Automated Interactive Sales Processes*

Tomas Klos[1], Koye Somefun[2], and Han La Poutré[3,4]

[1]Algorithmics Group,
Delft University of Technology, The Netherlands
[2]Fortis Investments, Amsterdam, The Netherlands
[3]National Research Institute
for Mathematics and Computer Science in The Netherlands
(CWI)
Amsterdam, The Netherlands
[4]Faculty of Technology Management,
Technical University Eindhoven, The Netherlands

December 24, 2009

Abstract

When we look at successful sales processes occurring in practice, we find they combine two techniques which have been studied separately in the literature. Recommender systems are used to suggest additional products or accessories to include in the bundle under consideration, and multi-issue negotiation focuses on optimizing the precise configuration of the bundle and its price. In this paper, we pursue the joint automation of such interactive sales processes.

We present some key insights about, as well as a procedure for locating mutually beneficial alternatives to the bundle currently under negotiation. The essence of our approach lies in combining aggregate (anonymous) knowledge of customer preferences, learnt by the shop

*We want to thank the anonymous reviewers for helpful comments on our paper.

in interactions with previous customers, with current data about the ongoing negotiation process with the current customer. We present a memory- and a model-based method for online learning customer preferences and discuss their pros and cons. The performance of our system is illustrated using extensive computer experiments involving simulated customers with highly non-linear preferences which the system has no trouble learning.

1 Introduction

In every shop in daily life, one can encounter the phenomenon of the sales process: sales people ask customers for their demands, and try to come to a good deal. Often, this is based on suggesting other, additional products forming an interesting bundle together, or by giving discounts and negotiating about the price. Price negotiation usually quickly shifts to negotiating about the contents of a bundle of items with an overall price, where a sales person may prefer to add an item to a bundle with a reduced extra price, rather than discounting the overall price of the existing bundle. An example is car sales, where accessories like light metal rims or a stereo-system may be included in the overall deal at a reduced price, but also attractive financing or insurance opportunities may be offered, and even trading-in one's old car may make a difference. Other examples include negotiating different aspects of work contracts between an employer and its workers, or of procurement contracts between firms.

Such adding, removing, or substituting items in a bundle is often initiated by the sales person, who has the most domain knowledge, regarding both interesting additional items for a specific customer as well as the products in general, and regarding the price/profit ratios of these products. Often, customers have limited knowledge about the possibilities and have limited awareness of their needs (say, limited "active knowledge"). We mean this in the sense that, even though customers can value specific offers by the sales person, they cannot easily generate all interesting combinations themselves, simply because they don't know what's available. This is where good sales people can help and enhance the sales process, while still keeping focused on their sales results.

When analysing the above sales process, one can distinguish two important aspects that have been addressed in the literature separately. On the one hand, the suggestion of new products in a bundle is done in recommender systems, where clients with limited active knowledge get recommendations about other products based on their recent choices. On the other hand, negotiation about the price and options in a transaction is studied in multi-issue negotiation, where both negotiating parties have full active knowledge about the product domain, but with their own (often private) preferences.

In this paper, we consider the sales process by two parties: a sales person with full active domain knowledge, and a customer with limited active domain knowledge (i.e., passive domain knowledge). We aim at automating this process by a combination of negotiation and recommendation in an integrated and integrative fashion, that answers the needs of the two involved parties. In particular, our goal is to automate the role and activities of the sales person, in the form of an intelligent system (or agent) that can interact with the customer, or possibly her software agent. We especially focus on the intelligent and algorithmic aspects of such a solution, with an eye on potential short-term applicability as well as feasibility for the sales domain. We also keep an eye on keeping the models, protocols and solutions sufficiently stylized in order to allow robust and thourough research and experimentation results.

We discuss existing results in the related fields of multi-issue negotiation and recommender systems.

1.1 Integrative Multi-Issue Negotiation

One technique that has been studied for reaching agreements among automated agents is the use of integrative negotiation [8]. This allows the agents involved to explore the space of outcomes for mutually satisfying deals. In an online retail e-commerce setting, outcomes may represent goods with different aspects, such as delivery date, quality level, price, etc. Alternatively, as in the scenario we study, different combinations of goods and accessories together with a price may be the subject of negotiation. The key idea is that although each agent's *individually optimal* outcome will generally be unacceptable for the other, there will also be many outcomes that *both* agents like. Negotiation enables the agents to locate those outcomes, by successively proposing concessions [13].

Multi-issue negotiation is especially effective at reaching such win-win outcomes, since issues with different weights to each of the agents may be traded off against each other [18]. From the point of view of the sales person, this also allows for personalization of deals towards individual customers. Most multi-issue negotiation work, however, has focused on a small number of issues, and, more importantly, on linear preferences [6]. For our setting of automated sales processes, we need to address non-linearity, however, because very often, the value of some items will depend on whether certain other items are also obtained, sometimes in positive, and sometimes in negative ways, representing complementarities and substitutes, respectively. And we also need to address that customers often only have limited active knowledge about their prefernces.

1.2 Bundling and Recommender Systems

Another relevant approach in the context of selling multiple items is *bundling*, which means combining two or more items and selling them as one good [1]. Bundling can be a very effective sales strategy due to the possibility of reducing the costs of producing, marketing and selling products. In addition, and more importantly, bundling can stimulate the demand for related goods. Bundles are usually composed offline, where the seller analyzes historical sales data or uses expert knowledge to decide which combinations of goods to sell at which (reduced) prices. Such bundles are therefore usually limited in number, compared to the total number of products available or to the (exponential) number of theoretically possible bundles. Alternatively, such historical data or expert knowledge may be used as input for *recommender systems*, which allow sellers to assemble bundles online. This is done by suggesting to customers additional items known to be popular or otherwise suitable together with their initial selection. Collaborative or cognitive filtering methods may be used to establish the relevant similarities [7]. Usually, adaptive pricing schemes for (extended) bundles are not occurring in recommender systems, due to the (static) nature of the system with many products. Some (general) pricing scheme techniques do exist [12], which often work with discounts based on the number of products purchased. Thus, complementarities or substitutabilities between products are not expressed in the price here.

The drawback of these approaches is that the scope for designing both bundles and recommendations is limited by the static nature of the fixed datasets they build on. So even thought these methods are able to capitalize on the aggregate knowledge gathered in interactions with many different customers, this provides only partial personalization. A recommender system presents two customers selecting the same initial item, with exactly the same recommendations. Even though their preferences may be quite different, they are treated as if they are identical. In reality, of course, no two customers are identical, or, in the words of amazon.com founder Jeff Bezos: “We have 6.2 million customers, we should have 6.2 million stores. There should be the optimum store for each and every customer.”[1] Also, pricing mechanisms are usually very limited in terms of personalization, due to the static nature of the corresponding kind of bundling, the limited number of bundles available, or the simplicity of the pricing schemes. The interactive nature of negotiation, on the other hand, does offer the shop possibilities for learning about an individual customer. In the past, however, this has not been exploited in combination with aggregate knowledge like available in recommender systems or human expert knowledge, but mainly for learning in interactions with single customers.

1.3 Automated Interactive Sales

In this paper, we introduce a solution for automatic sales processes, consisting of an intelligent combination of negotiation, bundle recommendation, and aggregate knowledge derivation. This approach relates to what happens in real life. In our approach, a shop and a customer negotiate about a bundle of products and its total price. The customer has limited active knowledge about his preferences. The shop uses aggregate knowledge for online decisions about which bundles to recommend to an individual customer during this process. This aggregated knowledge could come from human domain experts, but, as we will show, it can in particular be derived from past negotiations with anonymous customers (i.e., their identity need not be stored in relation to this data). This knowledge derivation technique makes the system self-contained, and it also allows for a high level of privacy for the customers. Also, it does not ask for additional efforts of customers like filling out interest forms and such, but merely uses past negotiation data. The shop's experience with an individual customer during a negotiation is used in addition to the aggregate background domain knowledge. This allows the interaction in the sales process to be truly tailored to an individual customer, and hence it enhances the level of personalization. Finally, the sales process is tailored

[1] Quote from a BusinessWeek Online Q&A published online at `http://www.businessweek.com/ebiz/9903/316bezos.htm` on March 16, 1999, last accessed on December 21, 2009.

towards reaching win-win results, so-called Pareto optimal outcomes, which are defined as outcomes in which one party can only get a better result at the cost of the other (and thus no 'free' opportunities are left unused).

2 Automated Interactive Sales: Approach and Stategy

We start from the problem description as given in Subsection 1.3. In more detail, we propose that the shop negotiates with a customer about the composition and price of a bundle of goods. A negotiation is modeled as an alternating exchange of offers and counter offers. Each offer is a proposed outcome or *contract*, consisting of a bundle of goods plus a price for that bundle. Such a proposal may be accepted by the other agent, or rejected and countered with a new proposal, a counter offer. Depending on the development of the negotiation process, the shop may recommend (negotiating about) an alternative bundle. The selection of such recommendations is based on an incremental search for more 'promising' bundles. The customer's counter offers to the shop's proposals provide the shop with valuable information about the customer's preferences, which may be used to further guide the search.

2.1 Pareto Efficiency and Gains from Trade

To explain what it means for a bundle to be 'promising,' we start by making the standard assumption that the customers and the shop are able to valuate contracts based on their *net monetary value* or 'utility' for them, i.e. the agents are able to express both valuations v and prices p of bundles in monetary terms. The utility of a contract (a bundle-price combination) (b, p_b) for a customer is then equal to her valuation of the bundle minus the price she needs to pay, and for the shop to the price paid, minus his valuation (usually interpreted as his cost):

$$u_c(b, p_b) = v_c(b) - p_b \quad \text{and} \quad u_s(b, p_b) = p_b - v_s(b).$$

If the shop and the customer 'trade the bundle for the price,' they both gain a certain amount of utility, so the sum of the utilities is equal to the gains arising from the trade.

Formally, the *gains from trade* (gft) associated with a bundle b are defined as the sum of the utilities for the shop u_s and for the customer u_c of contracts involving bundle b. Equivalently, the gains from trade are equal to the difference between the customer's and the shop's valuation v of the bundle [16]:

$$\begin{aligned}\mathrm{gft}(b) &= u_c(b, p_b) + u_s(b, p_b) \\ &= v_c(b) - p_b + (p_b - v_s(b)) \\ &= v_c(b) - v_s(b).\end{aligned}$$

Note that the price does not influence the size of the gains from trade for a bundle, but it does influence the way in which the bundle's gains from trade are split among the agents in terms of their respective utilities. The price resulting from a negotiation depends on the negotiation tactics employed by the agents, and on other external factors, such as outside options and the distribution of power among the agents. These factors are outside the scope of our research.

As explained in section 1.3, the shop wants to increase not only its own, but also the customer's satisfaction. This means the shop searches for Pareto optimal or Pareto efficient outcomes. Previously we have shown that such Pareto efficient outcomes are the ones involving the bundle(s) with the highest gains from trade [19],[2] so the shop needs to search for these bundles with the highest gains from trade attainable. (For contracts involving such bundles, increasing one partner's utility necessarily means making the other partner worse off, i.e. they do not allow Pareto improvements.) Since the shop knows its own valuation of bundles, the challenge is to estimate the customer's valuations in order to determine different bundles' gains from trade, and make an estimate of which ones represent Pareto improvements to the bundle currently being negotiated.

2.2 Searching Multi-Issue Recommendations

The problem facing the shop is to locate Pareto efficient bundles (or bundles with the highest joint utility or gains from trade), while the customer's valuations for bundles are not only unknown to the shop, but also non-linear in

[2] Because we do not focus on negotiation tactics, in our experiments we are only interested in the extent to which Pareto efficiency is approximated, and not in the agents' relative utilities obtained.

her valuations of the individual goods. In addition, the customer is assumed to have limited active knowledge about her preferences, i.e., she will not be able to propose alternative bundles, but only to value bundles proposed by the shop. The shop uses feedback from negotiations with the current and past customers to dynamically build up aggregate knowledge about its customers. It uses such aggregate knowledge together with negotiation feedback from the current customer to guide an incremental search process for bundles with progressively higher gains from trade.

Without loss of generality, we assume that the interactive process starts with the customer selecting an initial bundle and price. We call this bundle the (shop's estimation of the customer's) 'interest bundle.' The shop and the customer negotiate about the price for the interest bundle, until the shop decides that an alternative bundle should be negotiated about. This decision is based on the following. In each round of the negotiation, the shop estimates Δt, the remaining time required to reach a deal in the negotiation about the current bundle, by intersecting its own sequence of planned future concessions with a linear extrapolation of the customer's two most recent offers. If we let $O = (b, p)$ and $O' = (b, p')$ denote the customer's current and previous offers for bundle b, then the estimated remaining number of negotiation rounds is estimated as:

$$\Delta t = \frac{v_s(b) - p}{p - p'},$$

where $v_s(b)$ is the shop's valuation of the bundle, and p and p' are the customer's two consecutive offered prices for bundle b. The decision to generate a recommendation is then stochastic, with a probability $\Pr_{\text{recomm}}$ that increases in this estimated required time:

$$\Pr_{\text{recomm}} = 1 - \exp(-0.25\Delta t).$$

If the shop decides that a recommendation is due, an alternative bundle b' is selected from the set B of bundles in the interest bundle b's 'neighborhood,' which is defined as the set of bundles which differ from the interest bundle b by the presence or absence of just one good (a bitflip in the interest bundle's bitstring representation). Thus, estimations of alternative bundles' gains from trade have to be made only for a small subset of all possible bundles. Moreover, the customer will perceive the recommendation process as gradual, and will not perceive recommendations as appearing haphazard.

Selection of recommendations from the neighborhood B is done using the softmax method with a Boltzmann distribution and a decreasing temperature τ [20]. This means each bundle $b' \in B$ has a probability $\Pr(b')$ of being selected to be the recommendation, which is based on the bundle's estimated 'gains from trade difference:'

$$\Pr(b') = \frac{\exp(\Delta\text{gft}(b, b')/\tau)}{\sum_{b' \in B} \exp(\Delta\text{gft}(b, b')/\tau)},$$

where $\Delta\text{gft}(b, b') = \text{gft}(b') - \text{gft}(b)$ is the estimated difference in gains from trade between bundles b' and b. The time-decreasing parameter τ is the temperature, which determines the rate of exploration versus exploitation: if $\tau = 0$ there is no exploration, and softmax turns into greedy search. Having exploration is useful because the shop has imperfect knowledge about the specific customer it is interacting with.

With knowledge of his own valuation of the bundles b' in the neighborhood B, the shop thus needs to estimate the customer's valuation of those bundles, or equivalently, the gains from trade difference for each of the bundles $b' \in B$. The recommendation is presented to the customer along with a price determined by the shop's negotiation tactic, and the negotiation process continues. A crucial feature of our approach is that, in contrast to current 'passive' recommendation methods, embedding recommendations in a negotiation process enables the shop to learn about the appropriateness of the recommendation *for this particular customer*. Depending on the 'enthousiasm' of the customer's counter offer to the offer containing the recommendation, the shop designates the recommended bundle as the new interest bundle (and thus diverts the search to this new interest bundle's neighborhood), or abandons the recommended bundle and generates a new recommendation from the current interest bundle's neighborhood. This process continues until a deal is reached, or until one of the agents leaves the negotiation.

In this way, we are able to combine *aggregate knowledge* about the 'average' customer, learned in interactions with many customers, with *customer-specific knowledge* learned in the unfolding negotiation process. The aggregate knowledge is used to start off the interaction, while negotiation data may override aggregate knowledge-based decisions in favor of adaptations of the process to the responses of the individual customer.

2.3 Predicting Preferences

A very important component of our method is yet to be discussed: the shop needs a method for learning aggregate knowledge about customers' valuations for bundles. In line with concerns for customers' privacy, this method should use only *anonymized* negotiation data. We have chosen to use only implicit 'ratings,' i.e. ratings which are derived from (implicit in) the customer's behavior: they do not require the customer to deviate from what she was doing already. A prime example is sales data, and in our case negotiation data. Implicit ratings are easily anonymized, and they are less intrusive to collect than explicit ratings provided by customers by filling in forms and scoring items in ways other than by buying them. In particular, we use data collected in the negotiation process as ratings: not just for learning specific knowledge about the current customer, but, in anonymized form, also for learning aggregate knowledge about customers in general. The advantage of using negotiation data as ratings is that providing them is incentive compatible for the customer, who benefits from negotiating better prices, while the shop learns more about the customer's valuations. Furthermore, this yields a closed system, with no need for exogenous data.

Multi-issue negotiation has been studied thoroughly, for example studying the influence on the efficiency and speed of deals reached by agents of different protocols, often using mediators [10, 13]. Studies of learning in multi-issue negotiation generally find that enabling agents to learn about aspects of the opponent increases the performance of the process in terms of locating profitable negotiation outcomes, and using less negotiation rounds. This is the rationale underlying this part of our system as well. In an early example, Zeng and Sycara [21] use Bayes' rule to update beliefs about the opponent's reservation price. Others have proposed methods to learn the opponent's preference function, like Buffet and Spencer [3] who use Bayesian classification of an opponent's preferences function. The effectiveness of their model depends on the preference functions in the various classes, and furthermore they assume the opponent is making concessions, whereas we are able to handle the general case. Jonker, Robu and Treur [11] find that revealing little information can lead to efficiency improvement using a model for estimating attribute-weights for a linear utility function. Another approach for estimating weights of linear utility functions, is used by Coehoorn and Jennings in [4], and by Hindriks and Tykhonov in [9], who also allow an agent to learn the actual preference values. The restriction to linear preference functions is also used in [15], who develop an automated agent that learns an opponent profile from a given fixed set of possible profiles. The agent can outperform human negotiators. Here, we're explicitely interested in non-linear utilities. These are also treated by Lai, Li and Sycara in [14], who propose a model where agents propose multiple goods, or even complete iso-utility curves as offers. However, they show results for only 3 issues. The most closely related work is by Robu, Somefun and La Poutré [17], who use negotiation data to learn parameters of a so-called *utility graph*, which expresses interdependencies between (valuations for) goods. Their approach, however, requires that the seller knows the structure of the utility graph.

Two types of methods can be used for predicting customer preferences in recommender systems [2]: memory-based and model-based. Memory-based methods operate on collected data or 'ratings' directly, while model-based methods use such data to estimate the parameters of a given model, which is then used to predict preferences. Memory-based methods are more efficient, in that they generate predictions without the need for preprocessing data (i.e. estimating a model), but they suffer from scalability problems, unlike model-based methods which typically scale well. For model-based methods, on the other hand, the challenge lies in selecting the right model and in actually estimating it, while these methods are very fast at generating predictions once the models have been estimated.

First we assume that bundling will be worthwhile only for 'subdomains' of a shop's catalogue containing limited numbers of goods compared to the shop's complete catalogue of items. When considering the purchase of a camera, for example, a suitable bag and a tripod and possibly a limited number of other items are relevant accessories. In general, the range of individual goods the shop will consider in negotiating with his customers will be limited. For such domains (i.e. $n = 10$ individual goods), the exponential memory requirements of memory-based methods for storing data about all possible bundles is no impediment. We will present a memory-based method for this setting in Section 3. Its advantage is that it works independently of a model of the customers' preferences, while its main disadvantage of course, is that it doesn't scale well. In particular, for our experiments with this memory-based method, 20 items is already too much, so for those circumstances, we develop a model-based method which scales much better (Section 4). Content-wise, the first method is very general, in that it requires no model of the customer's preferences. The second method, on the other hand, estimates a model of the customer's valuation function, and thus requires that model to be of sufficiently manageable complexity to actually be estimated.

3 The Memory-based Method MeB

Our memory-based method MeB learns the gains from trade *difference* $\Delta\text{gft}(b, b')$ between all possible bundles b and all bundles b' in their respective neighborhoods. Right before and after a recommendation, the customer will bid consecutive offers about the original bundle b and the recommended bundle b' from b's neighborhood, respectively. A comparison of the prices in those offers indicates to the shop what the customer's *difference* in valuation of the 2 bundles is.

Knowing his own valuation, these offers also tell the shop the gains from trade difference $\Delta\text{gft}(b, b')$ between the 2 bundles b and b'. The shop averages these differences for each such (b, b')-pair across all customers who make consecutive bids for the bundles involved in this pair (b, b'). Although the customer may strategically misrepresent her valuations, we assume this effect on average to be negligible between consecutive offers because they are made so close together in time. Moreover, some customers will bid offers in the order (b, b') and some in the order (b', b), thus further diminishing the effect of strategic misrepresentation. Also note that each pair of offers obtained from *one* customer, may be used *twice*, namely to estimate the differences between both (b, b') as well as (b', b). This gives the shop an estimate of the average or 'typical' customer's gains from trade difference between each such pair. It uses these differences for softmax-selecting a bundle to recommend from a given interest bundle's neighborhood, as explained above.

3.1 Experiments

We have evaluated our methods in numerical experiments involving simulated customers. To test the MeB method, we used $n = 10$ individual goods, and compared the MeB method's performance against two benchmarks providing heuristic upper and lower bounds on performance.

We repeated each experiment a number of times using different random-seeds. In each experiment, we determined the Pareto optimal bundle and the gains from trade associated with it, by performing a brute force search of all bundles, for each distinct combination of instances for the customer's and the shop's valuations. This allows us to measure to what extent the Pareto efficient outcome is approximated.

Valuations and Negotiation Tactics A customer's valuation of a bundle is constructed as the sum of her valuations for the individual goods, plus her (positive or negative) valuations for the interactions among different subsets of goods, up to interaction effects among 3 goods. All these valuations are drawn uniformly at random, from the range $[0, 250]$ for individual goods and $[-250, 250]$ for the interaction effects. The customer's choice of an initial bundle is determined as a randomly chosen bundle at a distance of 3 bit-flips from the optimal bundle, so that the shop's method has some room for improvement.

In the experiments, we gave both the customer and the shop a time-dependent concession tactic [5], called TDF (for Time-Dependent Fraction). This tactic starts bidding at some distance above (for the shop) or below (for the customer) the agent's valuation, and gradually approaches the valuation over time, using a shrinking fraction of the valuation. Each agent had a 2% probability of breaking off negotiations in each round.

Benchmarks To the best of our knowledge, no prior work has addressed the same situation as we have. We have therefore chosen to implement two benchmarks to compare the performance of our system with. The first benchmark (called PK for Prior Knowledge) simulates the shop having access to aggregate knowledge in the form of the actual probability distributions underlying customers' valuations (but not to individual customers' valuations). This provides a heuristic *upper* bound for the performance of our methods. In essence, this benchmark indicates what is the most that can be expected from our system, although for achieving this it would still have to *learn* this aggregate knowledge. We will see that our system is indeed able to learn this. A *lower* bound is provided by the other benchmark, called RR for Random Recommendation. This benchmark simulates the shop generating random recommendations from the interest bundle b's neighborhood, without taking any aggregate knowledge about gains from trade into account. We would hope that our system performs at least as good, but preferably much better than this rather uninformed approach to generating recommendations. This is indeed confirmed in our experiments.

3.2 Results

The main results for this experiment are shown in Figure 1. The 'relative per-

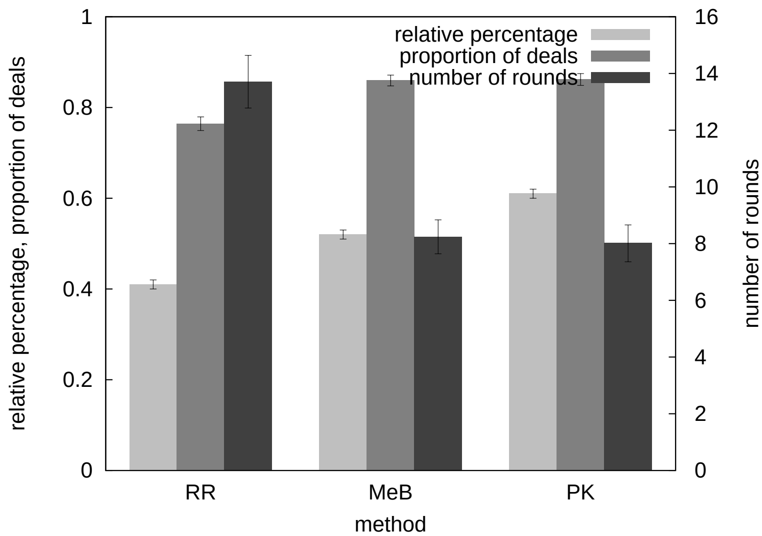

Figure 1: Comparison of MeB with the benchmarks PK and RR in experiment I. These results are averages across 10 runs with different random seeds, and 12000 customers per run. Standard deviations are indicated as errorbars. Note that for each method, the 2 bars on the left are measured on the left y-axis, while the bar on the right is measured on the right y-axis.

centage' numbers measure the extent to which the methods are able to reach Pareto optimal outcomes, *relative to* the gains from trade of the negotiation's starting bundle, $\mathrm{gft}(b_{\mathrm{init}})$:

$$\text{relative percentage} = \frac{(\mathrm{gft}(b_{\mathrm{final}}) - \mathrm{gft}(b_{\mathrm{init}}))}{(\text{max. gft} - \mathrm{gft}(b_{\mathrm{init}}))},$$

where b_{final} is the bundle under negotiation when the process ends, irrespective of whether this is because a deal is reached or because of the 2% exogenous break-off probability, and max. gft is the maximum gains from trade attainable across all bundles, which we established for each shop-customer interaction by brute force search over all possible bundles. Presenting performance relative to the minimum gains from trade attainable would obviously give higher percentages, but is deemed an unrepresentative measure.

Figure 1 shows that the MeB method enables the shop to reach deals with many customers very quickly, by online learning about the customers' preferences and applying this knowledge in negotiations with future customers. Especially compared to what would be possible when the shop already had access to the probability distributions underlying the customers' preferences, the MeB method is competitive in terms of both the fraction of deals reached, as well as the number of negotiation rounds required to reach a deal.

Figure 2 shows the development of the relative percentage score of the

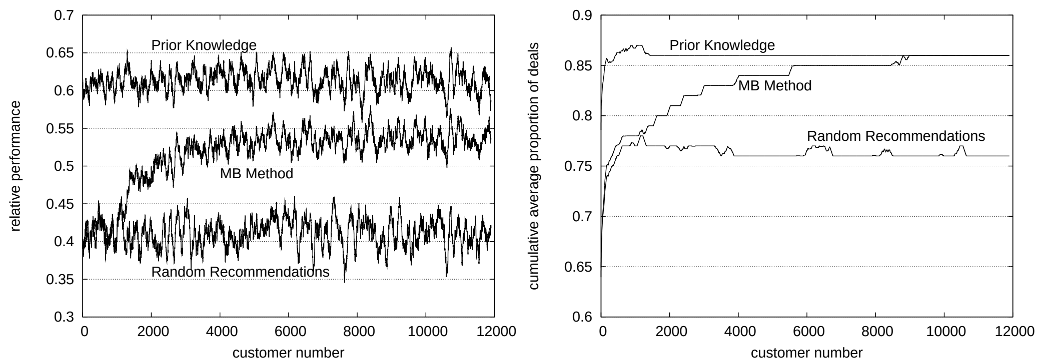

Figure 2: Relative percentage of the MeB method on the left, compared to the RR- and PK-benchmarks; cumulative average fraction of deals reached on the right. Both graphs show 100-customer moving averages.

MeB method over time (on the left). Each datapoint is the average across 10 customers, drawn from different random distributions. The MeB method's performance starts out at the level attained by recommending randomly chosen bundles from the interest bundle's neighborhood, which is precisely what this method does initially, before it has begun to build up aggregate knowledge. Eventually, the MeB method is able to attain about 85% of the performance of the PK benchmark, which has unrestricted access to the complete probability distributions underlying the customers' preferences. The graph on the right shows the performance in terms of the fraction of customers with which a deal is reached: starting at the level of random recommendations again, the MeB method eventually reaches deals with the same fraction of customers as the PK benchmark, although these deals are of lower quality in terms of gains from trade involved.

4 The Model-based Method MoB

The MoB method predicts a customer's valuation of a bundle as a function of the bundle's configuration. A customer's valuation is calculated as the sum of the customer's valuations of the individual goods, plus the valuations for all subsets of size ≥ 2 of goods in the bundle. In general, interaction effects may occur among subsets of goods up to a certain size $m \leq n$, the number of goods. In practical circumstances, however, m will be limited, for example because of limitations on humans' cognitive abilities. For choosing a suitable value for m, we consider that humans are able to consider a maximum of 7 ± 2 items at the same time, and we pick $m = 3$ so that we allow sufficient non-linearity while keeping our models manageable. In general, this gives a total of $k = \sum_{i=1}^{m} \binom{n}{i}$ parameters to be estimated, which is $k = \frac{1}{6}n^3 + \frac{5}{6}n$ for $m = 3$, so $k = 1350$ for $n = 20$.

A customer's valuation for a bundle, $v_c(\mathbf{b})$, is then calculated as

$$v_c(\mathbf{b}) = \sum_{p=1}^{n} a_{<p>} b(p) + \sum_{p=1}^{n}\sum_{q=1}^{n} a_{<p,q>} b(p)b(q) + \sum_{p=1}^{n}\sum_{q=1}^{n}\sum_{r=1}^{n} a_{<p,q,r>} b(p)b(q)b(r),$$

where $a_{<p,q>}$ is the customer's valuation (a real number) for obtaining just goods p and q together, etc., and $b(p)$ is 1 if bundle $\mathbf{b}$ contains good p, and 0 otherwise. Let $\mathbf{a}$ be the k-dimensional vector of these parameters $a_{<p>}$, $a_{<p,q>}$, and $a_{<p,q,r>}$ in an arbitrary, but systematically chosen order. Formally, there exists a fixed mapping $\mathbf{f} : \{0,1\}^n \mapsto \{0,1\}^k$, known by the shop, which maps a bundle in binary notation to a k-dimensional vector of bits signifying whether each of the corresponding elements of the k-dimensional vector of parameters $\mathbf{a}$ is relevant when determining a customer's valuation: for any bundle $\mathbf{b} \in \{0,1\}^n$, $v_c(\mathbf{b}) = \mathbf{f}(\mathbf{b})^T \mathbf{a}$.

As an example, when $n = 3$, a customer's valuation of the bundle $\mathbf{b} = (1,1,0)$, i.e. the bundle containing only goods 1 and 2, is $v_c(1,1,0) = a_1 + a_2 + a_{1,2}$. Depending on the encoding, we could have $\mathbf{f}(\mathbf{b}) = (1,1,0,1,0,0,0)$, but irrespective of the encoding, $\mathbf{f}(1,1,0)$ should contain three 1's and four 0's, since only the three parameters a_1, a_2 and $a_{1,2}$ contribute to the customer's valuation of this bundle.

The idea of the MoB method is that the shop estimates the k parameters in the vector $\mathbf{a}$ from observations of customers' valuations of bundles. If the shop has t observations $i = 1, \ldots, t$ of a customer's valuation v_i for bundle $\mathbf{b}_i$, then

$$\mathbf{v} = \mathbf{B}\mathbf{a}, \tag{1}$$

where $\mathbf{v} = [v_1, \ldots, v_t]^T$ and $\mathbf{B} = [\mathbf{f}(\mathbf{b}_1), \ldots, \mathbf{f}(\mathbf{b}_t)]^T$. Since the shop knows $\mathbf{f}$, he knows both $\mathbf{v}$ and $\mathbf{B}$, and if the rank of the matrix $\mathbf{B}$ is equal to k, he can calculate the vector of parameters $\mathbf{a}$ as $\mathbf{a} = \mathbf{B}^+\mathbf{v}$, where $\mathbf{B}^+$ is the (Moore-Penrose) generalized matrix inverse of $\mathbf{B}$, and $\mathbf{B}^+\mathbf{v}$ is the least squares solution of the system of linear equations described by equation 1. We solved the two problems this approach has, namely that the vector $\mathbf{a}$ is typically different for different customers (such that it is more accurately interpreted as a random vector), and that the shop does not observe a customer's *valuations* of bundles but rather her *bids* for those bundles. We first discuss the former problem (Section 4.1), and then show how a clever preprocessing of negotiation data can generate the high quality data necessary for the first step (Section 4.2).

4.1 Estimating Means and Co-variances

Since valuations will normally differ from one customer to the next, each customer really has an individual vector $\mathbf{a}_c$ containing her preferences, and the vector $\mathbf{a}$ would be a random vector, described by a vector of means $\mu_\mathbf{a}$ and a co-variance matrix $\mathbf{\Sigma_a}$ giving rise to a multivariate distribution from which the individual customers' vectors $\mathbf{a}_c$ are drawn. In order to be able to search Pareto optimal deals with new customers, the shop wants to estimate $\mu_\mathbf{a}$ and $\mathbf{\Sigma_a}$. Assume for the moment that the shop sees customers' valuations rather than bids. (As announced, we discuss below how the shop may go from bids to valuations.) If the shop would negotiate for a long time with customer c, while constantly suggesting new bundles, then eventually this would yield $\mathbf{v} = [v_1, \ldots, v_t]^T$ and $\mathbf{B} = [\mathbf{f}(\mathbf{b}_1), \ldots, \mathbf{f}(\mathbf{b}_t)]^T$ such that the rank of $\mathbf{B}$ is k, as required for solving equation 1. In the absence of strategic behavior, the shop would then be able to estimate $\mathbf{a}_c$ from negotiation data $\mathbf{v}$ and $\mathbf{B}$. It would then become straightforward to estimate $\mathbf{a}$'s mean $\mu_\mathbf{a}$ and co-variance matrix $\mathbf{\Sigma_a}$ from data about individual customers' preferences.

For large k, of course, customers will break off negotiations long before the matrix $\mathbf{B}$ grows into a rank k matrix. However, the general idea may be usefully employed to at least *partly* reveal $\mathbf{a}_c$ for *most* customers c, as follows. The general idea is that for each individual customer c, the shop obtains $\mathbf{a}'_c$, containing the values of a small subset of $\mathbf{a}_c$, and uses the combination of all these values over multiple customers to estimate $\mu_\mathbf{a}$ and $\mathbf{\Sigma_a}$. As with $\mathbf{a}_c$ above, this k'-dimensional vector $\mathbf{a}'_c$ (with $k' \ll k$) is obtained as the solution of the customer c-specific, and reduced version of equation 1, namely $\mathbf{a}'_c =$

$\mathbf{B}'^{+}\mathbf{v}'$. This requires that the rank of the matrix $\mathbf{B}'$ is k', which is guaranteed if the sequence of suggestions meets certain conditions. The sequence of bundles $\mathbf{b}_{l+1}, \ldots, \mathbf{b}_{l+\delta}$ suggested is constructed in such a way that strictly less individual goods are considered than is possible. In the experiments, 4, 5, or at most 6 individual goods are considered. By doing so, it becomes feasible to completely describe the required k'-dimensional subspace using just a limited number of recommendations. With these suggestions, the vector $\mathbf{f} = \sum_{i=1}^{\delta} \mathbf{f}(\mathbf{b}_{l+i})$ contains only a few nonzero elements. This sequence of recommendations gives rise to a matrix $\mathbf{B} = [\mathbf{f}(\mathbf{b}_{l+1}), \ldots, \mathbf{f}(\mathbf{b}_{l+\delta})]^T$ whose rank k' is equal to the number of nonzero columns. If the zero-columns are removed, we have the $\delta \times k'$ dimensional matrix $\mathbf{B}'$, which, along with the customer's valuations of the bundles in the sequence, $\mathbf{v} = [v_l, \ldots, v_{l+\delta}]^T$, the shop may use to estimate $\mathbf{a}'_c$, the k' dimensional vector containing all the *identifiable* parameters of $\mathbf{a}_c$.

Intuitively, the shop recommends a sequence of bundles involving only a small number of individual goods. This way, these bundles map to a sequence of corresponding vectors of parameters with the required rank for performing the matrix inversion. In addition, consecutive recommended bundles differ from each other with Hamming distance of 1, so that these recommendations don't appear too haphazard.

4.2 Pre-processing

With strategic behavior, the shop does not observe the customer's valuations $\mathbf{v}$, but rather the customer's corresponding vector of bids $\mathbf{p} = [p_l, \ldots, p_{l+\delta}]^T$. In order to go from observed bids to estimations of valuations, the shop needs a model of the customer's strategic behavior. The essence of our pre-processing method is to make this model explicit, estimate it, and check whether it is consistent with a customer's bidding behavior. Rather than valuations $\mathbf{v}$ in the previous subsection, the shop uses estimated valuations $\mathbf{v}^e$, which he predicts using the model

$$p_t = g(t)[cnst \cdot (\mathbf{f}(\mathbf{b}_t)^T \mathbf{a}_c)], \tag{2}$$

where $cnst \in \mathbb{R}$ is a constant, $\mathbf{f}(\mathbf{b}_t)^T \mathbf{a}_c$ denotes the customer's valuation for bundle $\mathbf{b}_t$, and $g : \mathbb{N} \mapsto \mathbb{R}$ is some function of the negotiation round. So, the bid in a given negotiation round t is assumed to be composed of a fixed and a variable, strategic, component, $cnst$ and $g(t)$, respectively. The fixed component specifies how much the customer is at most willing to pay for the bundle, and the variable component specifies how the customer's bidding proceeds over time.

First, the shop will use $l + \delta$ observations to estimate $g(t)$. For the first l of these observations, the shop makes no recommendation, so these are all about the same bundle so that, presumably, differences in the customer's consecutive bids represent $g(t)$. Given a fixed functional form, the shop then fits a function g^e (an estimation of $g(t)$ in Equation 2) through these first l bids. In real world applications, the shop might have a collection of bargaining models it can try, or estimate g^e assuming a different functional form, e.g. exponential instead of linear. Over the course of the next δ observations, the shop suggests different bundles and uses g^e to estimate valuations for these bundles as $v^e(t) = p(t)/g^e(t)$ for $l \leq t \leq \delta$. These valuations are used by the method described in the previous section to estimate $\mathbf{a}_c$. Finally, the shop will use an additional γ observations to perform consistency checks: if the estimated model does not predict these data, then the valuations estimated from this model are not included in the subsequent step of estimating coefficients of the customer's utility function (as described in Section 4.1).

4.3 Experiments

The experiments we performed to evaluate the performance of the MoB method were set up in exactly the same way as those in Section 3.1. The only difference is that we used $n = 20$ goods, and the customer's initial bundle was located at 5 bitflips from the optimal bundle. Also, we only simulated sequences of 5000 customers, and we replicated each of those 30 times. Table 3 gives the overal results for experiment II. The labeling of the rows is the same as that in Table 1. Note the high standarddeviations of the maximum and minimum gains from trade attainable, indicating significant differences between the 30 problem intances. However, the standard deviations of the important performance indicators is low, showing robustness of the results across those problem instances. The negative gains b_{final} for the RR benchmark means that it often finds a bundle for which there is no 'zone of agreement,' i.e. a bundle with negative gains from trade; this also partly explains the low average number of deals.

Figure 4 shows the quick learning of the MoB method. In particular, the MoB method quickly reaches the performance levels of the PK benchmark, which has direct access to the probability distributions underlying customers'

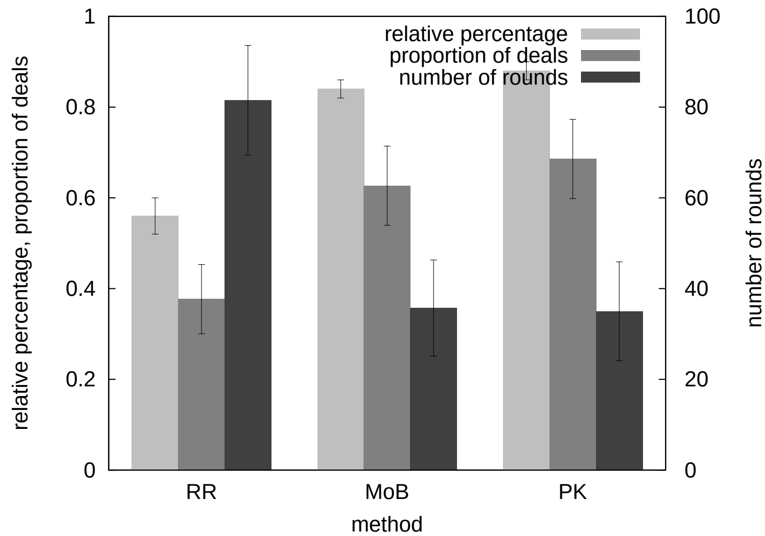

Figure 3: Comparison of MoB with the benchmarks PK and RR in experiment II. These results are averages across 30 runs with different random seeds, and 5000 customers per run. Standard deviations are given between brackets.

valuations. About 500 customers suffices for the MoB method because the systems obtains such large amounts of data from those individual early customers: abour 55 negotiation rounds, and valuations for different bundles. This is because we condensed the learning phase into the first 500 customers. In a practical application, the learning would normally have to be spread out across more customers, and they should be learned from more gradually. In the current setup, the initial customers are essentially exploited by having to respond to many different proposals from the shop, for the sole purpose of allowing the shop to learn. Many of the deals which are not reached by the MoB method are in fact not reached in interactions with exactly those initial customers.

5 Conclusions and Future Work

In this paper, we have outlined a novel utility-based approach to recommending, combining a number of distinct worlds. We view recommending in the context of bundling: recommendations are personalized bundles. Bundling

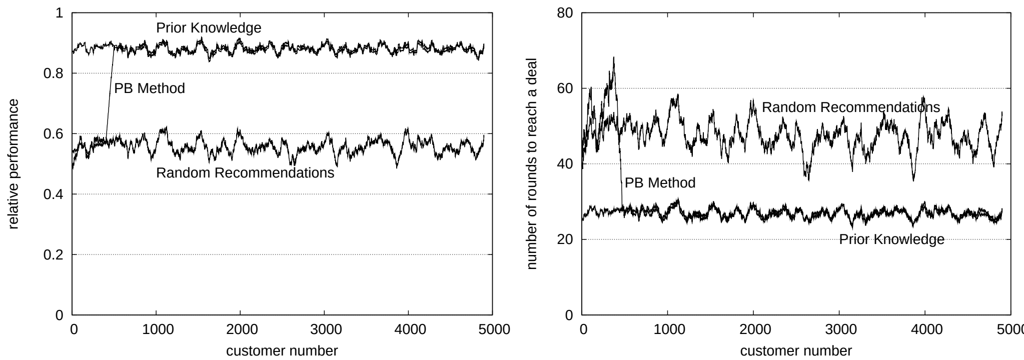

Figure 4: Relative percentage (on the left) and number of rounds required to reach deals (on the right) for the MoB method. The graphs present 100-customer moving averages.

provides opportunities for win-win outcomes, while multi-issue negotiation is especially effective at reaching win-win outcomes when they exist. Closing the circle, we connect recommendations with negotiations in a proposed system for adapting recommendations to knowledge learned about the customer in the interactive negotiation process. Numerical experiments involving simulated customers show our system's effectiveness in different circumstances. We feel that these first steps have opened up many opportunities for further research in this new and promising area.

It is important to note that the reason we don't report experiments with the MoB method beyond $n = 20$ is that we use brute force search of the exponential space of bundles for the Pareto optimal bundle (to be able to show the method's perfomance relative to the optimal bundle's gains from trade). The method itself can easily handle much larger bundles, since the number of parameters it needs to estimate is only a polynomial in the number of individual goods.

It is especially impressive that this method starts out with no prior knowledge about the distribution of its customers' valuations. It is able to learn such preferences quickly, up to point where its performance matches a system which starts out *with* such prior knowledge (the PK benchmark).

As for future work, we do not currently give the customer the opportunity to also adapt the bundle configuration. We chose not to include this possibility because in our experiments, this would have required an ad-hoc model of how the customer makes such decisions. In an actual application, on the other hand, the customer could easily be given this option. With currently available online product review sites and corresponding well-prepared customers, this would be all the more appropriate. All this would require in our system is that the shop would simply shift its estimation of the customer's interest bundle to whichever bundle the customer would suggest negotiating about.

Within negotiations, the shop would have the option of learning about the customer on the basis of more data than used now, such as trends in the customer's bidding behavior, and idiosyncratic choices not fitting with the aggregate knowledge. Furthermore, we have chosen a difficult application scenario by limiting ourselves to anonymized, privacy-protecting data. In certain settings, customers may very well be willing to give up some information about themselves, especially when the shop can make such information sharing incentive compatible like it can with negotation data.

A variety of extensions and alternative methods are possible for the domain problems reported in this paper. Recent related and concurrent work by several of the current authors is [17], which investigates a different trade-off between more effectively searching multi-issue negotiation proposals and making stronger assumptions about the (higher) level of active knowledge of the customer (the customer's level of 'rationality'). We expect that, with the current research in place, many subsequent research steps can be taken into various other directions.

References

[1] Yannis Bakos and Erik Brynjolfsson. Bundling information goods: Pricing, profits and efficiency. *Management Science*, 45(12), December 1999. 4

[2] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52. Morgan Kaufman, 1998. 11

[3] Scott Buffett and Bruce Spencer. A Bayesian classifier for learning opponents preferences in multi-object automated negotiation. *ECR&A*, 6:274–284, 2007. 10

[4] Robert M. Coehoorn and Nicholas R. Jennings. Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs. In *Proceedings ICEC*, pages 59–68. ACM, 2004. 10

[5] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3–4):159–182, 1998. 13

[6] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. Using similarity criteria to make issue trade-offs. *Artificial Intelligence*, 142(2):205–237, 2002. 4

[7] Alexander Felfernig, Gerhard Friedrich, and Lars Schmidt-Thieme. Recommender systems. *IEEE Intelligent Systems*, 22(3):18–21, 2007. 4

[8] Robert H. Guttman and Pattie Maes. Agent-mediated integrative negotiation for retail electronic commerce. In *AMET-98*, volume 1571 of *LNAI*, pages 70–90. Springer, 1999. 3

[9] Koen Hindriks and Dmytro Tykhonov. Opponent modelling in automated multi-issue negotiation using Bayesian learning. In *Proceedings AAMAS 2008*, pages 331–338, 2008. 10

[10] Takayuki Ito, Mark Klein, and Hiromitsu Hattori. A multi-issue negotiation protocol among agents with nonlinear utility functions. *Multi-Agent and Grid Systems*, 4:67–83, 2008. 10

[11] Catholijn M. Jonker, Valentin Robu, and Jan Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Journal of Autonomous Agents and Multi-Agent Systems*, 15(2):221–252, 2007. 10

[12] Jeffrey O. Kephart, Christopher H. Brooks, Rajarshi Das, Jeffrey K. MacKie-Mason, and Edmund H. Durfee. Pricing information bundles in a dynamic environment. In *Proceedings 3rd ACM Conference on Electronic Commerce (EC-2001)*, pages 180–190. ACM, 2001. 4

[13] Mark Klein, Peyman Faratin, Hiroki Sayama, and Yaneer Bar-Yam. Protocols for negotiating complex contracts. *IEEE Intelligent Systems*, 18(6):32–38, 2003. 3, 10

[14] Guoming Lai, Cuihong Li, and Katia Sycara. A decentralized model for automated multi-attribute negotiations with incomplete information and general utility functions. *Multi-Agent and Grid Systems*, 4:45–65, 2008. 11

[15] Raz Lin, Sarit Kraus, Jonathan Wilkenfeld, and James Barry. Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence*, 172:823–851, 2009. 11

[16] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Mircoeconomic Theory*. Oxford University Press, 1995. 7

[17] Valentin Robu, Koye Somefun, and Han La Poutré. Modeling complex multi-issue negotiations using utility graphs. In *Proceedings AAMAS 2005*, pages 280–287. ACM, 2005. 11, 22

[18] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994. 3

[19] Koye Somefun, Tomas B. Klos, and Han La Poutré. Online learning of aggregate knowledge about nonlinear preferences applied to negotiating prices and bundles. In *Proceedings* 6$^{\text{th}}$ *International Conference on Electronic Commerce (ICEC 2004)*, pages 361–370. ACM Press, 2004. 7

[20] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning*. MIT Press, 1998. 9

[21] Dajun Zeng and Katia Sycara. Bayesian learning in negotiation. *International Journal of Human-Computer Studies*, 48(1):125–141, 1998. 10