An Adaptive Agent Model Estimating Human Trust in Information Sources

Mark Hoogendoorn, S. Waqar Jaffry, Jan Treur
Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{mhoogen, swjaffry, treur}@few.vu.nl

Abstract

For an information agent to support a human in a personalized way, having a model of the trust the human has in information sources may be essential. As humans differ a lot in their characteristics with respect to trust, a trust model crucially depends on specific personalized values for a number of parameters. This paper contributes an adaptive agent model for trust with parameters that are automatically tuned over time to a specific individual. To obtain the adaptation, four different techniques have been developed. In order to evaluate these techniques, simulations have been performed. The results of these were formally verified.

1. Introduction

In order for humans to be more effective in performing certain tasks, Personal Assistant agents [1] have been proposed that support humans in these task operations. Such support can for instance take the form of the agent giving particular advice to the human (e.g. showing a manual for performing a particular operation in case the human is making insufficient progress) or could go further, for example by the agent assigning (part of) the task to a computer or another human. To be most effective, and in order for the humans to accept the support of the agent, the human should be assisted in a personalized way.

One aspect of importance for supporting the human in a personalized way is the trust the human has in various information sources that can potentially aid in performing the task. For instance, the manual for a task might contain a lot of flaws, whereas another human that has experience with the task can immediately aid in a correct fashion. Hence, the trust value for the manual will be much lower, and a personal assistant showing this manual will most likely be ignored. In demanding circumstances, for example on board of a Naval vessel, where critical tasks are performed, such information is crucial for personal assistant agents to be useful and help improve the overall effectiveness of a mission.

In trust research, a variety of computational models have been proposed which represent human trust [2][3][4]. Such models often assume that trust values of a human with respect to another party over time are defined by a certain trust function, which determines the trust value at some time point using the experiences of the human with the specific party up to that point. Hereby, a number of parameters are still tunable towards specific characteristics of the human. For instance, how fast the trust decays after a period without experiences, the initial trust value, how much one positive or negative experience counts, et cetera. In [3] an additional factor is taken into account, namely how the party performs relative to its competitors.

As humans can differ in their characteristics, the behavior of a trust model crucially depends on the specific values for the different parameters. How these parameters can be tuned to a specific individual is the main challenge addressed in this paper. An approach is proposed that can observe the behavior of the human in consulting other parties (e.g., looking up the manual), and estimate appropriate parameter settings of a trust model such that it describes the human's trust in an accurate way. To this end, an existing trust model is taken as a basis (cf. [3]). This trust model has four parameters: the initial trust value, the decay factor of trust, the weight of positive and negative experiences (trust flexibility), and the weight of experiences with competitors upon the trust value (trust autonomy). The observable behavior assumed consists of the choices a human makes in consulting competitive parties (e.g., do I ask the human, or look in the manual) as well as the (positive or negative) outcomes of these consultations. Using this observation information, methods are applied to find appropriate parameter settings that describe this human behavior.

To tune the parameters to a person over time within the adaptive agent model, different methods have been

used and compared. The first method is based on exhaustive search through the space of parameter combinations. The second method is a bisection method, where every step the parameter value intervals are divided in two. As a third method the bisection method has been combined with the use of a library of known solutions. Finally, a Simulated Annealing approach [5] was investigated. These methods have been compared based upon computation time, as well as the accuracy of the solution found.

This paper is organized as follows. In Section 2 the basic trust model used is briefly explained. Section 3 discusses the adaptive agent model and the methods used for parameter tuning. In Section 4 simulation results are described, whereas Section 5 addresses automated verification of properties of the adaptation. Section 6 discusses related work, and Section 7 concludes the paper and presents future work.

2. Trust Model

This section describes the model of human trust on information sources used in subsequent sections for adaptive parameter estimation (cf. [3]). In this model information sources are considered competitors, and the human trust on an information source depends on the relative experiences with the information source to the experiences from the other information sources. The model defines the total trust of the human as the difference between positive trust and negative trust (distrust) on the information source. It includes personal human characteristics like trust decay, flexibility, and degree of autonomy (context-independence) of the trust. Figure 1 shows the dynamic relationships in the model used.

In this trust model it is assumed that the human is bound to request one of the available information sources $\{IS_1, IS_2, \ldots IS_n\}$ at each time step. The human requests the information source IS_i with highest trust value from the vector of trust values $\{T_1(t), T_2(t), \ldots T_n(t)\}$ available on the information sources $\{IS_1, IS_2, \ldots IS_n\}$ respectively at time t. In response of the human's request IS_i gives experience value ($E_i(t)$) from the set {-1, 1} indicating a negative resp. positive experience. This experience is used to update the human trust value for the next time point. Besides values from {-1, 1} the experience value can also be 0, indicating that IS_i gives no experience to the human at time point *t* (it was not requested).

This trust model can be tuned to several personal human characteristics described in [3] including trust flexibility β (measuring the change in trust on each new experience), decay γ (decay in trust when there is no experience), and autonomy η (dependence of the trust calculation considering other options). β, γ and η are model parameters which have values between [0, 1].

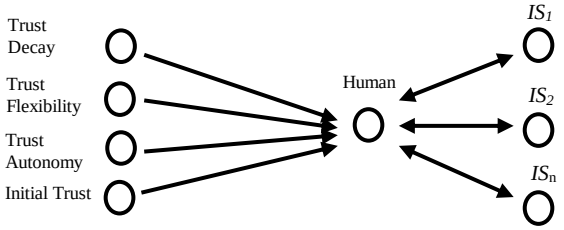


Figure 1: Dynamic relationships in the trust

As mentioned before, the model is composed from two models: one for the positive trust, accumulating positive experiences, and one for negative trust, accumulating negative experiences. Both negative and positive trust are a number between [0, 1]. The human's total trust $T_i(t)$ of IS_i is the difference of the human's positive and negative trust of IS_i at time point *t* that is a number between [-1, 1] where -1 and 1 represent minimum and maximum values of the trust respectively. In particular, also the human's initial total trust of IS_i at time point 0 is $T_i(0)$ which is the difference of human's initial trust $T_i^+(0)$ and distrust $T_i^-(0)$ in IS_i at time point 0.

In differential equation form the change in positive and negative trust over time is modeled by (cf. [3]):

$$\frac{dT_i^+(t)}{dt} = \beta * \begin{pmatrix} \eta * (1 - T_i^+(t)) + \\ (1-\eta) * (\tau_i^+(t) - 1) \\ * T_i^+(t) * (1 - T_i^+(t)) \end{pmatrix} * E_i(t) * (E_i(t) + 1)/2$$
$$- \gamma * T_i^+(t) * (1 + E_i(t)) * (1 - E_i(t))$$

$$\frac{dT_i^-(t)}{dt} = \beta * \begin{pmatrix} \eta * (1 - T_i^-(t)) + \\ (1-\eta) * (\tau_i^-(t) - 1) \\ * T_i^-(t) * (1 - T_i^-(t)) \end{pmatrix} * E_i(t) * (E_i(t) - 1)/2$$
$$- \gamma * T_i^-(t) * (1 + E_i(t)) * (1 - E_i(t))$$

In the above equations, $E_i(t)$ is the experience value given by the IS_i at time point *t*. Also $\tau_i^+(t)$ and $\tau_i^-(t)$ are the human's relative positive and negative trust on IS_i at time point *t* which is the ratio of the human's positive or negative trust of IS_i to the average human's positive or negative trust on all options at time point *t* defined as follows

$$\tau_i^+(t) = \frac{T_i^+(t)}{\sum_{j=1}^{n} T_j^+(t) / n} \quad \text{and} \quad \tau_i^-(t) = \frac{T_i^-(t)}{\sum_{j=1}^{n} T_j^-(t) / n}$$

3. The Adaptive Agent Model

This section describes the adaptive agent model for estimating human personality attributes to model trust in (competitive) information sources. The model is shown in Figure 2. In this model it is assumed that the

agent observes the behavior of the human (request to the information source) and the results of the information source (the positive or negative response to the request) over time. At each time step the agent adapts the model parameters using the available information. The agent starts with an initial parameter vector, calculates human trust on information sources and predicts the information source to be requested by the human, then observes the actual human request to the information source and information source response. If the human places request to the same information source as predicted by the agent then agent does not change parameter vector (considering this prediction being correct), otherwise parameter values are adopted accordingly.

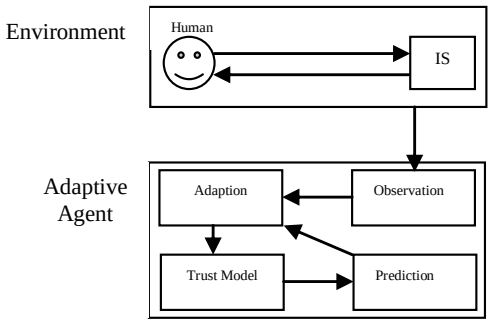


Figure 2: Adaptive Agent Model

To measure how accurately a parameter vector is representing human personality attributes, the accuracy of the parameter vector can be calculated as the ratio between the number of correct predictions to the number of total observations made by the agent by:

Accuracy=Correct Predictions / Observed Behaviors

3.1. Exhaustive Search Method

Using this method the entire attribute search space is explored to find the vector of parameter settings with maximum accuracy. This method guarantees the optimal solution, described as follows:

```
for each observed behavior B
    for each vector of parameter setting P
      calculate accuracy of P
    end for
output vector of parameter settings with max. accuracy
end for
```

In the above algorithm, calculation of the accuracy of a vector of parameter setting P entails that agent predicts the information source to be requested and observes the actual human request. It then uses the equation for calculating the accuracy described before.

Here if α parameters with precision τ are to be estimated, N is the number of information sources, and B the number of observed behaviors (i.e., time points), then the worst case complexity of the method can be expressed as $O((10)^{\alpha\tau} NB^2)$. In particular when α=3 (β, γ, and η) also τ=2 (or 0.01), N=3 and B=100 then complexity would result into 3 x 10^{10} steps which is exponential in number of parameters and precision.

3.2. Bisection Search Method

In the bisection method the attribute search space is reduced by halving the intervals for the parameter values at each step. The method is as follows:

```
for each observed behavior B
    P1 = vector of lowest possible values of parameters
    P2 = vector of highest possible values of parameters
        while P2 – P1 > required precision
            d1 = calculate accuracy of P1
            d2 = calculate accuracy of P2
            P3 = (P1 + P2)/2
                if d1 > d2
                then P2 = P3
                else P1 = P3
        end while
        if d1 > d2
        then output P1
        else output P2
end for
```

In the above algorithm P1, P2 and P3 are the vectors of parameters to be estimated. Operations should be considered as vector operations for example (P2 – P1 > required precision) in above algorithm means that it should hold for all components. The worst case complexity of this method is $O(\alpha\tau NB^2)$.

3.3. Extended Bisection Search Method

The extended bisection method is an extension to the bisection method. Here after finding a vector of parameter setting against a new observed human behavior, this vector is kept in a list for future use. On each next observed human behavior, bisection search finds a new vector of parameter settings which is then compared with the accuracy of all previously known vectors at the current time. The vector with maximum accuracy is outputted. The method is described as follows:

```
Solution-Parameter-List L is Empty
for each observed behavior B
    P1= vector of lowest possible values of parameters
    P2= vector of highest possible values of parameters
        while P2–P1 > required precision
            d1 = calculate accuracy of P1
            d2 = calculate accuracy of P2
            P3 = (P1 + P2)/2
```

```
            If d1 > d2
            then P2 = P3
            else P1 = P3
        end while
        if d1 > d2
        then Add P1 in L
        else Add P2 in L
        for all parameter vectors P in L
          recalculate accuracy of P
        end for
    output P with maximum accuracy from list L
end for
```

The worst case complexity of this method is $O(\alpha\tau NB^2)$.

3.4. Simulated Annealing Method

Simulated Annealing [5] uses a probabilistic technique to find a vector of parameter settings that best corresponds to human personality characteristics. In this method a random vector of parameter settings is chosen as the best available parameter setting at the start. Then a displacement is introduced into this vector to generate a neighbor of the current parameter settings in the search space. If this neighboring vector is found a more appropriate representation of the observed human behavior then it is marked as the best known vector of parameter settings, otherwise a new neighbor is selected to evaluate its appropriateness. The number of neighbors that could be trialed is limited by the computational budget available to the algorithm. The displacement in the vector of parameter settings to find a new neighbor depends on the temperature of the algorithm, in case the temperature is higher, the steps will become larger. The temperature of the algorithm at a certain time point is defined as follows:

Temperature = computational_budget_left * (1 – accuracy)

In the above expression accuracy is the accuracy of currently known best vector of parameter settings. The displacement in the vector of parameter (say ω) can be derived from the following two equations selecting one at random,

ω = ω + Temperature * (1–ω) * random_between[0,1]
ω = ω – Temperature * ω* random_between[0,1]

The algorithm is described as follows:

```
for each observed behavior B
    chose a random parameter vector R
    while computational-budget-remains
        find neighbor R1 of parameter vector R
          if accuracy of R1 > R
          then R=R1
        decrease computational-budget
    end while
    output R
end for
```

If C is the computational budget, then the worst case complexity of the method can be expressed as $O(CNB^2)$. Here it could be noted that the computational complexity of this method is independent of the number of parameters and desired precision.

Considering a case where an adaptive agent is to be designed for the trust model that has α number of parameters to be estimated with τ digits of precision in the estimated values then the worst case complexities of the methods described above are shown in Table 1.

Methods	**Complexity**
Exhaustive	$O((10)^{\alpha\tau} NB^2)$
Bisection	$O(\alpha\tau NB^2)$
Extended Bisection	$O(\alpha\tau NB^2)$
Simulated Annealing	$O(CNB^2)$

Table 1: The rate of the growth of time complexity

From Table 1 it is obvious that the exhaustive search method, being exponential in the number of parameters to be estimated and the precision required in the values of parameters, is impractical for use in an adaptive agent when the number of parameters of the trust model under consideration increases or a high precision in the parameter values is required. For instance, if the initial trust value of the information sources is taken as a parameter, computation time would severely increase for exhaustive search.

4. Simulation Results

This section describes the experimental configurations and their results for various settings. In order to use the above methods for personalization of the adaptive agent according to human personality attributes, it is necessary to have actual human behavior observed by the agent. For experimental purposes, first the human behavior is generated by the same trust model [3] for specific values of human personality attributes (namely β, γ and η) and then the methods described in Section 3 are used by the adaptive agent against these behaviors to predict human attributes. Configurations for generating human behavior are described in Table 2.

Within the experiments, the parameters β, γ, and η are estimated. The desired precision of the estimated parameters and the number of information sources were kept constant. Furthermore, it is assumed that one of the three information sources (IS_1) gives positive while the other two (IS_2 , IS_3) give negative responses over each human request. Five cases have been presented, each representing different human personalities (i.e.

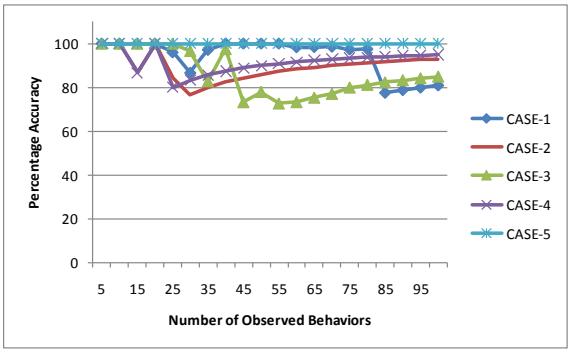
different values of the parameters). For the sake of simplicity, personality attributes (β, γ, η) are assumed having low (0.25, 0.01, 0.25) and high (0.75, 0.25, 0.75) values resp. Due to presentation limitations, only few of the possible combination of these attributes have been shown here. The experimental configurations used for the adaptive agent are the same as those shown in Table 2, except that the agent does not know the values of β, γ, and η in advance (as these are to be estimated).

Case	**1**	**2**	**3**	**4**	**5**
No of Parameters	*3*	*3*	*3*	*3*	*3*
Precision (digits)	*2*	*2*	*2*	*2*	*2*
Information Sources	*3*	*3*	*3*	*3*	*3*
Response of IS_1, IS_2, IS_3	*1,-1,-1*	*1,-1,-1*	*1,-1,-1*	*1,-1,-1*	*1,-1,-1*
Observed Behaviors	*100*	*100*	*100*	*100*	*100*
Trust Decay γ	*0.01*	*0.01*	*0.01*	*0.25*	*0.01*
Trust Flexibility β	*0.75*	*0.75*	*0.25*	*0.75*	*0.75*
Trust Autonomy η	*0.25*	*0.25*	*0.25*	*0.25*	*0.75*
Human initial Trust on IS_1, IS_2, IS_3	*0.00, 0.15, 0.30*	*0.00, 0.05, 0.10*	*0.00, 0.05, 0.10*	*0.00, 0.05, 0.10*	*0.00, 0.05, 0.10*
Computational Budget (for S.A.)	*1000*	*1000*	*1000*	*1000*	*1000*

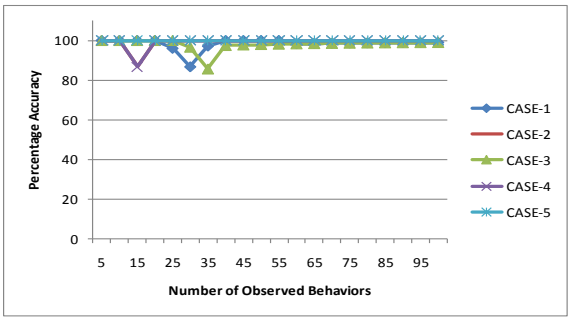
Table 2: Model Configurations used for Experiments

4.1. Accuracy of Estimated Parameter Settings

This section describes the accuracy of the adaptive agent's parameter estimation using the methods discussed in Section 3. Hereby, the proposed adaptive agent has been implemented in the C++ programming language. The graphs depicted in Figure 3 show the percentage accuracy of the parameter estimation for the bisection and extended bisection methods against the number of human behaviors observed. In Figure 3a it can be noted that initially for a smaller number of observed behaviors the accuracy of the estimated parameters is much higher. This is due to the fact that initially, the human behavior is slightly disclosed so there are many possible parameter settings that correspond to the observed human behavior hence the bisection method can find that with good accuracy. As the human behavior reveals itself more extensively over time, the set of the possible parameter settings that correspond to this behavior becomes smaller that makes good accuracy harder to achieve. In Figure 3b it can be noted that the extended bisection method gives much better accuracy then the original bisection method as it keeps all previously good known solutions in memory for future use.

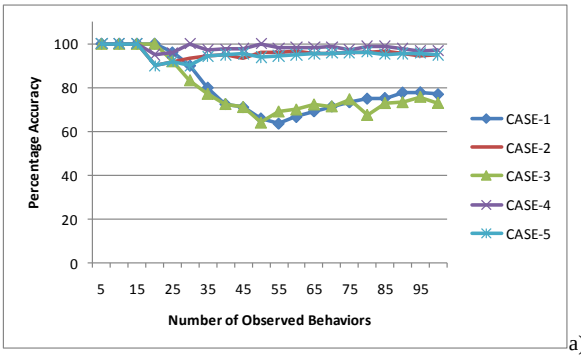


a)

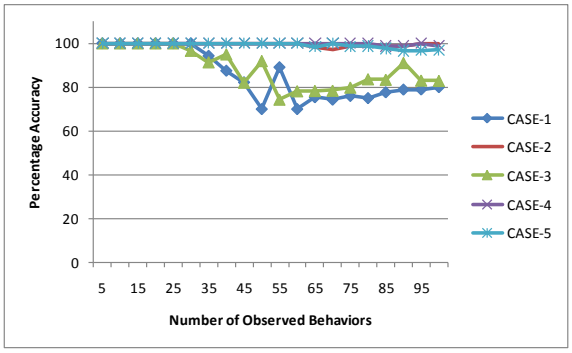


b)

Figure 3: Percentage accuracy of the agent using the a) Bisection and b) Extended Bisection Method



a)



b)

Figure 4: a) Maximum and b) Minimum percentage accuracy using Simulated Annealing.

As Simulated Annealing is a probabilistic method several simulations were conducted to find the behavior of this method. Figures 4a and 4b show the maximum and minimum percentage accuracy of the estimated

parameter settings for the simulated annealing method in ten simulation runs.

Figure 5 shows the percentage accuracy of the estimated parameter settings for different methods after observing 100 human behaviors.

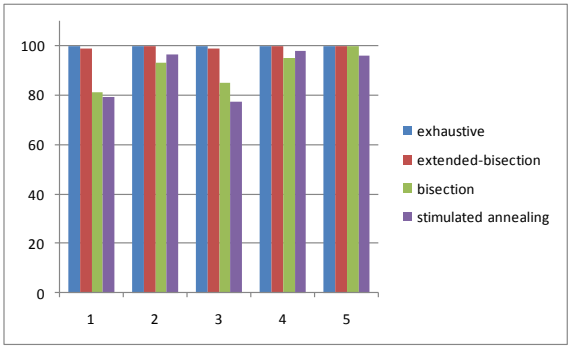


Figure 5: Accuracy of methods in different cases

Here it can be observed that the exhaustive search method gives 100 percent accuracy while the extended bisection outperforms the bisection method in all cases. Note that the accuracy mentioned here for the simulated annealing is the average accuracy for ten sample runs. It can be seen that bisection and simulated annealing are competitive in different cases.

It should be noted that three human personality attributes (β, γ, η) with two digit precision generates one million human personalities, hence different values of human personality attributes may generate the same behavior trace particularly when only few human behaviors are observed. The number of human personality attributes vectors exactly generating the behaviors of cases 1, 2, 3, 4 and 5 of Table 2 are found to be 2, 1343, 3, 2387 and 46742 respectively by the exhaustive search method. It could be noted that performance of all methods (except exhaustive search) in case one and three is below average than the others. This happened because behavior generated by human under the configurations of these two cases matches to a small number of human personality attributes vectors in the entire human personality attributes vectors set. This makes it harder for search methods to locate a vector corresponding to the human behavior.

Finally, during these experiments the execution time has been measured. Bisection method was found most efficient taking 0.013375 seconds for parameter estimation against 100 human behaviors while the extended bisection took 0.014070. Simulated Annealing for a computational budget 1000 took 0.022190 seconds that is approximately twice the time of the bisection method while the exhaustive search completed the task in 13.375 seconds. It can be observed that exhaustive search is much more expensive than the other approaches, while the extended bisection consumes almost the same computation time as the bisection method. The computational time of the simulated annealing depends on the computational budget assigned.

5. Verification of Properties

In this section, a number of relevant properties are presented that have been specified and used to evaluate the approach.

5.1. Accuracy of Estimated Parameter Settings

In this section, a number of identified properties are discussed. In order to conduct such an automated verification, the properties have been specified in a language called TTL (for Temporal Trace Language), [6] that features a dedicated editor and an automated checker. The language is a predicate logical temporal logic which allows for both qualitative as well as quantitative aspects to be expressed. TTL is built on atoms referring to *states* of the world, *time points* and *traces*, i.e. trajectories of states over time. In addition, *dynamic properties* are temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γ over state ontology Ont, the state in γ at time point t is denoted by state(γ, t). These states can be related to state properties via the infix predicate |=, where state(γ, t) |= p denotes that state property p holds in trace γ at time t. Based on these statements, dynamic properties can be formulated in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\forall$, $\exists$. For more details, see [6].

The ontology used to specify these properties is shown in Table 3. First of all, a property was specified which states that the number of possible solutions for the parameter setting of the trust model decreases as the number of observed behaviors of the human (i.e. choices made and experiences of the human) increases. Note that this property is merely useful for the exhaustive search method.

P1(trace, λ): Information reduces possible solutions
If at time point t the set of observed behaviors is of size *n* and the number of solutions found is *p1*, then if at a later point in time t2 the set of observed behaviors is larger (i.e. *m* > *n*), then the number of solutions found *p2* is less than or equal to λ * *p1*. Formally:

```
∀t:time, n:integer, p1:integer
[ state(γ, t) |= number_observed_behaviors(n) &
  state(γ, t) |= number_solutions(p1) ] ⇒
  [ ∀t2:time, m:integer, p2:integer
    [ [ t2 > t & m > n &
```

state(γ, t2) |= number_observed_behaviors(m) &
state(γ, t2) |= number_solutions(p2)] ⇒ p2 ≤ p1 * λ]]]

The properties can be made more specific, for example by selecting a value of λ = 1 which specifies that more observed behaviors of the humans will never increase the number of possible parameter settings.

Predicate	Explanation
number_observed_ behaviors: integer	The size of the history available to the agent (i.e. human behavior and experiences).
number_solutions: integer	The size of the set of possible solutions that describe the human behavior.
solution: real x real x real	The trust parameters specified (β, γ, η respectively) are a solution for the current set of human behaviors.
accuracy: real x real x real x real	The trust parameter values specified (β, γ, η respectively) describe the human behavior with an accuracy specified in the last real value.

Table 3: Ontology for properties

Besides this property, the accuracy of the solution can be investigated as well. Of course, in the case of exhaustive search this property is trivial as it will always be completely accurate. However for the case of the bisection approach the property is very relevant. Property P2 specifies a requirement upon a certain minimum accuracy (e.g. to make a personal assistant believable). Note that the informal form has been omitted for the sake of brevity.

P2(trace, x, t): Minimum accuracy at least x at time t
∀x2:integer, β, γ, η:real
[state(γ, t) |= solution(β, γ, η) &
state(γ, t) |= accuracy(β, γ, η, x2)]
⇒ x2 ≥ x

Besides the fact that if a solution is present, this solution is indeed accurate, of course an additional concern is that there is at least one solution available. This is specified in property P3.

P3(trace, t): At least one solution
∃β, γ, η:real,
[state(γ, t) |= solution(β, γ, η)]

The combination of property P2 and P3 guarantees that there is always a solution, which is also of a certain quality. This is expressed in property P4.

P4(trace, x): At least one solution of good quality
∀t:time [P2(γ, x, t) & P3(γ, t)]

Finally, a property was specified which expresses that the accuracy of the solutions found is at least as good in case the set of observed behaviors is larger.

P5(trace): Accuracy does not decrease over time
∀t:time, x1, β1, γ1, η1:real, n:integers
[[state(γ, t) |= number_observed_behaviors(n) &
state(γ, t) |= accuracy(β1, γ1, η1, x1) &
¬∃x':real < x1 [state(γ, t) |= accuracy(β1, γ1, η1, x')]]
⇒ [∀t2:time > t, m:integer > n, x2, β2, γ2, η2:real
[state(γ, t2) |= number_observed_behaviors(m) &
state(γ, t2) |= accuracy(β2, γ2, η2, x2) &
¬∃x'':real < x2 [
state(γ, t2) |= accuracy(β2, γ2, η2, x'')]]
⇒ x2 ≥ x1]]

5.2. Verification Results

This Section presents the results for the verification of the properties upon traces for each of the methods. The properties P1, P2, and P4 have been verified using the following parameters: P1 with a value of λ = 1; P2 with a value of x=1 and x=0.8, and P4 with a value x of 0.8.

For the case of **exhaustive search** all properties always hold, except the reduction of the set of possible solutions (P1). This property holds up till a certain time point only, since the set is eventually reduced to a set of one or two parameters that no longer reduces as new experiences come in. For case 1 and 3 the solution set reduces until time point 9/10, whereas for the other cases the set only reduces in the very beginning. Hence, those cases contain information that reduce the search space more rapidly compared to case 1 and 3.

In comparison, the results for the **bisection** show that property P1 also always holds (which is obvious, since only one solution is maintained per time point). Furthermore, the bisection does not always find the optimal solution (P2(1)), except for case 5. A minimum accuracy of 0.8 is only reached for case 4 and 5. The method does always find a solution (property P3), which (as already shown in P2) is not always of a quality higher than 0.8 for cases 1, 2, and 3. Finally, the quality of the solution fluctuates (i.e. does not always increase or at least stay the same as indicated in P5) for all cases, except for case 5.

The **extended bisection** method shows a higher accuracy than the standard bisection method. It always finds a solution which surpasses an accuracy of 0.8, whereas this is only seen for case 4 and 5 in the standard bisection case. The other properties show identical behavior.

For **Simulated Annealing** both the worst and the best case have been tested. The results are identical for the worst and best case: P1(1) is always satisfied, P2(1) never holds, whereas P2(0.8) holds for cases 2, 4, and 5 respectively. P3 always holds, P4(0.8) holds for cases 2, 4, and 5 again, and P5 always fails. Hence, the Simulated Annealing does perform worse (with respect to solution quality) compared to the extended bisection approach, as it never finds the optimal solution for all time points, and does not always find a reasonable solution (in this case indicated by at least 80% accuracy), whereas the extended bisection does.

6. Related Work

In this paper, the model for representing relative human trust as presented in [3] has been adopted. There also exist other approaches which deal with relative trust, such as [7]. Furthermore, trust models that do not explicitly represent relativeness are e.g. [2][4]. The approach presented in this paper for learning parameters is however generic, it is based on the possibility to observe behavior of a human, and tailor the parameters of an arbitrary trust model based upon this observed behavior. In [8] the necessity of good trust estimation is also acknowledged, it is stated that for the auctioning mechanisms inaccurate trust measures reduce the amount of trade, individual profits and social welfare.

In the domain of parameter learning, several approaches have been proposed, whereby approaches can either come to an exact solution or can make an approximation thereof. The exact solutions are typically computationally expensive and therefore less attractive (see e.g. [5]). Heuristics can be introduced, but are usually domain specific making the solutions nongeneric. As a result, a variety of approximate methods have been introduced, including Simulated Annealing [5], and Genetic Algorithms (see e.g. [9]). The methods used in this paper represent examples of both exact solution methods, as well as approximations. Both have been compared for the case of the trust model to show the tradeoff between computation time and the quality of the solution.

7. Conclusions and Future Work

In this paper, an approach has been presented to learn parameters of a given trust model based upon observed experiences of a human. This approach has been introduced to enable a personal assistant agent to take such human trust into account when giving advice. Hereby, an existing trust model (cf. [3]) has been taken as a basis. Several methods have been used to enable learning of these parameters, including exhaustive search, Simulated Annealing, bisection, and an extended form of bisection. The process is adaptive in the sense that new experiences can come in, and are taking into consideration by finding the most appropriate parameter setting. The algorithms have been tested for various cases, and the results thereof have been analyzed using formal verification techniques (cf. [6]). The results show that the computation time of the exhaustive search scales up worst, whereas Simulated Annealing scales up best. When looking at the accuracy however, the inverse is true: exhaustive search finds the most accurate point, whereas Simulated Annealing sometimes only comes up with poor solutions. The bisection, and the more advanced extended bisection approach are right in the middle: They do have a higher accuracy and are computationally less expensive. The choice of which method to choose ultimately depends on the domain. For particular domains a higher computation time might be acceptable as long as the results are good, whereas in other more time critical domain speed could be a necessity. In this respect, the bisection approaches are a good combination of both worlds.

For future work, it would be interesting to investigate what happens in case the human does not select an information source solely on the basis of the highest trust level. It could be imagined that other options are also selected to get a better insight for their trustworthyness. The learning will become more difficult as the choices and experiences no longer necessarily give information about the trust level of the human.

8. References

[1] Kozierok, R., and Maes, P., "A Learning Interface Agent for Scheduling Meetings", In: Proceedings of the 1st Int.Conf. on Intelligent User Interfaces, 1993, pp. 81-88.

[2] Falcone, R., and Castelfranchi, C., "Trust dynamics: How Trust is Influenced by Direct Experiences and by Trust Itself", In: Proc. of AAMAS 2004, 2004, pp. 740–747.

[3] Hoogendoorn, M., Jaffry, S.W., and Treur, J., "Modeling Dynamics of Relative Trust of Competitive Information Agents", In: Klusch, M., Pechoucek, M., Polleres, A. (eds.), Proc. of the 12th Int. Workshop on Cooperative Information Agents, CIA'08. LNAI, vol. 5180. Springer, 2008, pp. 55-70.

[4] Jonker, C.M., and Treur, J., "Formal Analysis of Models for the Dynamics of Trust based on Experiences", In: F.J. Garijo, M. Boman (eds.), Proc. of MAAMAW'99, LNAI, vol. 1647, Springer Verlag, Berlin, 1999, pp. 221-232.

[5] Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., "Optimization by Simulated Annealing", Science, New Series, vol. 220, 1983, pp. 671-680.

[6] Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., and Treur, J., "Specification and Verification of Dynamics in Agent Models. Int. Journal of Cooperative Information Systems", vol. 18, 2009, pp. 167 - 193.

[7] Beth, T., Borcherding, M., and Klein, B., "Valuation of Trust in Open Networks", Proc. of 3rd European Symp.on Research in Computer Security (ESORICS), 1994, pp. 3-18.

[8] Braynov, S. and Sandholm, T., "Incentive Compatible Mechanism for Trust Revelation", In: Proc. AAMAS 2002, pp. 310–311.

[9] Jong, K. de, "Learning with Genetic Algorithms: An Overview", In: Machine Learning, vol. 2, 1998, pp. 121-138.