

Automated negotiation: An efficient approach to interaction among agents

Citation for published version (APA):

Weiss, G., & Chen, S. (2018). Automated negotiation: An efficient approach to interaction among agents. In *Interactions in Multiagent Systems* (pp. 149-177). World Scientific Publishing Company.

Document status and date:

Published: 01/01/2018

Document Version:

Publisher's PDF, also known as Version of record

Document license:

Taverne

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Chapter 1

Automated negotiation: an efficient approach to interaction among agents

1.1 Abstract

Negotiation is any process through which the players on their own try to reach an agreement. It is a task that has a broad spectrum of practical applications to a variety of social, economic and politic phenomena. When it comes to complicated problems such as negotiations with a large number of issues, finding good agreements is however a tough challenge for human beings, especially in the case that they are in lack of negotiation experience, opponent information and the available negotiation time is limited. In order to overcome these limitations, there exist considerable interests in automating their negotiation process by means of software agents to assist humans in the decision-making process. Automated negotiation therefore provides people with a realistic alternative solution. This chapter first overviews forms, protocols and three main approaches of automated negotiation, namely, heuristic, game theoretic and argumentation approaches. Then the focus is on the study of complex practical negotiation – multi-issue negotiation that runs under real-time constraints and in which the negotiating agents have no prior knowledge about their opponents' preferences and strategies. Finally, two classes of state-of-the-art negotiation agents for complex negotiation is presented, namely, the agents based on regression techniques and the agents based on transfer learning to support its decision-making process during negotiation.

1.2 Introduction

A frequent and important form of interaction in multi-agent systems is termed automated negotiation. Automated negotiation, as a fundamental

and powerful mechanism for managing interaction among autonomous agents, has become a subject of central interest in the multi-agent systems community. The main reason behind it is two-fold. First, it can support and facilitate human negotiators in reaching more efficient outcomes by compensating for the limited computational abilities of humans, especially in complicated negotiations, e.g., huge number of rounds, proposals of hundreds of issues, etc. Second, the spectrum of potential real-world applications is very broad and diverse, from electronic Commerce, cloud Computing, smart Grid to supply chain management. In this chapter we start with the definitions of automated negotiation, and then focus on the study of complex practical negotiation, namely, multi-issue negotiation that runs under real-time constraints and in which the negotiating agents have no prior knowledge about their opponents' preferences and strategies. Lastly, two classes of state-of-the-art negotiation agents for complex negotiation are detailed respectively.

1.3 Automated negotiation

Negotiation can be any process through which the players on their own try to reach an agreement – a task that has many practical applications to a variety of social, economic and politic phenomena [Muthoo (1999)]. Consider the following example: John owns a car that he values at \$25,000 in his opinion, and his friend – Hendric would like to buy the car at a maximum price of \$50,000. They would therefore have a common interest to have the deal if a contract in which the agreed price lies between \$25,000 and \$50,000 can be made. In addition to those scenarios from everyday life, negotiations may also take place at a larger scale, e.g., diplomatic negotiations over territory dispute between a number of countries.

In many cases negotiation a time consuming process, and, moreover, finding good agreements often a tough challenge for human negotiators, especially when they are in lack of negotiation experience and the negotiation time is limited [Lin and Kraus (2010)]. To overcome these limitations, there exist considerable interests in automating their negotiation process by means of autonomous agents to assist humans in the decision-making process. In this context, an agent could be either a piece of software (programme) that acts on behalf of a certain party and is able to interact autonomously with other agents (or humans) for reaching an agreement on a range of issues that are of conflicting interest to each involved party [Wooldridge and Jennings (1995)]. The negotiations operated by au-

tonomous agents are so-called automated negotiations. In contrast with human-based negotiations, automated negotiations enable further complicated negotiation forms because agents have stronger computational power and better memory capability [Williams (2012)]. Recent years have witnessed a rapidly growing interest in automated negotiations, mainly due to the broad application range in fields as diverse as electronic commerce and electronic markets [Lau *et al.* (2008); Ragone *et al.* (2008)], supply chain management [Wang *et al.* (2009)], task and service allocation [Dang and Huhns (2006)], and combinatorial optimization [Duan *et al.* (2012)].

Negotiation can be carried out in a number of ways and forms, such as single-issue or multi-issue, bilateral or multi-lateral negotiations. Several different research trends can be distinguished in automated negotiation models. For example, game-theoretic models consider a negotiation to be a game played by two or more players with a set of actions. Reaching optimal solutions are their main focus under assumptions of full rationality, complete (or partial) information regarding the strategies and preferences of other parties. By contrast, heuristic models aim at searching for solutions to be as close as possible to the optimum, instead of obtaining optimum agreements. Thus, heuristic models assume imperfect knowledge regarding the opponent and the environment, and provide computationally affordable solutions in order to achieve good results. In the following, forms, protocols and main approaches of automated negotiation are overviewed, respectively.

1.3.1 *Negotiation forms*

Due to its diverse applications, automated negotiation could come in a range of forms [Lomuscio *et al.* (2003); Lopes *et al.* (2008)]. The typical (and significant) types are listed as follows:

1.3.1.1 *Single-issue versus Multi-issue Negotiations*

In single-issue negotiations there is only one issue under consideration for negotiators. To a large extent, it reduces the negotiation complexity since learning opponent in the single-issue case is relatively simple as the loss of one side becomes the gain of the other side. However, the complexity of the negotiation problems increase rapidly as the number of issues grows (i.e., more complex and larger outcome space).

Agent mediated negotiation with multiple issues places a high demand on agents' capabilities of choosing proper negotiation moves. In such sort of negotiations, there are likely to be a number of proposals, which are

at the same utility level from one side's perspective but different from the other side's perspective. The agents, therefore, are expected to find more of those proposals that are mutually beneficial for both parties in order to increase the quality and efficiency of the negotiation.

1.3.1.2 *Bilateral versus Multi-lateral Negotiations*

Bilateral negotiation normally refers to the scenario in which two parties negotiate with each other to search for a mutually acceptable agreement between them. The progress of bilateral negotiations is relatively easy to assess because its dominant form of communication can be regarded as offer exchanges between the two parties.

By contrast with bilateral negotiation, multi-lateral negotiation deals with conflict among multiple parties that sit around the table. The efficiency of multi-lateral negotiations tends to be too low since multi-lateral negotiations are time-consuming process and even need decades to finish (e.g., the negotiation over China's participation in the World Trade Organization). As a matter of fact, multi-lateral negotiations are far more complex than bilateral ones due to the complicated communication mechanisms, the great variety of interests at stake, the many variables involved in the process and so forth. Therefore, little work has been done in the area of multi-lateral negotiations, and most research efforts still focus on bilateral negotiations so far.

1.3.1.3 *Sequential versus Concurrent Negotiations*

When it comes to the negotiation problem that one party needs to negotiate with multiple parties on same items or services, two main options are available for them: sequential or concurrent negotiations [Ponka (2009)]. For sequential negotiations, a negotiator negotiates sequentially with others, e.g., all the providers of a service; while in the form of concurrent negotiations, a negotiator negotiates with many or all the providers at the same time. Concurrency has the advantage that participants are allowed to interact with more opponents at a time, thus boosting their learning performance and making the high flexibility of decision-making possible (e.g., altering their negotiation tactics in some threads according what they learn from others or seeking for proposals with a better profit given an agreement has been reached in one thread). This feature, on the other hand, gives rise to the management problem of coordinating multiple concurrent negotiating threads, which requires additional computational resources, and gets

even worse when concurrent negotiation takes place on a many-to-many basis. In contrast, sequential negotiations are much easier to study and analyze mathematically. The good understanding of sequential negotiations paves the way for building further complicated negotiation forms; therefore in much existing work focus is still on sequential negotiations.

1.3.1.4 *Complete versus Incomplete information*

In many cases, information about the opponent is often unknown. Specifically, the opponent's utility function, negotiation strategy and reservation value may be unknown, as revealing such information would lead to exploitation of its behaviors. In a negotiation with incomplete information, it is a real challenge for an agent to know exactly how its actions affect the opponent, as this would depend on information that is unknown. By contrast, if an agent acquires complete information about its opponent, the agent can then determine which outcomes are high performing for both parties. It is clear that the agents in this context are more likely to find a good solution for both sides, and learning opponents (e.g., their utility function, negotiation strategy and the like) is not that important any more.

To summarize, existing literature focuses on automated bilateral sequential multi-issue incomplete-information negotiation (e.g., [Baarslag *et al.* (2013); Chen *et al.* (2013a); Chen and Weiss (2012, 2014); Lai *et al.* (2004); Williams *et al.* (2011)]), which is also common for real-world negotiation scenarios. Characteristic to this type is that there are two autonomous agents exchanging offers in turn with the goal to agree on a common set of issues such as price, delivery time, quantity and quality, where these issues may be of conflictive importance for the negotiators.

1.3.2 *Negotiation protocol*

Negotiation protocol defines the basic rules of the whole negotiation stages, including the types of participants, the negotiation states, the actions that cause negotiation states to change and the actions that participants can make in each state [Jennings *et al.* (2001)]. In simple terms, negotiation protocol governs who can be involved in the negotiation as well as how the negotiation should proceed. Some good examples of them include simultaneous offers and alternating offers protocols.

1.3.2.1 *Simultaneous Offers*

Under the simultaneous offers negotiation protocol, the parties on the table make their offers at the same time. The Nash demand game [Nash (1953)] is a single shot simultaneous game, where the two parties simultaneously make a single offer. Consider a concrete negotiation as an example. There are two players negotiating over how to share a single cheesecake. Players p and q make a single offer each, $O_p \in [0,1]$ and $O_q \in [0,1]$, which indicates how much of the pie they wish to eat. If the total amount of pie the players specified in their offers did not exceed one (that is to say $(O_p + O_q) \leq 1$), then an agreement is reached between them. Otherwise, if $O_p + O_q > 1$, no agreement is reached, and the negotiation ends in conflict (i.e., negotiation failure). However, it is quite possible that a reached agreement is not efficient, because some of the cake maybe remain unallocated (e.g., $O_p + O_q < 1$). These inefficient outcomes that are not Pareto-optimal¹ fail to maximize the joint expectation of the participating parties.

1.3.2.2 *Alternating Offers*

The parties, according to the alternating offers protocol, in turn make offers and counter-offers. This continues until one of the parties accepts the opponent's offer, or alternatively, one of the parties chooses to terminate the negotiation (without agreement), or the negotiation deadline is reached. Compared to the simultaneous offers protocol, the alternating offers protocol is more appropriate for automated negotiation situations where the negotiators are assumed to only have incomplete information most of the time. This is because this protocol could enforce the offers of the two parties compatible and even Pareto-efficient without a costly coordination mechanism. Therefore, automated negotiation strategies tend to be on the basis of the alternating offers protocol.

1.3.3 *Negotiation approaches*

Although fruitful work on automated negotiation has been contributed to the community of multi-agent systems, automated negotiation is generally solved in the following three ways by researchers: heuristic, game-theoretic and argumentation models.

¹An outcome is considered to be Pareto-optimal, if there are no other outcomes that a single participant could select with no other participants objecting [Wooldridge (2009)].

1.3.3.1 Heuristic Approaches

Negotiation has traditionally been investigated in game theory [Raiffa (1982); Osborne and Rubinstein (1994)], and over previous years it has also developed into a core topic of multiagent systems [Mor *et al.* (1996); Lopes *et al.* (2008); Weiss (2013)]. A number of approaches have been proposed that explore the idea to equip an agent with the ability to build a model of its opponent and to use this model for optimizing its negotiation moves (see [Hendrikx (2011)] for a good overview). Modeling the opponent's behavior, however, is practically challenging because negotiators usually do not reveal their true preferences and/or negotiation strategies in order to avoid that others exploit this information to their advantage [Raiffa (1982); Coehoorn and Jennings (2004)].

Negotiation approaches employ computationally extensive methods for precise modeling and/or make simplifying assumptions about the negotiation settings (i.e., the environment and the opponent behavior). For example, there are approaches assuming that negotiations are simple single-issue and the opponents have a rather simple (e.g., non-adaptive) behavior. In the following, representative model-based negotiation approaches are overviewed.

Some of the approaches learn opponents by estimating preferences or the reservation value. [Faratin *et al.* (2002)] proposed a trade-off strategy to increase the chance of getting own proposals accepted without decreasing the own profit. The strategy applied the concept of fuzzy similarity to approximate the preference structure of the opponent and used a hill-climbing technique to explore the space of possible trade-offs for its own offers most likely to be accepted. The effectiveness of this method highly depended on the availability of prior domain knowledge that allowed to determine the similarity of issue values. [Coehoorn and Jennings (2004)] used Kernel Density Estimation for estimating the issue preferences of an opponent in multi-issue negotiations. Their method assumed that the negotiation history was available and that the opponent employed a time-dependent tactic (i.e., the opponent's concession rate depends on the remaining negotiation time, see, e.g., [Faratin *et al.* (1998)] for details on this kind of tactic). The distance between successive counter-offers was used to calculate the opponent's issue weights and to assist an agent in making trade-offs in negotiation.

Besides, researchers also applied Bayesian learning to automated negotiation. For instance, [Zeng and Sycara (1998)] used a Bayesian learning rep-

resentation and updating mechanism to model beliefs about the negotiation environment and the participating agents under a probabilistic framework; more precisely, they aimed at enabling an agent to learn the reservation value of its opponent in single-issue negotiation. [Lin *et al.* (2008)] was another approach based on Bayesian learning, where the usage of a reasoning model based on a decision-making and belief-update mechanism is proposed to learn the likelihood of an opponent's profile; thereby it is assumed that the set of possible opponent profiles is known as a priori. [Hindriks and Tykhonov (2008)] presented a framework for learning an opponent's preferences by making assumptions about the preference structure and rationality of its bidding process. It is assumed that (i) the opponent starts with optimal bids and then moves towards the bids close to the reservation value, (ii) its target utility can be expressed by a simple linear decreasing function, and (iii) the issue preferences (i.e., issue weights) are obtainable on the basis of the learned weight ranking. Moreover, the basic shape of the issue evaluation functions is restricted to downhill, uphill or triangular. In order to further reduce uncertainty in high-dimensional domains, issue independence is assumed to scale down the otherwise exponentially growing computational complexity. [Oshrat *et al.* (2009)] developed a negotiating agent for effective multi-issue multi-attribute negotiations with both human counterparts and automated agents. The successful negotiation behavior of this agent is, to a large extent, grounded in its general opponent modeling component. This component applies a technique known as Kernel Density Estimation to a collected database of past negotiation sessions for the purpose of estimating the probability of an offer to be accepted, the probability of the other party to propose a bid, and the expected averaged utility for the other party. The estimation of these values plays a central role in the agent's decision making. While the agent performs well, the approach taken is not suited for the type of negotiation we are considering (real-time, no prior knowledge, etc.) because opponent modeling is done offline and requires knowledge about previous negotiation traces.

In addition to learning opponents' preferences, other existing approaches try to learn the negotiation strategy and decision model of the opposing negotiator. For instance, [Saha *et al.* (2005)] applied Chebychev's polynomials to estimate the chance that an opponent accepts an offer relying on the decision history of its opponent. This work dealt with single-issue negotiation, where an opponent's response can only be an accept or a reject. [Brzostowski and Kowalczyk (2006)] investigated online prediction of future counter-offers on the basis of the past negotiation exchanges by using

differentials. They assume that there mainly exist two independent factors that influence the behavior of an opposing agent, namely, time and imitation. The opponent is assumed to apply a weight combination of time- and behavior-dependent tactic.² [Hou (2004)] presented a learning mechanism that employs non-linear regression to predict the opponent's decision function in a single-issue negotiation setting. Thereby it was assumed that the opponent behavior could only be time-, behavior- or resources-dependent (with decision functions as proposed in [Faratin *et al.* (1998)]). In the work of [Carboneau *et al.* (2008)] an artificial neural network (ANN) was constructed with three layers that contain 52 neurons to model a negotiation process in a specific domain. The network exploited information about past counter-offers to simulate future counter-offers of opponents. The training process required a very large sample database and therefore became ineffective when applied in online mode.

1.3.3.2 Game Theoretic Approaches

Game theory is the study of strategic behaviors and interaction among autonomous agents. It has been applied to disciplines as diverse as economics, biology, linguistics and computer sciences [Shoham and Leyton-Brown (2009)]. Basically, there are two kinds of game theoretic approaches to negotiation – cooperative and non-cooperative game theory. Cooperative game theory considers games in which it is possible for participants to form coalitions, or in other words the basic unit is a group of agents – so that they together achieve a greater joint utility than they would obtain if they played the game alone. Non-cooperative game theory [Binmore (1992)] instead considers the strategies that can be used by the self-interested negotiation individuals (for maximizing their own utility) and the protocols used in negotiation (for guaranteeing some desirable properties, e.g., guaranteed success, simplicity, individual rationality, stability, etc.).

In a particular agent-based negotiation scenario, game theoretic techniques can help resolve two key problems [Jennings *et al.* (2001)] : 1) the design of an appropriate protocol that will govern the interactions between the negotiation participants; and 2) the design of a particular strategy that individual agents can use while negotiating – an agent will aim to use a strategy that maximizes its own individual welfare. While proven to be useful and helpful, there are a number of shortcomings associated with the

²The concepts of time-dependent and behavior-dependent tactics were introduced in [Faratin *et al.* (1998)].

application of game theory to automated negotiations. One major problem of the theory is related to the strong assumptions about the settings. For instance, game theory assumes that it is possible to characterize an agent's preferences with respect to possible outcomes. However, finding or defining humans' preferences over outcomes are challenging tasks in practice. Furthermore, game theory approaches often assume unbounded computational capability, which means that no computational/time cost needs to be considered in search of mutually accepted solutions within a ranges of outcomes, even when the outcome space is extremely large. In addition, the theory has failed to generate a general model governing rational choice in inter-dependent situations [Zeng and Sycara (1997)].

Despite these obvious problems, game theory can characterize the solution space of negotiation. One well-known and important concept is Nash Equilibrium, where neither participant can benefit by choosing an alternative action, given that all other participants do not change their action [Nash *et al.* (1950)]. For instance, as regards efficiency of a specific solution, Pareto optimality serves as a good indicator – an outcome is said to be Pareto optimal if no negotiation participant can be better off without making at least one other participant worse off. There are also some fairness solution concepts from game theory such as the utilitarian [Myerson (1981)], kalai-smorodinsky [Kalai and Smorodinsky (1975)] considering the fairness of proposed solutions.

1.3.3.3 *Argumentation*

Argumentation-based negotiations provide additional information exchange between negotiators other than the trading of complete (or partial) proposals. This additional information exchanged between agents can be of many different forms, but all of which are arguments which explicitly the reason why an agent makes the argument. As such, a major advantage of exchanging extra information is to enable agents to offer a critique of the received proposal as well as explanations why and where it is not satisfying, rather than simply rejecting the proposal. On the other hand, it is likely for an agent to persuade its negotiation partner with certain arguments to accept a solution by altering its preferences.

In negotiation allowing argumentation, participants benefit from this communication mechanism so that they might reach agreements to which other approaches could not lead. This is however at the expense of significant overheads, due to the reasoning process that the agent has to conduct

in order to construct and evaluate the arguments. Moreover, it is also possible for agents to make arguments that are not truthful, which further complicates the negotiation, since the agent needs to evaluate each argument's credibility [Jennings *et al.* (2001)]. Lacking of suitable argumentation protocols, building fully functional agents capable of handling practical negotiations are not well addressed so far. Besides, it would be difficult to evaluate the performance of an agent in an environment with varying preferences.

1.4 Characters of Complex practical negotiation

The settings of complex negotiations are of relevance to a wide range of practical applications and are also common to many human-human negotiation scenarios. Complex negotiations thus differentiate itself from simple negotiations (e.g., single-issue negotiations, negotiations with prior opponent information, etc) through the following ways:

1.4.1 *Zero Prior Opponent Knowledge*

In many negotiations, agents are allowed to access some knowledge of the opponents before interacting with them. The information may be vital for decision-making process, including opponents' utility function, negotiation strategy and time constraints. Moreover, in a negotiation with complete information about opponents, an agent is able to calculate which proposals satisfy both sides best. It is therefore much easier for the agents to propose and reach a good agreement. Due to the conflicting nature of negotiations (at least for self-interested participants), opponents however tend to not be willing to reveal this information. In a negotiation where an agent has zero prior knowledge about its opponent, it is hard for it to know exactly which proposals could reach a high level of satisfaction and how the opponent would react to its proposals, as this would depend on opponent information, e.g., preference over negotiation issues, strategies, etc. It thus requires agents to acquire useful opponent information in negotiation via machine learning or other related techniques.

1.4.2 *Continuous-time Constraints*

For most cases there is always a deadline by which negotiations should be completed. The constraints of negotiation are a limited amount of

continuous-time in many real-life scenarios. The continuous-time constraints refer to constraints based on the amount of physical elapsed time rather than the number of interactions (or rounds) that is studied in previous literature. Negotiators in this context have a real-time deadline to reach an agreement; otherwise, the negotiation ends up with a failure (this deadline is denoted by t_{max} afterwards). For instance, in a car sales scenario, the customer needs to make his purchase prior to a specific date. In order to force agreements to be reached in a timely manner, real-time constraints are normally used between the customer and the car dealer.

Due to the effect of the continuous-time constraints, negotiating agents are needed to be more adaptive to opponent strategy [Williams (2012)]. That is to say, an agent should not concede too quickly to its opponent, unless the opponent is willing to make a similar compromise. On the other hand, it is also important that an agent does not employ a tough approach (like Boulware strategy [Faratin *et al.* (2002)]), as this could delay the negotiation unnecessarily, causing the value of an agreement to be decreased.

Furthermore, in such contexts, an increased delay grows the likelihood that the agents will fail to reach an agreement. It is therefore a challenge to develop an agent-based strategy that will maximize the agent's gains. Using constraints based on real time makes it more difficult to predict the utility of an offer that an agent might make in the future (other than during the current step). This is because the time at which any future offer will be made by an agent depends on the time that its opponent spends in making its offers. For the same reason, it is also impossible to know how many interactions will occur before the deadline is reached. Therefore, another party influences an agent's utility by delaying the negotiation.

1.4.3 Discounting Effect and Reservation Value

The discounting effect is used as an indicator of a cost based on the duration of the negotiation, with the purpose of encouraging agents to complete a negotiation as soon as possible. It has the effect that any particular agreement that is reached at a given time is of higher value than that same agreement at a later time. This causes agents that are slow to reach agreement (e.g. intentionally exploiting opponents) to be punished, as the value of the agreement will be reduced due to the time delay [Williams (2012); ana (2012)].

The reservation utility is the minimum utility level for which an agent

can expect. It also specifies the maximum concession for one side when playing against a non-concessive opponent. Such maximum concession on the other hand restricts the possible exploitation of the opponent, thereby placing higher demand on the flexibility of automated strategy. We next formalize the two factors, beginning with introducing the model of complex negotiation used in existing work.

Let $I = \{a, b\}$ be a pair of negotiating agents, i represent a specific agent ($i \in I$), J be the set of issues under negotiation, and j be a particular issue ($j \in \{1, \dots, n\}$ where n is the number of issues). The goal of a and b is to establish a contract for a product or service. Thereby a contract consists of a package of issues such as price, quality and quantity. Each agent has a minimum payoff as the outcome of a negotiation; this is called the reservation value ϑ . Further, w_j^i ($j \in \{1, \dots, n\}$) denotes the weighting preference which agent i assigns to issue j . The issue weights of an agent i are normalized summing to one (i.e., $\sum_{j=1}^n (w_j^i) = 1$). During a negotiation session agent a and b act in conflictive roles that are specified by their preference profiles. In order to reach an agreement they exchange offers (i.e., O) in each round to express their demands. An offer is thereby a vector of values, with one value for each issue. The utility of an offer for agent i is obtained by the utility function defined as:

$$U^i(O) = \sum_{j=1}^n (w_j^i \cdot V_j^i(v_{jk})) \quad (1.1)$$

where v_{jk} is the k -th value of the issue j and V_j^i is the evaluation function for agent i , mapping a value of issue j (e.g., v_{jk}) to a real number.

If no agreement can be reached at the end or one side breaks off before deadline, the negotiation then ends up with the disagreement solution (and thus obtain a reservation value ϑ as its payoff). Note that the number of remaining rounds are not known and the outcome of a negotiation depends crucially on the time sensitivity of the agents' negotiation strategies. This holds, in particular, for discounting domains, in which the utility is discounted with time. Discounting factor δ ($\delta \in [0, 1]$) is used to calculate the discounted utility as follows:

$$D^\delta(U, t) = U \cdot \delta^t \quad (1.2)$$

where U is the (original) utility and t is the standardized time (i.e., $t \in [0, 1]$). As an effect, the longer it takes for agents to come to an agreement the lower is the utility they can achieve. Note that a decrease in δ increases the discounting effect.

1.5 State of the art

The focus of current research is on complex practical negotiations. One of the biggest driving forces behind research into complex negotiations is the broader spectrum of applications over simple negotiations. This emerging research trend in automated negotiations has brought a substantial body of work such as [Williams *et al.* (2011); Hao and Leung (2012); Chen *et al.* (2013a,b); Chen and Weiss (2013); Chen *et al.* (2015)] that can be roughly classed into two groups. One group is based on regression techniques to learn opponents and adjust its behaviors through the learnt model, while the other is transferring knowledge between different tasks and agents creating new sources to aid learning opponents. The remainder of this section illustrates each of them by introducing a typical agent.

1.5.1 *Agents based on regression techniques*

OMAC is a good example [Chen and Weiss (2013)] that models opponents using regression technique. It includes two core stages – opponent modeling and decision-making mechanism. Regression analysis of opponents is done by discrete wavelet transformation (DWT), which is a type of multi-resolution wavelet analysis that provides a time-frequency representation of a signal and, based on this, it is capable of capturing time localizations of frequency components. DWT has become increasingly important and popular as an efficient multi-scaling tool for exploring features since it can offer with modest computational effort high-quality solutions (with complexity $O(n)$) to non-trivial problems such as feature extraction, noise reduction, function approximation and signal compression. OMAC employs DWT to extract the main trend of the opponent's concession over time from its previous counter-offers.

In DWT a time-frequency representation of a signal is obtained through digital filtering techniques, where two sets of functions are utilized: scaling functions using low-pass filters and wavelet functions using high-pass filters. More precisely, a signal is passed through a series of high pass filters to analyze the high frequencies, and similarly it is passed through a series of low pass filters to analyze the low frequencies. In so doing, DWT decomposes a signal into two parts, an approximation part and a detailed part. The former is smooth and reveals the basic trend of the original signal, and the latter is rough and in general corresponds to short-term noise from the higher-frequency band.

The decomposition process can be applied recursively as follows:

$$\begin{aligned} y_{low}[k] &= \sum_n f[n] \cdot h[2k - n] \\ y_{high}[k] &= \sum_n f[n] \cdot g[2k - n] \end{aligned} \quad (1.3)$$

with $f[n]$ being the signal, $h[n]$ a halfband high-pass filter, $g[n]$ a halfband low-pass filter, and $y_{low}[k]$ and $y_{high}[k]$ the outputs of the low-pass and high-pass filters, respectively. The iterative application of DWT results in different levels of detail of the input signal; in other words, it decomposes the approximation part into a “further smoothed” component and a corresponding detail component. The further smoothed component contains longer period information and provides a more accurate trend of the signal. For instance, f can firstly be decomposed into a rough smooth part (a_1) and a detail part (d_1), and then the resulting part a_1 can be decomposed in finer components, that is, $a_1 = a_2 + d_2$, and so on. This iterative process is captured by the below diagram:

$$\begin{array}{ccccccc} f & \cdots & a_1 & \cdots & a_2 & \cdots & a_3 \cdots a_n \\ & \ddots & & \ddots & & \ddots & \\ & & d_1 & & d_2 & & d_3 \cdots d_n \end{array}$$

where a_1, a_2, \dots, a_n are the approximation parts and d_1, d_2, \dots, d_n are the detail parts of f .

Given the discrete wavelet function $\psi_{j,k}(t)$ transformed by a mother wavelet $\psi(t)$,

$$\psi_{j,k}(t) = a_0^{-j/2} \psi(a_0^{-j} t - kb_0), \quad j, k \in Z \quad (1.4)$$

DWT corresponds to a mapping from the signal $f(t)$ to coefficients $C_{j,k}$ which are related to particular scales, where these coefficients are defined as follows:

$$C_{j,k} = \int_{-\infty}^{+\infty} f(t) \overline{\psi_{j,k}(t)} dt, \quad j, k \in Z \quad (1.5)$$

The $\psi(t)$ is normally required to be an orthogonal wavelet in practice, the set $\{\psi_{j,k}(t) | j, k \in Z\}$ is then an orthogonal wavelet basis such that the signal $f(t)$ can be reconstructed.

A concrete example of applying DWT in negotiation is given in Fig. 1.1, which shows the curve of the received utilities (i.e., the original signal) in the domain *Airport site selection* when negotiating with the agent IAMhagler2011. The curve at the top of the figure represents received utilities χ ,

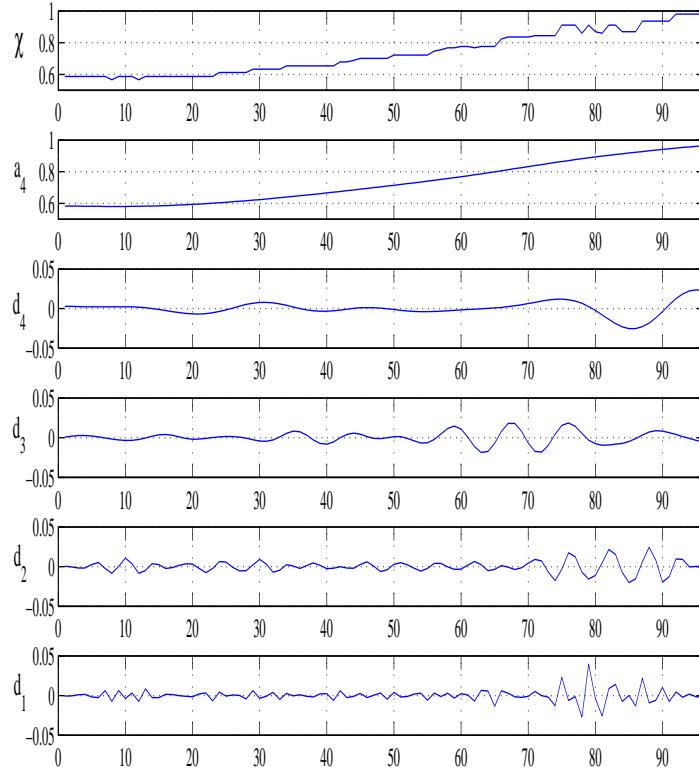


Fig. 1.1 4-level decomposition of the utilities series χ obtained from IAMhaggler2011 in the negotiation domain *Airport site selection*. The vertical axis shows score and the horizontal axis represents the percentage of time (%). The 4-level DWT decomposes the original input χ into five sub-components including $d_1 \dots d_4$ (detail parts) and a_4 (approximation part). The resulting components are shown below χ , their frequency increases from d_1 to d_4 (i.e., they are more and more smooth). The final approximation part – a_4 is the long-term trend of χ .

d_n is the detail component of the n^{th} decomposition layer and a_4 is the approximation on the final layer (i.e., the fourth). This figure clearly shows that a_4 is a pretty good approximation of the main trend of the original signal. As can be seen, the noise/variation represented by those detail components (e.g., d_1 to d_4) is irrelevant to its trend.

With DTW opponent modeling of OMAC is to estimate the utilities of future counter-offers it will receive from its opponent. Opponent modeling is done through a combination of wavelets analysis and cubic smoothing spline. When receiving a new bid from the opponent at the time t_c , the agent records the time stamp t_c and the utility $U(O_{opp})$ this bid has according to the agent's utility function. The maximum utilities in consecutive (equal) time intervals and the corresponding time stamps are used periodically as basis for predicting the opponent's behavior.

With recursive application of DWT to the signal $f(t)$, the approximation (low frequency) and detail (high frequency) components are recovered, respectively. For instance, f can first be decomposed into $a_1 + d_1$ and the resulting part a_1 can then be decomposed in finer components, that is, $a_1 = a_2 + d_2$, and so on. Based upon this recursive process, the signal can be expressed as $f = a_1 + a_2 + \dots + a_n + d_n$ (further details on wavelets are given in e.g. [Daubechies (2006)]). We use the following notation:

$$\chi = v + \sum_{n=1}^{\lambda} d_n \quad (1.6)$$

where v represents the approximation component of χ and d_n is n-layer detail part (n is determined by the decomposition level λ). An example can be found in Figure 1.2 which shows χ and its corresponding approximation part v along with the estimated upper and lower bounds of χ . The two bounds are represented by $v \pm \sigma$, where σ is the standard deviation of the ratio between χ and v .

In order to forecast the opponent's future behavior, cubic smoothing spline is used by OMAC to extend the smooth component v . Cubic spline is widely used as a tool for prediction, see [Yousefi *et al.* (2005)]. For equally spaced time series, a cubic spline is a smoothing piecewise function, denoted as the function $\hat{g}(t)$ which minimizes:

$$p \sum_{t=1}^n w(t)(f(t) - \hat{g}(t))^2 + (1-p) \int (\hat{g}(u)''')^2 du \quad (1.7)$$

where p is the smoothing parameter controlling the rate of exchange between the residual error described by the sum of squared residuals and local variation represented by the square integral of the second derivative of g and w is the weight vector (for further details, refer to [de Boor (1978)]).

Given the extended version of the smooth part – α , a simple function R , called reserved utility function, is used to realize concession adaptation. This function guarantees the minimum utility at each given time step. This

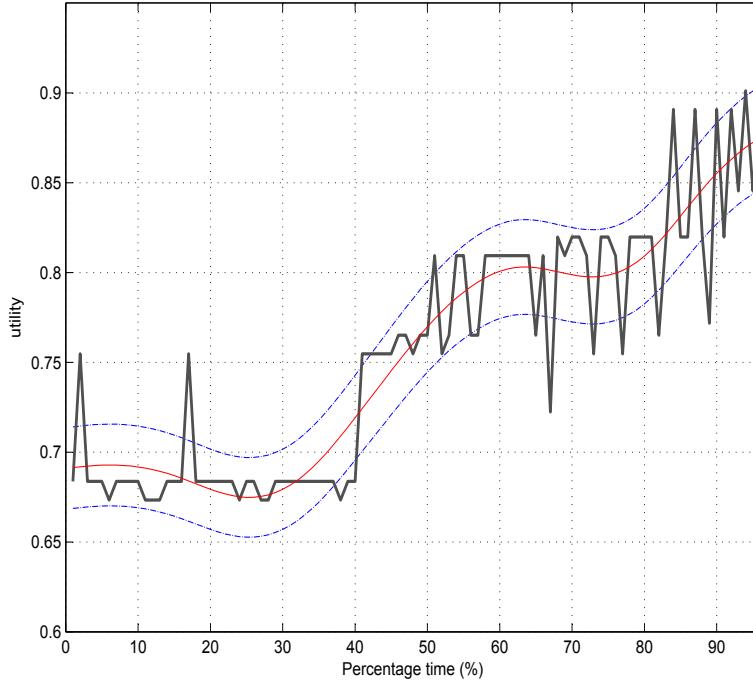


Fig. 1.2 Illustrating the opponent’s concession (given by χ , the thick solid line) and the corresponding approximation part v (the thin solid line) when negotiating with Agent_K2 in the *Camera* domain. The two dash-dot lines represent the estimated upper and lower bounds of χ .

is because the function values are set as the lower bound of our expected utilities. Moreover, in principle it makes concession over time, thereby taking into account the impact of the discounting factor.

The estimated received utility $E_{ru}(t)$, which indicates the expectation of opponent’s future concession, is defined as follows:

$$E_{ru}(t) = D(\alpha(t)(1 + Stdev(ratio_{[t_b, t_c]})), t), \quad t \in [t_c, t_s] \quad (1.8)$$

where $Stdev(ratio_{[t_b, t_c]})$ is the standard deviation of ratio between the smooth part v and the original signal χ from the beginning of negotiation(t_b) till now and t_s is the end of α .

Suppose the future expectation the agent has obtained from $E_{ru}(t)$ is optimistic, in other words, there exists an interval $\{T | T \neq \emptyset, T \subseteq [t_c, t_s]\}$, so that

$$E_{ru}(t) \geq D(R(t), t), \quad t \in T \quad (1.9)$$

OMAC then sets the time \hat{t} at which the optimal estimated utility \hat{u} is reached as:

$$\hat{t} = \operatorname{argmax}_{t \in T} (E_{ru}(t) - D(R(t), t)) \quad (1.10)$$

and \hat{u} is simply assigned by:

$$\hat{u} = E_{ru}(\hat{t}) \quad (1.11)$$

1.5.2 Agents based on transfer learning

Even though autonomous agents have gained remarkable achievements in various negotiation domains, a significant shortcoming shared by autonomous agents is the omission of knowledge reuse. The problem of opponent modeling is tough due to the lack of enough information about opponents. Knowledge re-use in the form of transfer can serve as a potential solution for such a challenge. Transfer is substantially more complex than simply using the previous history encountered by the agent. Since for instance, the knowledge for a target agent can potentially arrive from a different source agent negotiating in a different domain, in which case a mapping to correctly configure such knowledge is needed. Transfer is of great value for a target negotiation agent in a new domain. The agent can use this “additional” information to learn about and adapt to new domains more quickly, thus producing more efficient strategies.

[Chen *et al.* (2013b)] firstly applied this novel transfer learning mechanism to automated negotiation. Specifically, the proposed agent adopts conditional restricted Boltzmann machines (CRBMs) as the basis for opponent modeling, which is in turn used for determining a negotiation strategy. Conditional restricted Boltzmann machines (CRBMs), introduced in [Taylor and Hinton (2009)], are rich probabilistic models used for structured output predictions. CRBMs include three layers: (1) history, (2) hidden, and (3) present layers. These are connected via a three-way weight tensor among them. CRBMs are formalized using an energy function. Given an input data set, these machines learn by fitting the weight tensor such that the energy function is minimized. Although successful in modeling different time series data [Taylor and Hinton (2009); Mnih *et al.* (2012)], full CRBMs are computationally expensive to learn. Their learning algorithm, contrastive divergence (CD), incurs a complexity of $\mathcal{O}(N^3)$. Therefore a factored version, the factored conditional restricted Boltzmann machines (FCRBM), has been proposed in [Taylor and Hinton (2009)]. FCRBM factors the three-way weight tensor among the layers, reducing the complexity to $\mathcal{O}(N^2)$. Next the mathematical details will be explained.

Define $\mathcal{V}_{< t} = [v_{< t}^{(i)}, \dots, v_{< t}^{(n_1)}]$, with n_1 being the number of units in the history layer. Further, define $\mathcal{H}_t = [h_t^{(1)}, \dots, h_t^{(n_2)}]$, with n_2 being the number of nodes in the hidden layer. Finally, define $\mathcal{V}_t = [v_t^{(1)}, \dots, v_t^{(n_3)}]$, with n_3 being the number of units in the present layer.

In automated negotiation the inputs are typically continuous. Therefore, for the history and present layers, a Gaussian distribution is adopted, with a sigmoidal distribution for the hidden.

The visible and hidden units joint probability distribution is given by:

$$p(\mathcal{V}_t, \mathcal{H}_t | \mathcal{V}_{< t}, \mathbf{W}) = \exp\left(\frac{-E(\mathcal{V}_t, \mathcal{H}_t | \mathcal{V}_{< t}, \mathbf{W})}{Z}\right)$$

with the factored energy function determined using:

$$\begin{aligned} E(\mathcal{V}_t, \mathcal{H}_t | \mathcal{V}_{< t}, \mathbf{W}) = & - \sum_i \frac{(v_t^{(i)} - a^{(i)})}{\sigma_i^2} - \sum_j h_t^{(j)} b^{(j)} \\ & - \sum_f \left(\sum_i \mathbf{W}_{if}^{\mathcal{V}_t} \frac{v_t^{(i)}}{\sigma_i} \sum_j \mathbf{W}_{jf}^{\mathcal{H}} h_t^{(j)} \sum_k \mathbf{W}_{kf}^{\mathcal{V}_{< t}} \right) \end{aligned}$$

where Z is the potential function, f is the number of factors used for factoring the three-way weight tensor among the layers, and σ_i is the variance of the Gaussian distribution in the history layer. Furthermore, $\mathbf{W}_{if}^{\mathcal{V}_t}$, $\mathbf{W}_{jf}^{\mathcal{H}}$, and $\mathbf{W}_{kf}^{\mathcal{V}_{< t}}$ are the factored tensor weights of the history, hidden, and present layer, respectively. Finally, $a^{(i)}$ and $b^{(j)}$ are the biases of the history and hidden layers, respectively.

Since there are no connections between the nodes of the same layer, inference is done parallel for each of them. The values of the j^{th} hidden unit and the i^{th} visible unit are, respectively, defined as follows:

$$\begin{aligned} s_{j,t}^{\mathcal{H}} &= \sum_f \mathbf{W}_{jf}^{\mathcal{H}} \sum_i \mathbf{W}_{if}^{\mathcal{V}} \frac{v_t^{(i)}}{\sigma_i} \sum_k \mathbf{W}_{kf}^{\mathcal{V}_{< t}} \frac{v_{< t}^{(k)}}{\sigma_k} + b^{(j)} \\ s_{i,t}^{\mathcal{V}} &= \sum_f \mathbf{W}_{if}^{\mathcal{V}} \sum_j \mathbf{W}_{jf}^{\mathcal{H}} h_t^{(j)} \sum_k \mathbf{W}_{kf}^{\mathcal{V}_{< t}} \frac{v_{< t}^{(k)}}{\sigma_k} + a^{(i)} \end{aligned}$$

These are then substituted to determine the activation probabilities of each of the hidden and visible units as:

$$\begin{aligned} p(h_t^{(j)} = 1 | \mathcal{V}_t, \mathcal{V}_{< t}) &= \text{sigmoid}(s_{j,t}^{\mathcal{H}}) \\ p(v_t^{(i)} = x | \mathcal{H}_t, \mathcal{V}_{< t}) &= \mathcal{N}(s_{i,t}^{\mathcal{V}}, \sigma_i^2) \end{aligned}$$

Learning in the full model means to update the weights when data is available. This is done using persistence contrastive divergence proposed in [Mnih *et al.* (2012)]. The update rules for each of the factored weights are:

$$\begin{aligned}\Delta \mathbf{W}_{if}^{\mathcal{V}} &\propto \sum_t \left(\left\langle v_t^{(i)} \sum_k \mathbf{W}_{kf}^{\mathcal{V}_{<t}} v_{<t}^{(k)} \sum_j \mathbf{W}_{jf}^{\mathcal{H}} h_t^{(j)} \right\rangle_0 \right. \\ &\quad \left. - \left\langle v_t^{(i)} \sum_k \mathbf{W}_{kf}^{\mathcal{V}_{<t}} v_{<t}^{(k)} \sum_j \mathbf{W}_{jf}^{\mathcal{H}} h_t^{(j)} \right\rangle_K \right) \\ \Delta \mathbf{W}_{jf}^{\mathcal{H}} &\propto \sum_t \left(\left\langle h_t^{(j)} \sum_k \mathbf{W}_{kf}^{\mathcal{V}_{<t}} v_{<t}^{(k)} \sum_i \mathbf{W}_{if}^{\mathcal{V}} v_t^{(i)} \right\rangle_0 \right. \\ &\quad \left. - \left\langle h_t^{(j)} \sum_k \mathbf{W}_{kf}^{\mathcal{V}_{<t}} v_{<t}^{(k)} \sum_i \mathbf{W}_{if}^{\mathcal{V}} v_t^{(i)} \right\rangle_K \right) \\ \Delta \mathbf{W}_{kf}^{\mathcal{V}_{<t}} &\propto \sum_t \left(\left\langle v_{<t}^{(k)} \sum_i \mathbf{W}_{if}^{\mathcal{V}_t} v_t^{(i)} \sum_j \mathbf{W}_{jf}^{\mathcal{H}} h_t^{(j)} \right\rangle_0 \right. \\ &\quad \left. - \left\langle v_{<t}^{(k)} \sum_i \mathbf{W}_{if}^{\mathcal{V}_t} v_t^{(i)} \sum_j \mathbf{W}_{jf}^{\mathcal{H}} h_t^{(j)} \right\rangle_K \right) \\ \Delta a^{(i)} &\propto \sum_t (\langle v_t^{(i)} \rangle_0 - \langle v_t^{(i)} \rangle_K) \\ \Delta b^{(j)} &\propto \sum_t (\langle h_t^{(j)} \rangle_0 - \langle h_t^{(j)} \rangle_K)\end{aligned}$$

where $\langle \cdot \rangle_0$ is the data distribution expectation and $\langle \cdot \rangle_K$ is the reconstructed distribution after K-steps sampled through a Gibbs sampler from a Markov chain starting at the original data set.

Upon receiving a new counter-offer from the opponent, the agent records the time stamp t_c , the utility of the latest offer proposed by the agent $u_{own}^{(l)}$ and the utility, u_{rec} , in accordance with the agent's own utility function. Since the agent may encounter a large amount of data in a single session when operating in real time, the CRBM is trained every short period (interval) to guarantee its robustness and effectiveness in the real-time environments. The number of equal intervals is denoted by ζ .

The agent predicts the optimal utility to receive from the opponent, u_{rec}^* , by training the conditional restricted Boltzmann machine with $(t_c, u_{own}^{(l)})$ as inputs.

To maximize the received utility from the opponent, the agent should select the optimal offer that maximizes:

$$\max_{t, u_{own}} u_{rec} = \max_{t, u_{own}} \mathcal{N}(\mu, \sigma^2) \quad (1.12)$$

$$\text{s.t. } t \leq T_{\max} \quad (1.13)$$

with, $\mu = \sum_{ij} \mathbf{W}_{ij1} v^{(i)} h^{(j)} + c^{(1)}$ being the mean of the Gaussian distribution of the node in the present layer of the Boltzmann machine. In other words, the output of the CRBM on the present layer is the predicted utility, u_{rec} , of the opponent. To solve the maximization problem of Equation 1.13, the Lagrange multiplier technique is used. Namely, the above problem is transformed to the following:

$$\max_{t, u_{own}} J(t, u_{own}) \quad (1.14)$$

where $J(t, u_{own}) = [\mathcal{N}\left(\sum_{ij} \mathbf{W}_{ij1} v^{(i)} h^{(j)} + c^{(1)}, \sigma^2\right) - \lambda(T_{\max} - t)]$. The derivatives of Equation 1.14 with respect to t and u_{own} are calculated as follows:

$$\begin{aligned} \frac{\partial}{\partial t} J(t, u_{own}) &= \frac{1}{2\sigma^2} (u_{rec} - \mu) \mathcal{N}(\mu, \sigma^2) \left[\sum_j \mathbf{W}_{1j1} h^{(j)} \right] \\ \frac{\partial}{\partial u_{own}} J(t, u_{own}) &= \frac{1}{2\sigma^2} (u_{rec} - \mu) \mathcal{N}(\mu, \sigma^2) \left[\sum_j \mathbf{W}_{2j1} h^{(j)} \right] \end{aligned}$$

These derivatives are then used by gradient ascent to maximize Equation 1.14. The result is a point $\langle t^*, u_{own}^* \rangle$ that corresponds to u_{rec}^* .

Having obtained $\langle t^*, u_{own}^* \rangle$, the agent has now to decide on how to concede to t^* from t_c . [Williams *et al.* (2011)] adopted a linear concession strategy. However, such a scheme potentially omits a lot of relevant information about the opponent's model. To overcome such a shortcoming, this work uses the functional (i.e., learnt by the CRBM) manifold in order to determine the concession rate over time. First, a vector of equally spaced time intervals between t^* and t_c is created. This vector is passed to the CRBM to predict a vector of relevant u_{rec} 's (i.e., phase one in Figure 1.3). Having attained u_{rec} , the machine runs backwards, denoted by phase two in Figure 1.3, (i.e., CD with fixing the present layer and reconstructing on the input) to find the optimal utility to offer at the current time t_c . It is clear that the concession rate follows the manifold created by the factored three-way weight tensor, and thus follows the surface of the learnt function. Adopting the above scheme, the agent potentially reaches $\langle t^*, u_{own}^* \rangle$, which are used to attain u_{rec}^* such that the expected received utility can be maximized.

Given a utility (u_τ) to offer, the agent needs to validate whether the utility of the counter-offer, u_{rec} , is better than u_τ and the (discounted) reservation value, or whether it had been already proposed earlier by the

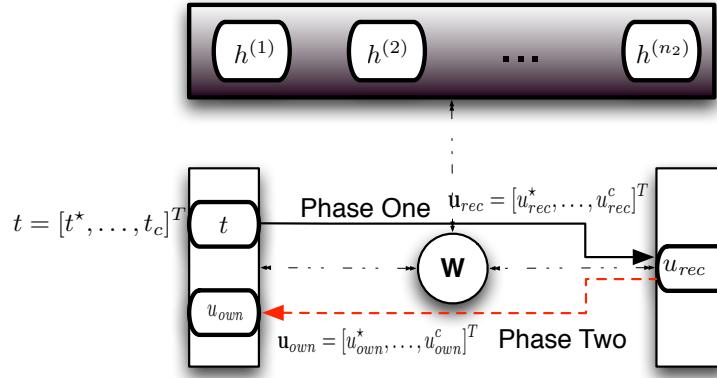


Fig. 1.3 High level schematic of the overall concession rate determination.

agent. If either of the conditions is met, the agent accepts this counter-offer and an agreement is reached. Otherwise, the proposed method constructs a new offer which has a utility of u_τ . Furthermore, for negotiation efficiency, if u_τ drops below the value of the best counter-offer, the agent chooses that best counter-offer as its next offer, because such a proposal tends to well satisfy the expectation of the opponent, which will then be inclined to accept it.

Knowledge transfer has been an essential integrated part of different machine learning algorithms. The idea behind transferring knowledge is that a target agents when faced by new and unknown domains, may benefit from the knowledge gathered by other agents in different source domain(s). Opponent modeling in complex negotiations is a tough challenge due to the lack of enough information about the opponent. Transfer learning is a well suited potential solution for such a problem.

The overall transfer mechanism of *TCRMB* is shown in Figure 1.4. After learning in a source task, this knowledge is then mapped to the target. The mapping is manifested by the connections between the two tasks' hidden layers (Figure 1.4). This can be seen as an initialization of the target task machine using the source model. The intuition behind this idea is that even though the two tasks are different, they might have shared features. This relation is discovered using the weight connections between the hidden layers shown in Figure 1.4. To learn these weight connections contrastive divergence on the corresponding layers is performed. The weights are learnt

such that the reconstruction error between these layers is minimized. It is worth noting, that there is no need for the target hidden units to be same as these of the source. The following update rules for learning the weights are used:

$$\mathbf{W}_{ij}^{\mathcal{S} \rightarrow \mathcal{T}(\tau+1)} = \mathbf{W}_{ij}^{\mathcal{S} \rightarrow \mathcal{T}(\tau)} + \alpha \left(\langle h_{\mathcal{S}}^{(i)} h_{\mathcal{T}}^{(j)} \rangle_0 - \langle h_{\mathcal{S}}^{(i)} h_{\mathcal{T}}^{(j)} \rangle_K \right)$$

where, $\mathbf{W}^{\mathcal{S} \rightarrow \mathcal{T}}$ is the weight connection between the source negotiation task, \mathcal{S} , and the target task \mathcal{T} , τ is the iteration step of contrastive divergence, α is the learning rate, $\langle \cdot \rangle$ is the data distribution expectation and $\langle \cdot \rangle_K$ is the reconstructed distribution after K-steps sampled through a Gibbs sampler from a Markov chain starting at the original data set.

When the weights are attained, an initialization of the target task hidden layer using the source knowledge is available. This is used at the start of the negotiation to propose offers for the new opponent. The scheme in which these propositions occur is the same as explained previously in the no transfer case. When additional target information is gained, it is used

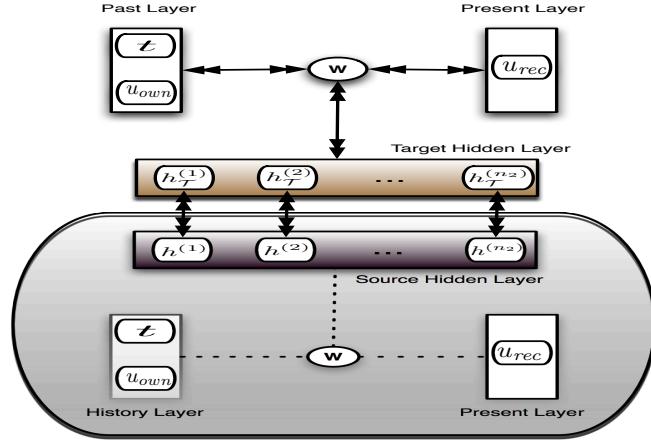


Fig. 1.4 Transfer CRBMs.

to train the target CRBM as in the normal case. It is worth noting, that this proposed transfer method is not restricted to one source task. On the contrary, multiple source tasks are equally applicable. Furthermore, such a transfer scheme is different than just using the history of the agent. The transferred knowledge can arrive from any other task as long as this infor-

mation is mapped correctly to suit the target. This mapping is performed using the CD algorithm as described above.

The difference between *TCRMB* and *CRMB*, lie in the initialization procedure of the three-way weight tensor at the start of the negotiation. In case of *TCRMB* the initialization is performed using the transferred knowledge from the source task, while in *CRMB* this is done using the standard method, where the weights are sampled from a uniform Gaussian distribution with a mean and standard deviation determined by the designer.

1.6 Conclusion

This chapter begins with introduction of the popular forms, protocols and approaches of automated negotiation. Then state-of-the-art approaches for complex practical negotiation – multi-issue negotiation that runs under real-time constraints and in which the negotiating agents have no prior knowledge about their opponents’ preferences and strategies – are studied by two fine examples of agents based on regression techniques and agents based on transfer learning. These two types of techniques are used wildly in the field of complex negotiation with good performance across a variety of scenarios.

Despite of the successes gained by those agents, two issues still stand out. First, the computational complexity of learning opponents is still not trivial especially when tackling high-dimensional problems. One possible solution is to further reduce computational complexity via adopting more advanced schemes (e.g., sparse techniques). Another issue is how to extend the proposed agents to other more complex negotiation forms like concurrent multilateral negotiations. As far as we are concerned, there is little research focus on the applicability and generality of negotiating agents.

Acknowledgements

This research is supported by National Natural Science Foundation of China (Program number: 61602391), and is also supported by Southwest University and Fundamental Research Funds for the Central Universities (Grant number: SWU115032, XDKJ2016C042).

Bibliography

- (2012). The Third International Automated Negotiating Agent Competition (ANAC 2012), <http://anac2012.ecs.soton.ac.uk/results/final/>.
- Baarslag, T., Fujita, K., Gerding, E. H., Hindriks, K., Ito, T., Jennings, N. R., Jonker, C., Kraus, S., Lin, R., Robu, V., and Williams, C. R. (2013). Evaluating practical negotiating agents: Results and analysis of the 2011 international competition, *Artificial Intelligence* **198**, pp. 73 – 103.
- Binmore, K. (1992). Fun and games, a text on game theory, .
- Brzostowski, J. and Kowalczyk, R. (2006). Predicting partner's behaviour in agent negotiation, in *Proceedings of the Fifth Int. Joint Conf. on Autonomous Agents and Multiagent Systems* (ACM, New York, NY, USA), ISBN 1-59593-303-4, pp. 355–361.
- Carboneau, R., Kersten, G. E., and Vahidov, R. (2008). Predicting opponent's moves in electronic negotiations using neural networks, *Expert Syst. Appl.* **34**, pp. 1266–1273.
- Chen, S., Ammar, H. B., Tuyls, K., and Weiss, G. (2013a). Optimizing complex automated negotiation using sparse pseudo-input Gaussian processes, in *Proceedings of the 12th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems* (IFAAMAS), pp. 707–714.
- Chen, S., Ammar, H. B., Tuyls, K., and Weiss, G. (2013b). Using conditional restricted boltzmann machine for highly competitive negotiation tasks, in *Proceedings of IJCAI'2013* (AAAI Press), pp. 69–75.
- Chen, S., Hao, J., Weiss, G., Zhou, S., and Zhang, Z. (2015). Toward efficient agreements in real-time multilateral agent-based negotiations, in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 896–903, doi:10.1109/ICTAI.2015.130.
- Chen, S. and Weiss, G. (2012). An efficient and adaptive approach to negotiation in complex environments, in *Proceedings of the 20th European Conference on Artificial Intelligence* (IOS Press), pp. 228–233.
- Chen, S. and Weiss, G. (2013). An efficient automated negotiation strategy for complex environments, *Engineering Applications of Artificial Intelligence* **26**, 10, pp. 2613 – 2623, doi:<http://dx.doi.org/10.1016/j.engappai.2013.08.012>, <http://www.sciencedirect.com/science/>

- article/pii/S0952197613001693.
- Chen, S. and Weiss, G. (2014). An intelligent agent for bilateral negotiation with unknown opponents in continuous-time domains (In press), *ACM Transactions on Autonomous and Adaptive Systems* .
- Coehoorn, R. M. and Jennings, N. R. (2004). Learning on opponent's preferences to make effective multi-issue negotiation trade-offs, in *Proceedings of the 6th Int. conf. on Electronic commerce* (ACM, New York, NY, USA), ISBN 1-58113-930-6, pp. 59–68.
- Dang, J. and Huhns, M. N. (2006). Concurrent multiple-issue negotiation for internet-based services, *IEEE Internet Computing* **10**, pp. 42–49.
- Daubechies, I. (2006). *Ten lectures on wavelets* (Society for Industrial and Applied Mathematics), ISBN 9780898712742.
- de Boor (1978). *A Practical Guide to Splines* (Springer-Verlag).
- Duan, L., Dogru, M. K., Ozen, U., and Beck, J. (2012). A negotiation framework for linked combinatorial optimization problems, *Autonomous Agents and Multi-Agent Systems* **25**, 1, pp. 158–182.
- Faratin, P., Sierra, C., and Jennings, N. R. (1998). Negotiation decision functions for autonomous agents. *Rob. Autom. Syst.* **24**, 4, pp. 159–182.
- Faratin, P., Sierra, C., and Jennings, N. R. (2002). Using similarity criteria to make issue trade-offs in automated negotiations, *Artificial Intelligence* **142**, 2, pp. 205–237.
- Hao, J. and Leung, H. (2012). ABiNeS: An adaptive bilateral negotiating strategy over multiple items, in *Proceedings of WI/IAT'2012* (IEEE Computer Society), pp. 95–102.
- Hendrikx, M. (2011). A survey of opponent models in automated negotiation, Tech. rep., Delft University of Technology, The Netherlands.
- Hindriks, K. and Tykhonov, D. (2008). Opponent modelling in automated multi-issue negotiation using bayesian learning, in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems* (ACM, Estoril, Portugal), ISBN 978-0-9817381-0-9, pp. 331–338.
- Hou, C. (2004). Predicting agents tactics in automated negotiation, in *Intelligent Agent Technology, IEEE / WIC / ACM International Conference on*, Vol. 0 (IEEE Computer Society, Los Alamitos, CA, USA), ISBN 0-7695-2101-0, pp. 127–133.
- Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C., and Wooldridge, M. (2001). Automated negotiation: prospects, methods and challenges, *International Journal of Group Decision and Negotiation* **10**, 2, pp. 199–215.
- Kalai, E. and Smorodinsky, M. (1975). Other solutions to nash's bargaining problem, *Econometrica: Journal of the Econometric Society* , pp. 513–518.
- Lai, G., Li, C., Sycara, K., and Giampapa, J. (2004). Literature review on multi-attribute negotiations, Tech. rep., Robotics Institute, Carnegie Mellon University.
- Lau, R. Y., Li, Y., Song, D., and Kwok, R. C. W. (2008). Knowledge discovery for adaptive negotiation agents in e-marketplaces, *Decision Support Systems* **45**, 2, pp. 310 – 323.

Bibliography

29

- Lin, R. and Kraus, S. (2010). Can automated agents proficiently negotiate with humans? *Communications of the ACM* **53**, 1, pp. 78–88.
- Lin, R., Kraus, S., Wilkenfeld, J., and Barry, J. (2008). Negotiating with bounded rational agents in environments with incomplete information using an automated agent, *Artificial Intelligence* **172**, pp. 823–851.
- Lomuscio, A. R., Wooldridge, M., and Jennings, N. R. (2003). A classification scheme for negotiation in electronic commerce, *Group Decision and Negotiation* **12**, 1, pp. 31–56.
- Lopes, F., Wooldridge, M., and Novais, A. (2008). Negotiation among autonomous computational agents: principles, analysis and challenges, *Artificial Intelligence Review* **29**, pp. 1–44.
- Mnih, V., Larochelle, H., and Hinton, G. E. (2012). Conditional restricted boltzmann machines for structured output prediction, *CoRR* **abs/1202.3748**.
- Mor, Y., Goldman, C. V., and Rosenschein, J. S. (1996). Learn your opponent's strategy (in polynomial time)! in *In Proceedings of IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems* (Springer-Verlag), pp. 164–176.
- Muthoo, A. (1999). *Bargaining theory with applications* (Cambridge University Press).
- Myerson, R. B. (1981). Utilitarianism, egalitarianism, and the timing effect in social choice problems, *Econometrica: Journal of the Econometric Society*, pp. 883–897.
- Nash, J. (1953). Two-person cooperative games, *Econometrica* **21**, 1, pp. pp. 128–140, <http://www.jstor.org/stable/1906951>.
- Nash, J. F. et al. (1950). Equilibrium points in n-person games, *Proceedings of the national academy of sciences* **36**, 1, pp. 48–49.
- Osborne, M. and Rubinstein, A. (1994). *A Course in Game Theory* (MIT Press).
- Oshrat, Y., Lin, R., and Kraus, S. (2009). Facing the challenge of human-agent negotiations via effective general opponent modeling, in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems- Volume 1* (International Foundation for Autonomous Agents and Multiagent Systems), pp. 377–384.
- Ponka, I. (2009). *Commitment models and concurrent bilateral negotiation strategies in dynamic service markets*, Ph.D. thesis, University of Southampton, School of Electronics and Computer Science.
- Ragone, A., Noia, T., Sciascio, E., and Donini, F. (2008). Logic-based automated multi-issue bilateral negotiation in peer-to-peer e-marketplaces, *Autonomous Agents and Multi-Agent Systems* **16**, 3, pp. 249–270.
- Raiffa, H. (1982). *The art and science of negotiation* (Harvard University Press Cambridge, Mass), ISBN 9780674048133.
- Saha, S., Biswas, A., and Sen, S. (2005). Modeling opponent decision in repeated one-shot negotiations, in *Proceedings of the Fourth international joint conference on Autonomous agents and multiagent systems* (ACM, New York, NY, USA), ISBN 1-59593-093-0, pp. 397–403.
- Shoham, Y. and Leyton-Brown, K. (2009). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations* (Cambridge University Press).

- Taylor, G. W. and Hinton, G. E. (2009). Factored conditional restricted Boltzmann Machines for modeling motion style, in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09 (ACM, New York, NY, USA), ISBN 978-1-60558-516-1, pp. 1025–1032, doi:10.1145/1553374.15533505, <http://dx.doi.org/10.1145/1553374.15533505>.
- Wang, M., Wang, H., Vogel, D., Kumar, K., and Chiu, D. K. (2009). Agent-based negotiation and decision making for dynamic supply chain formation, *Eng. Appl. Artif. Intell.* **22**, 7, pp. 1046 – 1055.
- Weiss, G. (ed.) (2013). *Multiagent Systems*, 2nd edition (MIT Press).
- Williams, C., Robu, V., Gerdin, E., and Jennings, N. (2011). Using gaussian processes to optimise concession in complex negotiations against unknown opponents, in *Proceedings of IJCAI'2011* (AAAI Press), pp. 432–438.
- Williams, C. R. (2012). *Practical Strategies for Agent-Based Negotiation in Complex Environments*, Ph.D. thesis, University of Southampton.
- Wooldridge, M. (2009). *An introduction to multiagent systems* (John Wiley & Sons).
- Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice, *The knowledge engineering review* **10**, 02, pp. 115–152.
- Yousefi, S., Weinreich, I., and Reinarz, D. (2005). Wavelet-based prediction of oil prices, *Chaos Solitons Fractals* **25**, 2, pp. 265–275.
- Zeng, D. and Sycara, K. (1997). How can an agent learn to negotiate? in *Intelligent Agents III Agent Theories, Architectures, and Languages* (Springer), pp. 233–244.
- Zeng, D. and Sycara, K. (1998). Bayesian learning in negotiation, *International Journal of Human-Computer Studies* **48**, 1, pp. 125–141.