binary dependencies were generated for each test set, 50 of which were positive dependencies and 25 negative ones.

4.1 Results for cosine-based similarity vs. correlation based similarity: an interpretation of results

The overall conclusion which can be drawn from our tests (see Fig. 3 and Fig. 4) is that one of the techniques we investigated, namely correlation-based similarity is considerably more successively than the simpler, cosine-based similarity technique. This can be easily observed from Fig. 3 and Fig. 4: while correlation-based similarity can extract 96% (+/- 7%) of dependencies correctly given enough data (from around 1500

completed negotiations) and strong enough dependency effects (above 1), cosine-based similarity achieves a maximum of just above 40%. Thus, we conclude that correlation-based similarity is the most suitable one for the problem of learning the structure of utility graphs, from the item-based collaborative filtering techniques mentioned in existing literature we are aware of.

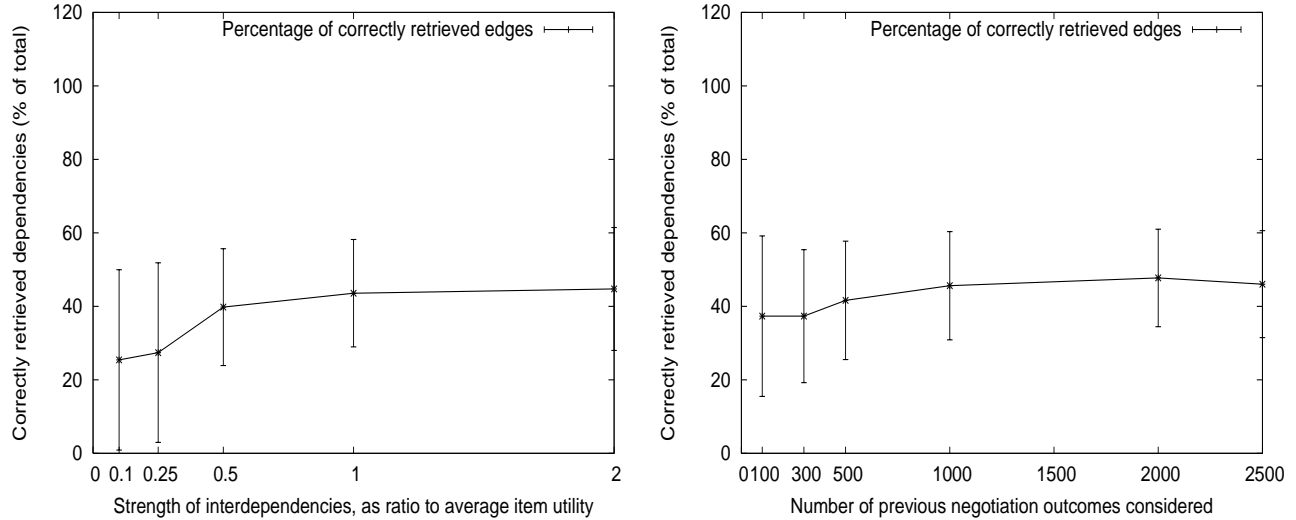


Fig. 3. Results for the cosine-based similarity. Left-side graph gives the percentage of correctly retrieved dependencies, with respect to the average interdependency strength, while right-side graph gives the percentage of correctly retrieved dependencies with respect to the size of the available dataset of past negotiation traces.

4.2 Effect of errors in the retrieved utility graph on the negotiation process

As we have shown in Sec. 2.3, in order for the second phase of our approach to reach efficient outcomes, it is necessary that the seller starts the negotiation with the minimal super-graph of utility interdependencies for the class of buyers he will encounter in negotiations. However, as shown in Sec. 4.1, it is not always possible to retrieve the structure of this graph with 100% accuracy, if inter-dependencies to be detected are not strong enough or if insufficient data is available from previous negotiations. Therefore, in further tests, we investigated the effect of an inaccurate (partially incorrect) graph on the performance of the negotiation algorithm itself (stage two of our model [10]). The results from these tests are shown in Fig. 5.

Tests from Fig. 5 show that our negotiation algorithm exhibits a ‘graceful degradation’ property. This means that it is not necessary to have 100% accuracy (in terms of correctly retrieved edges), since a certain degree of error (under 10%) would not degrade the Pareto-efficiency of the outcomes very significantly, though it may lead

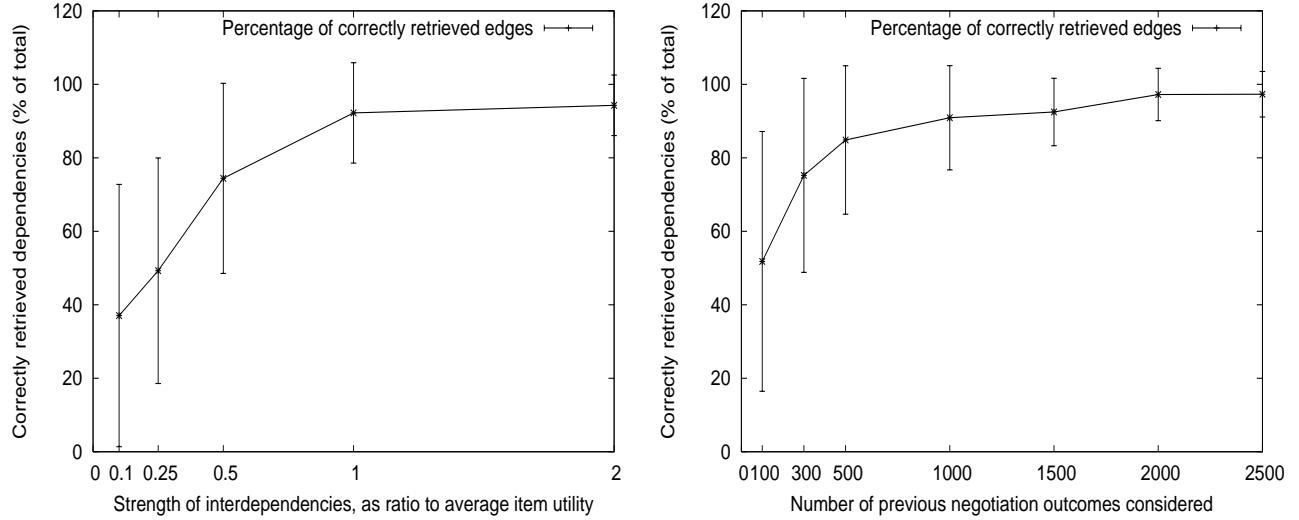


Fig. 4. Results for the correlation-based similarity. Left-side graph gives the percentage of correctly retrieved dependencies, with respect to the average interdependency strength, while right-side graph gives the percentage of correctly retrieved dependencies with respect to the size of the available dataset of past negotiation traces.

to slightly longer negotiations, in terms of the number of steps needed to reach these outcomes.

5 Discussion

In this section we provide a review of related work, with special attention to the features relevant for our approach. We conclude by summarizing the main contributions of our work and identifying directions for future research.

Several previous results model automated negotiation as a tool for supporting the buyer's decision process in complex e-commerce domains [13, 4, 5, 3]. Most of the work in multi-issue negotiations has focused on the independent valuations case. Faratin, Sierra & Jennings [8] introduce a method to search the utility space over multiple attributes, which uses fuzzy similarity criteria between attribute value labels as prior information. Coehoorn and Jennings [7] extends this model with a method to learn the preference weights that the opponent assigns to different issues in the negotiation set, by using kernel density estimation. These papers have the advantage that they allow flexibility in modeling and deal with incomplete preference information supplied by the negotiation partner. They do not consider the question of functional interdependencies between issues, however.

Other approaches to multi-issue negotiation problem are the agenda based approach (Fatima et. al. [11]) and the constraint-based negotiation approach (Luo et. al. [14]). Debenham [3] proposes a multi-issue bargaining strategy that models the iterative in-

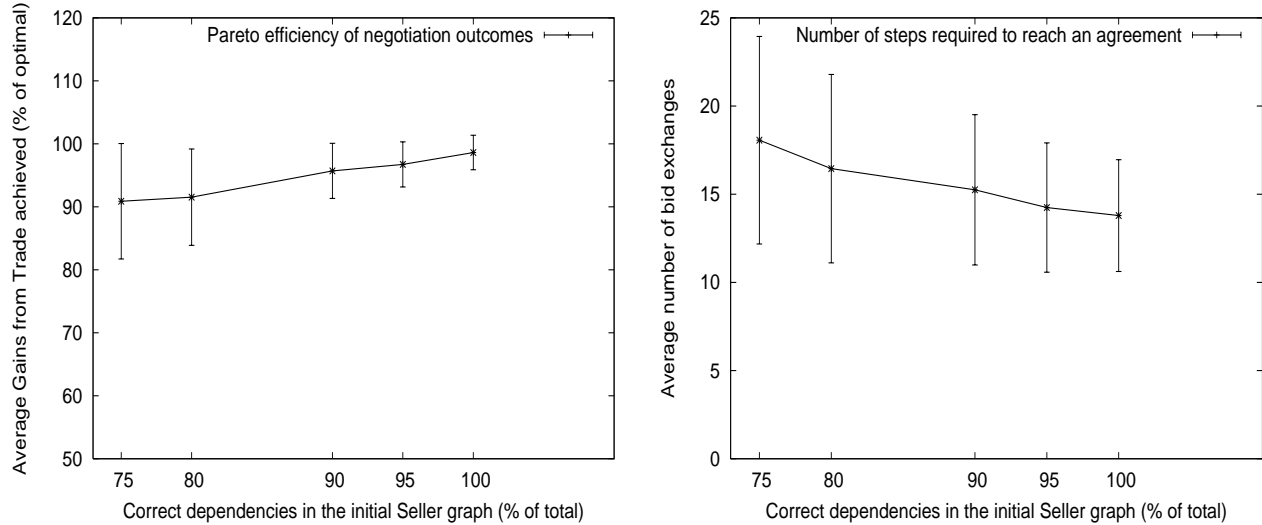


Fig. 5. Effect of errors (i.e. a number of incorrectly retrieved dependencies in the super-graph used by the seller) on the negotiation process

formation gathering which takes place during the negotiation. The agents in [3] do not explicitly model the preferences of their opponent, but construct a probability distribution over all possible outcomes. However, these approaches are not explicitly designed to address the problem of complex and high dimensional negotiations.

Two negotiation approaches that specifically address the problem of complex interdependencies between multiple issues — and are therefore most related to our work — are [5, 6]. Klein et. al. [5] use a setting similar to the one considered in this paper, namely bilateral negotiations over a large number of boolean-valued issues with binary interdependencies. In this setting, they compare the performance of two search approaches: hill-climbing and simulated annealing and show that if both parties agree to use simulated annealing, then Pareto-efficient outcomes can be reached. In a similar line of work, Lin [6] uses evolutionary search techniques to reach optimal solutions. Both of these approaches have the advantage that they are scalable to large numbers of issues and Pareto-efficient outcomes can be reached without any prior information (such as the utility super-graph for a class of buyers in our model). However, a drawback of these learning techniques is the large number of negotiation steps needed to reach an agreement (around 2000 for 50 issues [5]).

By comparison with this work, our approach uses an explicit model of the buyer utility function - in the form of a utility graph. A difference of our approach (presented both in this paper and in [10]) from other existing negotiation approaches is that we use information from previous negotiations in order to aid buyer modeling in future negotiation instances. This does not mean that negotiation about specific customers need to be stored, only aggregate information about all customers. The main intuition behind our model is that we explicitly utilize, during the negotiation, the clustering effect between

utility functions of a population of buyers, an effect which is used by most Internet recommendation engines today. This allows us to handle very high dimensional and complex negotiations efficiently (with a limited number of negotiation steps).

The main contribution of this paper, in addition to the one highlighted in [10], is that it shows that the whole process can be automatic: no human input is needed in order to achieve efficient outcomes. We achieve this by using techniques derived from collaborative filtering (widely used in current e-commerce practice) to learn the structure of utility graphs used for such negotiations. We thus show that the link between collaborative filtering and negotiation is a fruitful research area, which, we argue, can lead to significant practical applications of automated negotiation systems.

References

1. Bela Bollobas. *Random Graphs*. Academic Press, London, England, 1985.
2. U. Chajewska and D. Koller. Utilities as random variables: Density estimation and structure discovery. In *Proceedings of sixteenth Annual Conference on Uncertainty in Artificial Intelligence UAI-00*, pages 63–71, 2000.
3. J. K. Debenham. Bargaining with information. In *3rd Int. Conf. on Autonomous Agents & Multi Agent Systems (AAMAS), New York, July 19-23, 2004*, pages 663–670, 2004.
4. E. Gerding, D.J.A. Somefun, and J. A. La Poutré. Multi-attribute bilateral bargaining in a one-to-many setting. In *Proc. of the AMEC VI Workshop, New York, USA, 2004*.
5. M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation*, 12:111–125, 2003.
6. R. Lin. Bilateral multi-issue contract negotiation for task redistribution using a mediation service. In *Proc. Agent Mediated Electronic Commerce VI*, New York, USA, 2004.
7. R. M. Coehoorn N. R. Jennings. Learning an opponent’s preferences to make effective multi-issue negotiation tradeoffs. In *Proc. 6th Int Conf. on E-Commerce, Delft, 2004*.
8. N. R. Jennings P. Faratin, C. Sierra. Using similarity criteria to make issue trade-offs in automated negotiations. *Journal of Artificial Intelligence*, 142(2):205–237, 2002.
9. H. Raiffa. *The art and science of negotiation*. Harvard University Press, Cambridge, Massachusetts USA, 1982.
10. Valentin Robu, D.J.A. Somefun, and J. A. La Poutré. Modeling complex multi-issue negotiations using utility graphs. In *4th Int. Conf. on Autonomous Agents & Multi Agent Systems (AAMAS), Utrecht, The Netherlands, 2005 (to appear as full paper)*. Available at: <http://homepages.cwi.nl/~robust/AAMAS05.pdf>, 2005.
11. N. Jennings S. Fatima, M. Woolridge. Optimal negotiation of multiple issues in incomplete information settings. In *3rd Int. Conf. on Autonomous Agents & Multi Agent Systems (AAMAS), New York*, pages 1080–1087, 2004.
12. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Tenth International WWW Conference (WWW10), Hong Kong*, 2001.
13. D.J.A. Somefun, T.B. Klos, and J.A. La Poutré. Online learning of aggregate knowledge about nonlinear preferences applied to negotiating prices and bundles. In *Proc. 6th Int Conf. on E-Commerce, Delft*, pages 361–370, 2004.
14. N. Shadbolt H. Leung J. H. Lee X. Luo, N. R. Jennings. A fuzzy constraint based model for bilateral multi-issue negotiations in semi-competitive environments. *Artificial Intelligence Journal*, 142 (1-2):53–102, 2003.