Trustworthy Advice

Nardine Osman, Patricia Gutierrez, Carles Sierra

Artificial Intelligence Research Institute (IIIA-CSIC), Campus de la UAB, 08193 Bellaterra, Spain

Abstract

'If you go to Ferran Adria's restaurant you will have the time of your life!' 'If you study everyday for two hours you will get very good marks next semester.' These are examples of advice. We say an advice has two components: a plan to perform and a goal to achieve. In dynamic logic, an advice could be formalised as: $[P_\eta]G$. That is, if η performs plan P, then goal G will necessarily be achieved. An adviser is an entity which provides such advice. An adviser may be an agent, a planner, or a complex recommender system. This paper proposes a novel trust model for assessing the trustworthiness of advice and advisers. It calculates the expectation of an advice's outcome by assessing the probabilities of the advised plan being picked up and performed, and the goal being achieved. These probabilities are learned from an analysis of similar past experiences using tools such as semantic matching and action empowerment.

Keywords: trust, probability theory, information theory, semantic similarity

2010 MSC: 68T01, 68W01

1. Introduction

Advice is what one relies on when making decisions on future actions. As such, advice is crucial in directing actions and interactions. Advice may be provided by a physician to a patient on what they can do to lead a healthier life. It could be provided by a tutor, suggesting the best exercises to solve in order to pass an exam. It could be provided by a personal assistant agent, suggesting an itinerary for a fabulous vacation. It could be provided by recommender systems, suggesting what would be the best movie one can rent.

But how can one choose which advice to follow and which advice to discard? This paper proposes a computational trust model, CONSUASOR, that assesses the trustworthiness of advice and their advisers. We say an advice has two components: a plan to perform and the goal intended to be achieved. In dynamic logic, this may be formalised as $[P_\eta]G$, where P is the recommended plan for η in order to fulfil goal G. That is, if η performs plan P, then goal G will necessarily be achieved. We note that the adviser may be a human, an agent, or a recommender system. The proposed model is based on the concept that an adviser is a good adviser if it is knowledgable about three main issues: (1) *compliance*, which describes how much compliant is the person being advised with following recommendations; (2) *honour*, which describes how much honourable is the person being advised in performing a recommended plan that he has accepted, and (3) *goal realisation*, which describes whether the recommended plan actually causes the goal to be fulfilled. Compliance is important, because good advisers are those that are knowledgable about who is willing to accept what advice, and personalising their plans accordingly. Honour is also important, because knowledge about whether the one being advised actually performs the recommended plan is fundamental. Finally, goal realisation is crucial, since a good adviser should be an adviser whose recommended plans can actually fulfil the intended goals.

Email addresses: nardine@iiia.csic.es (Nardine Osman), patricia@iiia.csic.es (Patricia Gutierrez), sierra@iiia.csic.es (Carles Sierra)

The proposed model then computes the trustworthiness of advice and advisers based on predicting the outcome of advice. The model calculates the expectation of an advice's outcome by assessing the probabilities of the advised plan being picked up and performed, and the goal being realised. These probabilities are learned from an analysis of similar past experiences using tools such as semantic matching and action empowerment.

The remainder of this paper is divided as follows. Section 2 presents our proposed trust model, CONSUASOR. Section 3 presents our experimental platform, benchmarks and evaluation. Finally, Section 4 provides a brief comparison to related work, before concluding with Section 5.

2. The CONSUASOR Model

The question that this section addresses is: *How much should one trust a recommender's advice?* In other words, how much should Γ trust ρ when ρ recommends a plan $[P_\eta]G$? We get inspiration from previous work (Osman et al., 2014), where trust was based on the expectation of a particular observation given a commitment, which was specified as a conditional probability:

$$p(Observe(\Gamma, \phi') \mid Commit(\rho, \phi))$$

where the term *Commit* had two arguments — the one making the commitment (ρ) and the action he was committing to (ϕ) — and the term *Observe* had two arguments — the one observing the outcome of the commitment (Γ) and the outcome of the commitment describing what ρ actually performed (ϕ'). The idea was that past commitments helped in assessing the expected outcome of similar current commitments. For example, if a seller has always delivered good quality goods, then one may expect the seller's next delivered goods to be of good quality as well.

In this section, we adopt the basic idea that a trust measure is based on the expectation of observing the possible outcomes of a commitment. When assessing the trustworthiness of advice, this expectation is specified as a conditional probability of observing an advice $[P_\eta]G$ realising its goal:

$$p(Observe(\Gamma, [P_\eta]G) \mid Commit(\rho, [P_\eta]G)) \tag{1}$$

where P is the plan recommended by ρ for η in order to fulfil goal G, and Γ represents the party that observes the realisation of the advice $[P_\eta]G$'s goal.

The remainder of this section is divided as follows. Section 2.1 presents the preliminaries needed for understanding the proposed model, Section 2.2 presents how the probability $p(Observe(\Gamma, [P_\eta]G) \mid Commit(\rho, [P_\eta]G))$ may be computed by relying on similar past experiences, Section 2.3 illustrates how the probability distribution is used to compute a final trust measure, and Section 2.4 closes with a trust algorithm that provides one example of a concrete implementation of the model.

2.1. Preliminaries

CONSUASOR is an experience-based trust model that relies on past experiences to predict future outcomes. As such, calculating the similarity between experiences is crucial. In this section, we present the preliminaries of our proposed model by defining experiences (Section 2.1.1), similarity measures (Sections 2.1.2), and the update of probabilities and probability distributions (Section 2.1.3) in the light of new experiences. Additionally, we also presents the general concept of information decay (Section 2.1.4), which is a basic notion that underlies our work, as it describes how information loses its value over time.

2.1.1. Experiences

A Single Experience. The advice that we are interested in assessing are conditional statements of the form: 'if the recommended plan is performed, then the intended goal will be realised'. As such, past experiences should not only keep track of advice and their realised goals, but of the fulfilment of the conditional part of the advice as well. This is because the adviser might give good advise, but the one being advised might not fulfil its duties in carrying out the recommended plan. As such, an experience should keep note of several issues:

- **The advice.** We interpret an advice as a commitment made by the adviser ρ that the goal G will be realised if η performs plan P. An advice is specified as $Commit(\rho, [P_\eta]G)_t$, where t specifies the time at which the advice $[P_\eta]G$ was recommended by ρ.

- **The accepted plan.** We interpret η accepting an advice as a commitment made by η to actually perform P'. This is specified as $Commit(\eta, P')_{t'}$, where t' describes the time at which η accepts plan P'.
- **The performed plan.** When η performs a plan P'', some entity β needs to observe (or verify) this performance, which is specified as $Observe(\beta, P''_{\eta})_{t''}$, where t'' describes the time at which β observed P''_{η}, or η's performed plan.
- **The realised goal.** The realised goal G' needs to be observed by some entity α, and this observation is specified as $Observe(\alpha, G')_{t'''}$, where t''' describes the time at which α observed the goal G' being realised.

A single experience μ is then recorded as follows:

$$\mu = \langle Commit(\rho, [P_{\eta}]G)_t, Commit(\eta, P')_{t'}, \\ Observe(\beta, P''_{\eta})_{t''}, Observe(\alpha, G')_{t'''} \rangle_{t<t'<t''<t'''}$$

Note that η may commit to a variation of the plan: $P' \neq P$. For example, assume an advice stating that one should "practice his piano twice a day", the one being advised may decide to commit to a variation of this plan, say "practicing his piano once a day". We also say that what may be observed may also be a variation of what has been committed to: $P'' \neq P'$ and $G' \neq G$.

In the general case, observers will be different from each other, and different from the adviser and the one being advised. Although it is possible to have particular cases where $\beta = \alpha$, or $\rho = \beta$, or $\eta = \alpha$, and so on.

Each element of the experience should have a different timestamp. An integrity constraint is then needed to check that a plan is accepted (by committing to it) after it has been recommended by an adviser, and that the plan has been performed (and observed) after it has been accepted, and that the goal has been realised (and observed) after the plan has been performed. This integrity constraint is specified by the condition $t < t' < t'' < t'''$.

History of Experiences. Numerous and different histories of experiences may exist, and we use the notation $H_{\alpha} = \{\mu, \mu', \ldots\}$ to describe α's history of experiences.

Populating the history of experiences needs to address numerous issues. For instance, how is information collected? In other words, when an adviser makes an advice, or when one accepts an advice, how is this information recorded? Also, who is trusted to observe a plan being performed or a goal being realised, and how are such observations carried out? How is the relation between elements recognised? For example, recognising that observing goal G' being realised is the result of plan P'' being performed, or that observing plan P'' being performed is the result of ρ honouring its commitment to P', or that committing to P' is the result of ρ's compliance with the advice $[P_{\rho}]G$.

In this paper, we do not dwell much on how a history of experiences is populated, as this could be context dependent. For instance, the entity maintaining a given history of experiences, whether this history is centralised or not, will need to specify who is trusted to record elements of an experience, and how are these elements recorded. The entity maintaining the history of experiences may also specify how experiences may be shared, and how reliable are shared experiences. As an example, consider an online classroom where the history is maintained by the online system. The tutor may advise its student to "focus on hand posture when playing the piano to improve performance". It is then the student's duty to confirm its willingness to follow the advice. An "Okay" from the student could signal its commitment to the advice. The student should also confirm whether they performed the recommended plan or not. For example, the student may say "I had troubles focusing on hand posture". The teacher may then assess the student's performance by listening and marking their uploaded performance. The marks may provide an indication on goal realisation. As such, different signals may be considered for recording experiences in different scenarios, and each system will need to define its own.

2.1.2. Similarity Measures

When assessing the level of similarity between a past experience and a current one, we need to take into consideration a number of similarity measures, such as the similarity of plans, or the similarity of goals, which we define next.

Plan and Goal Similarity. We assume there is a set of actions $\mathcal{A}$ that form the taxonomy of actions $T_{\mathcal{A}}$. Plans are sets of actions and the set of all possible plans is $\mathcal{P} = 2^{\mathcal{A}}$.[1] We assume that there is a semantic similarity relationship between actions $S : \mathcal{A} \times \mathcal{A} \rightarrow [0, 1]$ that shows the degree of relationship between actions. We also assume there is a set of propositional terms $\mathcal{T}$ that form the taxonomy of propositional terms $T_{\mathcal{T}}$. Goals are sets of propositional terms and the set of all possible goals is $\mathcal{G} = 2^{\mathcal{T}}$. We assume that there is a semantic similarity relationship between propositional terms describing goals (overloading symbol S) $S : \mathcal{T} \times \mathcal{T} \rightarrow [0, 1]$ that shows the degree of relationship between goals.

Plan similarity and goal similarity are then computed in the same manner accordingly:

$$Sim(Q,Q') = \frac{1}{2} \cdot \left(\min_{\phi \in Q}\{\max_{\phi' \in Q'}\{S(\phi,\phi')\}\} + \min_{\phi \in Q'}\{\max_{\phi' \in Q}\{S(\phi,\phi')\}\} \right) \quad (2)$$

where $\phi \in Q$ describes either an action of plan Q (if Q was a plan), or it describes a propositional term in goal Q (if Q was a goal), and S describes the *semantic similarity* between actions, or propositional terms.

In other words, Equation 2 states that to calculate the semantic similarity between two plans (goals), we first measure the semantic similarity between each action of the first plan (propositional terms of the first goal) with all the actions of the second plan (propositional terms of the second goal), and only the actions of the second plan (propositional terms of the second goal) that result with maximum similarity are then considered. This provides the maximum similarity measure that each action of the first plan (propositional term of the first goal) can have with the second plan (second goal). To aggregate those maximum similarity measures, we then take the minimum of those similarity measures. This describes the similarity between the first plan (goal) and the second plan. Then, to ensure that the function *Sim* is symmetric, we repeat the same process but in reverse order of plans (goals) — that is, we calculate the similarity between the second plan (goal) and the first — and we take the average of the two similarity measures between the two plans (goals). We note that the range of *Sim* is $[0, 1]$.

But what is the motivation behind choosing the min operator when calculating *Sim*? The basic idea behind this approach is that when considering the similarity of two entities, we need to consider how do the elements composing each entity relate to that entity. In our case, we say the entity (whether a plan or a goal) may be viewed as a set composed of a conjunction of elements. In mathematical terms (see Figure 1), there are a number of conjunctive operators that may be used, such as the product operator ($\prod$) and the minimum operator (min). We adopt the minimum operator, which describes an optimistic approach. For instance, if we are comparing $a \wedge b$ to c and $S(a, c) = 0.3$ and $S(b, c) = 0.2$, then we have $\min\{S(a, c), S(b, c)\} = 0.2$. For a more pessimistic approach, one can replace the minimum operator (min) with the product operator ($\prod$). In this case, $\prod\{S(a, c), S(b, c)\} = 0.06$, which is drastically smaller than considering the minimum. We note that the choice of operator will be domain dependent. Alternative methods for calculating *Sim* may be used as long as symmetry is maintained.

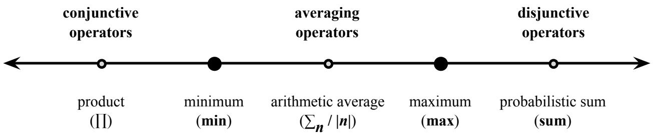


Figure 1: Classification of conjunctive, average, and disjunctive operators

Finally, we adopt the following definition of semantic similarity for S (Li et al., 2003):

$$S(\phi, \phi') = \mathrm{e}^{-\kappa_1 l} \cdot \frac{\mathrm{e}^{\kappa_2 h} - \mathrm{e}^{-\kappa_2 h}}{\mathrm{e}^{\kappa_2 h} + \mathrm{e}^{-\kappa_2 h}} \quad (3)$$

where e is Euler's number, l is the length (i.e. number of hops) of the shortest path between the terms ϕ and ϕ' in a taxonomy, h is the depth of the deepest concept subsuming both concepts, and κ_1 and κ_2 are parameters scaling the contribution of shortest

[1] A plan is usually understood as a temporal set of actions, describing the detailed steps (along with their conditions) needed for achieving a given goal. In this paper, we simplify the notion of a plan by reducing it to a *set of actions*. The only impact of this simplification is keeping the definition of plan similarity relatively straightforward. Adopting the definition of a plan as a *temporal set of actions* and redefining plan similarity accordingly is left for future work.

path length and depth, respectively. Essentially, κ_1 and κ_2 are parameters that α could use to customise the weight given to l and h, respectively. The function S is symmetric (i.e. $S(\phi, \phi') = S(\phi', \phi)$), and its range is $[0, 1]$.

The basic idea of semantic similarity is that the concepts within a taxonomy are closer, semantically speaking, depending on how far away are they in the taxonomy's graph. Equation 3 calculates the *semantic similarity* between two concepts based on the path length (more distance in the graph means less semantic similarity), and the depth of the subsumed concept (common ancestor) in the shortest path between the two concepts (the deeper in the hierarchy, the closer the meaning of the concepts). We note, however, that we provide Equation 3 just as an example. As such, we refer the interested reader to the work by Li et al. (2003) for further details on Equation 3, and we stress that alternative approaches can be used to replace this equation. There is no universal measure for semantic similarity, and it usually depends on the structure of the taxonomy, amongst other things. Different contexts and different taxonomies may require different approaches and equations. Similarly, different systems may also prefer different equations for their own taxonomies.

Plan Empowerment. When the *capability* of performing similar actions is relevant, we say measures of *empowerment* are needed, as opposed to *semantic* similarity measurements. For example, driving a truck and driving a car may be similar. However, if α is capable of driving a truck then it will be capable of driving a car, but not vice versa. As such, when considering the capabilities of performing actions we are not only interested in similar actions, but whether one action empowers another. As illustrated by the truck/car driving example, empowerment measures need not be symmetric. We say, while similarity measures are computed by considering taxonomies (based on the *is-a* relation), empowerment measures are computed by considering meronomies (based on the *empowered-by* relation).

To compute the empowerment measure between two nodes of a meronomy, we make use of the OpinioNet algorithm (Osman et al., 2010). OpinioNet highlights the importance of the structural relations (based on the *part-of* relation) linking related entities and their use in indicating the flow of opinions from one entity to another. OpinioNet's mechanism allows a single agent, after it has formed opinions about a few entities (nodes) in a structural graph, to be able to infer its opinion concerning unfamiliar related entities. For example, say a new coffee machine is now out in the market and it has not been rated yet. What can an interested customer infer about this new item's reputation? Clearly, the reputation of other coffee machines of the same brand, or even other products of this brand in general, could be of help here. Hence, OpinioNet highlights the need for representing the structural relations linking those entities together. A structural graph may then be used, and the brand may be represented as one node in this graph, the brand's coffee machines as a child node to the former, the new coffee machine model as a child node to the latter, and so on. Such a representation will not only facilitate the flow of opinions amongst related entities, but also permit raters to choose the granularity level at which they would prefer to leave their opinions at. For instance, while one agent might be interested in rating this specific model in the future, it might also be interested in providing a rating for the brand's coffee machines in general.

Now consider, for example, the simple meronomy of actions on music performance presented by Figure 2. The arrows describe the *empowered-by* relations. For instance, "Practice Scales" may be thought of as empowered by "Practice Piano". In other words, for one to practice their piano, they should already know how to practice the scales (or if one is capable of practicing the piano, then they are capable of practicing the scales). In general, we say the child node is empowered by the parent node.

In this paper, we map the *empowered-by* relation to the *part-of* relation in order to make use of OpinioNet when working with meronomies, and we interpret the opinions of OpinioNet to describe the capability of performing an action, specified as a node in a meronomy. In other words, if a node in a meronomy receives the best opinion possible (specified as the proba-

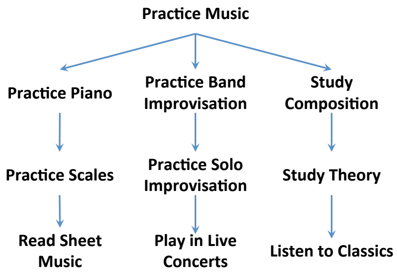


Figure 2: A meronomy describing actions about music practice

bility distribution $\mathbb{B}$), this is interpreted as the full capability of performing that node, or action. The propagation of an opinion from a node ϕ to a node ϕ' is then interpreted as deducing what the capability of performing ϕ' is, given the capability of performing ϕ. We then say the difference between the original opinion at ϕ (describing capability of performing ϕ) and the propagated opinion at ϕ' (describing the deduced capability of performing ϕ', given the capability of performing ϕ) specifies the empowerment of ϕ on ϕ'. For example, if the full capability of ϕ implies a strong capability of ϕ' (where the distance between those two capabilities would be very small), then this describes that ϕ greatly empowers ϕ'.

Let us say "Practice Piano" of Figure 2 receives the best opinion possible (describing the full capability of practicing the piano). Propagating this opinion to other nodes in the meronomy, OpinioNet can help us deduce that one is also very much capable of practicing the scales (the distance between the best opinion possible at "Practice Piano" and the propagated opinion to "Practice Scales" is 0.1), but only half as capable when it comes to practicing band improvisation (the distance between the best opinion possible at "Practice Piano" and the propagated opinion to "Practice Band Improvisation" is 0.5).

Formally, we say the empowerment of ϕ on ϕ' is:

$$\phi \rhd \phi' = 1 - \left| emd(\mathbb{B}, opinioNet(\mathcal{M}, \mathbb{B}, \phi, \phi') \right| \tag{4}$$

where *opinioNet* is a function that returns the propagated opinion at ϕ' (describing the capability of performing ϕ') by propagating the best opinion possible $\mathbb{B}$ from ϕ (describing the full capability of performing ϕ) in meronomy $\mathcal{M}$ following the OpinioNet propagation algorithm of Osman et al. (2010); and *emd* is the earth mover's distance that calculates the distance (whose range is $[1, 0]$) between two probability distributions (Rubner et al., 1998).[2]

We note that as long as a meronomy does not change, empowerment measures remain fixed. Empowerment measures between terms may then be computed in advance (or when the meronomy changes) for every pair of nodes.

Based on Equation 4, we define the empowerment of plan P on plan P' accordingly:

$$Emp(P, P') = \min_{a' \in P'}\{\max_{a \in P}\{a \rhd a'\}\} \tag{5}$$

Note that we use the 'min-max' operators, following the same reasoning as that of Equation 2.

2.1.3. Probability Distribution Update

Our proposed trust model is based on the idea that the information provided by past experiences can help us assess the outcome, or expectation, of a current experience. We say, given an experience μ, we need to update the probabilities of expectations according to the information presented by μ. To do so, we first update the probability of at least one single expectation $(X = x)$ with respect to μ. Then, we update the probability distribution over all possible expectations in a single step, following the minimum relative entropy approach. We explain these two steps in further detail below.

Updating the probability of a single expectation $X = x$. We say, if the information provided by experience μ suggests the increase in the probability of expectation $X = x$, then the amount by which this probability is increased should be dependent on the relevance of the experience μ in the context of updating the probabilities of expectations. Furthermore, we design this increase in such a way that avoids unstable behaviour: that is,

[2] If probability distributions are viewed as piles of dirt, then the earth mover's distance measures the minimum cost for transforming one pile into the other. This cost is equivalent to the 'amount of dirt' times the distance by which it is moved, or the distance between elements of the probability distribution's support. The range of *emd* is $[0, 1]$, where 0 represents the minimum distance and 1 represents the maximum possible distance.

even if an experience is fully relevant, a *single* experience cannot result in *considerable* change in the probability in question. In other words, considerable changes in expected behaviour can only be the result of information learned from an accumulation of experiences (as opposed to information learned from a single experience). As such, we update the probability of expectation $X = x$ accordingly:

$$p^{t_\mu}(X = x) = p^{t_{last}}(X = x) + (1 - p^{t_{last}}(X = x)) \cdot \epsilon \cdot R_\mu(\mathbb{P}(X)) \quad (6)$$

where $p^{t_{last}}(X = x)$ specifies the probability of expectation x at time t_{last} (the time when the probability of x was last updated), $p^{t_\mu}(X = x)$ specifies the probability of expectation x after considering the information provided by μ at time t_μ (the time of experience μ), ϵ specifies the maximum percentage of increase that a probability is allowed (which controls unstable behaviour), and $R_\mu(\mathbb{P}(X))$ specifies the relevance of the experience μ in the context of updating the probabilities of expectations ($\mathbb{P}(X)$).

Equation 6 states how to calculate the probability of expectation $X = x$ after considering experience μ, whose relevance in this context is $R_\mu(\mathbb{P}(X))$. The update is based on increasing the latest probability $p^{t_{last}}(X = x)$ (or the probability of the expectation x that was calculated at an earlier point in time, $t_{last} < t_\mu$, and that did not consider the experience μ) by a fraction ($\epsilon \cdot R_\mu(x)$) of the maximum potential increase $(1 - p^{t_{last}}(X = x))$. This fraction is defined by a fixed percentage (ϵ) tuned by the relevance of μ ($R_\mu(\mathbb{P}(X))$). In other words, if $R_\mu(\mathbb{P}(X)) = 1$, or if μ was maximally relevant, then $p^{t_{last}}(X = x)$ is increased by ϵ percent of $1 - p^{t_{last}}(X = x)$. For instance, if the probability of x was 0.6, the maximum percentage of increase is 0.1, and the experience μ updating the probability of x has a relevance of 1, then the new probability of x becomes $0.6 + (1 - 0.6) \cdot 0.1 \cdot 1 = 0.64$. We note that $\epsilon \in [0, 1]$ and $R_\mu(\mathbb{P}(X)) \in [0, 1]$. We also note that the ideal value of ϵ should be closer to 0 than to 1 so that even if an experience is very relevant, a *single* experience does not result in considerable changes in expected behaviour.

Updating the probability distribution over all possible expectations. With the probability of one (or more) expectation(s), we update the probability distribution over all possible expectations in one single and simple step, following the entropy-based approach of Sierra and Debenham (2005). The entropy-based approach updates a distribution $\mathbb{P}^{t_{last}}(X)$ (where t_{last} describes the time when the probability has been updated last) into $\mathbb{P}^{t_\mu}(X)$ (where t_μ describes the time of the experience μ resulting in this update) such that: (1) the new distribution satisfies the constraint(s) imposed by the new point(s) (that is, if the probability of expectation $X = x$ was updated to $p^{t_\mu}(X = x)$, then the new distribution's value at x should be equivalent to $p^{t_\mu}(X = x)$), and (2) the new distribution's relative entropy with respect to $\mathbb{P}^{t_{last}}(x)$ is minimal. In other words, we look for distributions that satisfy the updated probabilities of expectations and are at a minimal distance from the original distribution $\mathbb{P}^{t_{last}}(X)$ (as the relative entropy is a measure of the difference between two probability distributions). This is described accordingly:

$$\begin{aligned} \mathbb{P}^{t_\mu}(X) &= \arg\min_{\mathbb{P}(X)} \sum_i p^{t_{last}}(X = i) \log \frac{p^{t_{last}}(X = i)}{p(X = i)} \\ &\text{such that} \quad \{p(X = x) = p^{t_\mu}(X = x), \ldots\} \end{aligned} \quad (7)$$

where $\{p(X = x) = p^{t_\mu}(X = x), \ldots\}$ specifies that one or more constraints of the form $p(X = x) = p^{t_\mu}(X = x)$ need to be satisfied.

2.1.4. Decay of Information

An important notion in our proposal is the notion of information decay. We say the integrity of information decreases with time. That is, the information provided by a probability distribution should lose its value over time and decay towards a default value. We refer to this default value as the *decay limit distribution*. Calculating the decay limit distribution is outside the scope of this paper, although we argue that one may have background knowledge concerning the expected integrity of a precept as $t \to \infty$. Such background knowledge may be expressed in terms of an individual's knowledge, and is represented as a decay limit distribution $\mathbb{D}_x$, where x describes the context.[3]

[3] For example, when calculating the probability distribution of the expected outcome for the goal of submitting one's work on time, one might expect the default probability of submitting on time to be very high for computer science conferences, whereas the default probability of submitting on time will be much lower for an internal technical report, for instance.

In summary, given a distribution, $\mathbb{P}$, and a decay limit distribution $\mathbb{D}_x$, $\mathbb{P}$ decays from one point in time (t') to a later point in time (t, where $t > t'$) by:

$$\mathbb{P}^{t' \rightsquigarrow t} = \Lambda(\mathbb{D}_x, \mathbb{P}^{t'}) \tag{8}$$

where Λ is the *decay function* satisfying the property: $\lim_{t\to\infty} \mathbb{P}^{t' \rightsquigarrow t} = \mathbb{D}_x$.

One possible definition for Λ could be:

$$\mathbb{P}^{t' \rightsquigarrow t} = \nu^{\Delta_{t,t'}} \cdot \mathbb{P}^{t'} + (1 - \nu^{\Delta_{t,t'}})\mathbb{D}_x \tag{9}$$

where ν is the decay rate, and:

$$\Delta_{t,t'} = \begin{cases} 0 & \text{, if } t - t' < \omega \\ 1 + \dfrac{t - t'}{t_{max}} & \text{, otherwise} \end{cases}$$

The definition of $\Delta_{t,t'}$ above serves the purpose of establishing a minimum *grace* period, determined by the parameter ω, during which the information does not decay, and that once reached the information starts decaying. The parameter t_{max}, which may be defined in terms of multiples of ω, controls the *pace of decay*. The main idea behind this is that after the grace period, the decay happens very slowly; in other words, $\Delta_{t,t'}$ decreases very slowly.

Of course, one might also think of either the decay function or the decay limit distribution to be also a function of time, if the context requires this.

2.2. Probability of an Advice Realising its Goal

To help assess the trustworthiness of an advice, CONSUASOR calculates the expectation of the advice's outcome, specified as the probability of an advice realising its goal: $p(Observe(\Gamma,[P_\eta]G) \mid Commit(\rho,[P_\eta]G))$. Note that we are interested in the probability of the advice being the responsible of the fulfilment of the goal; that is, we rule out accidental fulfilment's of the goal. As such, CONSUASOR considers that an advice, which recommends plan P for η, fulfils its goal G if three things happen:

1. **Compliance: η agrees to perform P, specified as $Commit(\eta, P)$.** This describes η's compliance with following ρ's recommended advice $[P_\eta]G$.[4]
2. **Honour: η performs P, specified as $Observe(\beta, P_\eta)$.** This describes η's honour in following its own commitment to perform the plan it has agreed to.[5]
3. **Goal Realisation: Plan P realises goal G, specified as $Observe(\alpha, G)$.**

Accordingly, a good adviser is then one who not only knows the causal relation between plans and goals, but also knows about the advisee's compliance and honour (that is, it can correctly guess what plan will the advisee agree to and perform). In other words, a good adviser becomes one who modifies his advice, taking into consideration its knowledge about the the advisee's compliance and honour in order to ensure the intended goal is fulfilled.

Given that we define an advice realising its goal as a conjunction of three events where the advisee agrees to and performs the recommended plan and the performed plan realises the intended goal, the probability of an advice realising its goal becomes:

$$\begin{aligned} & p(Observe(\Gamma,[P_\eta]G) \mid Commit(\rho,[P_\eta]G)) = \\ & \quad p(Observe(\alpha,G) \text{ and } Observe(\beta,P_\eta) \text{ and } Commit(\eta,P) \\ & \qquad \mid Commit(\rho,[P_\eta]G)) \end{aligned} \tag{10}$$

[4] While CONSUASOR does not dwell on the motivation behind one being compliant with an advice, we note that compliance may be influenced by willingness, capabilities, and/or obligations. Social commitments, for instance, may result in obligations, and hence compliance. One may be obliged to accept their superior's advice. Personal interest in the intended goal and the trustworthiness in the adviser may also result in the willingness to accept the recommended plan. In such a case, the probability of compliance will depend on the trust on the adviser, although the probability of compliance will also be used for computing the trustworthiness of advice, and hence, the adviser. This could result in non-linear equations, which will require alternative computational approaches, such as following fixed point methods. We leave this for future work.

[5] While CONSUASOR does not dwell on the motivation behind one honouring its commitments, we note that honour may again be influenced by willingness, capabilities, and/or obligations. For instance, one may be willing to perform a given plan, and hence accept it, only to discover later on that they are in fact incapable of performing the accepted plan. Social commitments may also force one to accept a plan that they are in fact not to willing to perform.

where $\Gamma = \{\alpha, \beta\}$. In other words, Γ is the coalition that observes that the advise fulfills its goal. Note that Γ's observation is deduced from α and β's observations. If β observes the plan being performed and α observes the goal being realised, then we can say that the coalition Γ observes the advice fulfilling its goal. This is similar to coalition logic (Ågotnes et al., 2008), a special type of modal logic. Properties in coalition logic can describe what a coalition, or a group of agents, may achieve as a whole. In other words, it can describe what the group is capable of. In Equation 10, the observation on the left hand side describes what the coalition $\Gamma = \{\alpha, \beta\}$ observes, and it is deduced from the individual observations on the right hand side of the equation.

Given Equation 10 for the probability of an advice fulfilling its goal, we can derive the following proposition:[6]

Proposition 1 (Probability of an advice realising its goal I)**.** *The probability of an advice $[P_\eta]G$ realising its goal is the product of the probability of compliance ($p(Commit(\eta,P) \mid Commit(\rho,[P_\eta]G))$), the probability of honour ($p(Observe(\beta,P_\eta) \mid Commit(\eta,P)$ and $Commit(\rho,[P_\eta]G))$), and the probability of goal realisation ($p(Observe(\alpha,G) \mid Observe(\beta,P_\eta)$ and $Commit(\eta,P)$ and $Commit(\rho,[P_\eta]G))$). That is:*

$$\begin{aligned}
&p(Observe(\gamma,[P_\eta]G) \mid Commit(\rho,[P_\eta]G)) = \\
&\quad p(Commit(\eta,P) \mid Commit(\rho,[P_\eta]G)) \cdot \\
&\quad p(Observe(\beta,P_\eta) \mid Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G)) \cdot \\
&\quad p(Observe(\alpha,G) \mid Observe(\beta,P_\eta) \text{ and } Commit(\eta,P) \text{ and } \\
&\qquad\qquad Commit(\rho,[P_\eta]G))
\end{aligned}$$

Proof.

$$\begin{aligned}
&p(Commit(\eta,P) \mid Commit(\rho,[P_\eta]G)) \cdot \\
&p(Observe(\beta,P_\eta) \mid Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G)) \cdot \\
&p(Observe(\alpha,G) \mid Observe(\beta,P_\eta) \text{ and } Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G))
\end{aligned}$$

$$\begin{aligned}
=&\frac{p(Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G))}{p(Commit(\rho,[P_\eta]G))} \cdot \\
&\frac{p(Observe(\beta,P_\eta) \text{ and } Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G))}{p(Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G))} \cdot \\
&\frac{p(Observe(\alpha,G) \text{ and } Observe(\beta,P_\eta) \text{ and } Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G))}{p(Observe(\beta,P_\eta) \text{ and } Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G))}
\end{aligned}$$

$$=\frac{p(Observe(\alpha,G) \text{ and } Observe(\beta,P_\eta) \text{ and } Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G))}{p(Commit(\rho,[P_\eta]G))}$$

$$=p(Observe(\alpha,G) \text{ and } Observe(\beta,P_\eta) \text{ and } Commit(\eta,P) \mid Commit(\rho,[P_\eta]G))$$

□

Of course, when ρ recommends P to η for realising goal G, η may agree to a variation of the recommended plan ($P' \neq P$), perform a variation of what it has agreed upon ($P'' \neq P'$), and eventually, a variation of the intended goal ($G' \neq G$) may be realised. As such, the probability distribution that describes all possible expectations of the advice $[P_\eta]G$ is defined accordingly, by considering all possible plans agreed upon and performed, as well as all possible realised goals:

$$\begin{aligned}
&\mathbb{P}(Observe(\gamma,[P_\eta]X) \mid Commit(\rho,[P_\eta]G)) = \\
&\{p(Observe(\gamma,[P_\eta]X{=}G') \mid Commit(\rho,[P_\eta]G)), \ldots\}_{\forall G' \in \mathcal{G}}
\end{aligned} \tag{11}$$

where X is a variable and the range over which X varies is $\mathcal{G}$, which describes the set of all possible goals that may be realised.

2.2.1. Assumptions

Two assumptions are made by CONSUASOR. First, we say the plan that is performed is independent of the original recommended plan, yet dependent on the accepted (or committed to) plan. For example, if the tutor recommends the plan "practice three times a week", and the student commits to "practicing two times a week". Then what the student actually performs will not be dependent on what the tutor has recommended, but on what it has committed to.

Second, we say the goal that is realised is independent of the original recommended plan and independent of the accepted (or committed to) plan, yet dependent on the performed plan. This is because the realised goal is the outcome of a causal relation that links the performed plan to the realised goal. For example,

[6] The proof for Proposition 1, which is presented below, is a straightforward proof that makes use of the conditional probability definition (or axiom): $p(A|B) = p(A \text{ and } B)/p(B)$.

assume that the plan “do not submit assignment” results in the goal “fail course”. Then whatever the tutor recommended, and whatever the student accepted (or committed to do), if the student eventually does not submit his assignment then he will fail the course.

We formally define these two assumptions next.

Assumption 1 (Conditional Independence I)**.** *The events* $Observe(\beta,P_\eta)$ *and* $Commit(\rho,[P_\eta]G)$ *are conditionally independent given* $Commit(\eta,P)$*. That is:*

$$p(Observe(\beta,P_\eta) \mid Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G)) = p(Observe(\beta,P_\eta) \mid Commit(\eta,P))$$

Assumption 1 states that, given knowledge that η agreed to P ($Commit(\eta,P)$), knowledge of whether ρ recommended $[P_\eta]G$ ($Commit(\rho,[P_\eta]G)$) provides no information on the likelihood of η performing P ($p(Observe(\beta,P_\eta))$, and the knowledge of η performing P ($p(Observe(\beta,P_\eta))$ provides no information on the likelihood of ρ recommending $[P_\eta]G$ ($Commit(\rho,[P_\eta]G)$).

We note that, depending on the context or the character of β, Assumption 1 might not always hold. For example, if ρ suggested that β should practice his piano three times a week, β might commit to practicing twice a week, and later on try to follow ρ’s advice by practicing three times a week. In this paper, we keep the CONSUASOR model simple by making this assumption (Assumption 1), and we present a brief discussion (see Footnotes [12] and [16] of Section 2.2.4) that illustrate how the model may be adapted to consider situations when Assumption 1 does not hold.

Assumption 2 (Conditional Independence II)**.** *The events* $Observe(\alpha,G)$ *and* ($Commit(\eta,P)$ *and* $Commit(\rho,[P_\eta]G)$) *are conditionally independent given* $Observe(\beta,P_\eta)$*. That is:*

$$p(Observe(\alpha,G) \mid Observe(\beta,P_\eta) \text{ and } Commit(\eta,P) \text{ and } Commit(\rho,[P_\eta]G)) = p(Observe(\alpha,G) \mid Observe(\beta,P_\eta))$$

Assumption 2 states that, given knowledge that η performed P ($Observe(\beta,P_\eta)$), knowledge of whether ρ recommended $[P_\eta]G$ and η agreed to P ($Commit(\eta,P)$ and $Commit(\rho,[P_\eta]G)$) provides no information on the likelihood of goal G being realised ($Observe(\alpha,G)$), and the knowledge of goal G being realised ($Observe(\alpha,G)$) provides no information on the likelihood of ρ recommending $[P_\eta]G$ and η agreeing to P ($Commit(\eta,P)$ and $Commit(\rho,[P_\eta]G)$).

Given Assumptions 1 and 2, the probability of an advice realising its goal may then be simplified accordingly:[7]

Proposition 2 (Probability of an advice realising its goal II)**.** *The probability of an advice realising its goal, under Assumptions 1 and 2, is:*

$$\begin{aligned} p(Observe(\gamma,[P_\eta]G) \mid Commit(\rho,[P_\eta]G)) = {} & p(Commit(\eta,P) \mid Commit(\rho,[P_\eta]G)) \cdot \\ & p(Observe(\beta,P_\eta) \mid Commit(\eta,P)) \cdot \\ & p(Observe(\alpha,G) \mid Observe(\beta,P_\eta)) \end{aligned}$$

2.2.2. Notation

For simplification, in the remainder of this paper we will use the notation $\mathbb{P}^t_{\mathrm{C}}(X|[P^\rho_\eta]G)$ to describe the probability distribution of η committing to a plan X at time t given that the advice $[P_\eta]G$ has been recommended by ρ. In other words, $\mathbb{P}^t_{\mathrm{C}}(X|[P^\rho_\eta]G) = \mathbb{P}^t(Commit(\eta,X) \mid Commit(\rho,[P_\eta]G))$. We refer to this as the probability distribution on *compliance*, which describes the compliance of η with ρ’s recommended advice $[P_\eta]G$.

Similarly, we use the notation $\mathbb{P}^t_{\mathrm{H}}(X|P_\eta)$ to describe the probability distribution of η performing plan X at time t given that it has committed to the plan P. In other words, $\mathbb{P}^t_{\mathrm{H}}(X|P_\eta) = \mathbb{P}^t(Observe(\beta,X_\eta) \mid Commit(\eta,P))$. We refer to this as the probability distribution on *honour*, which describes η’s honour in performing the plan it has committed to perform.

We also use the notation $\mathbb{P}^t_{\mathrm{R}}(X|P)$ to describe the probability distribution of realising goal X at time t given that P has been performed. In other words, $\mathbb{P}^t_{\mathrm{R}}(X|P) = \mathbb{P}^t(Observe(\alpha,X) \mid Observe(\beta,P_\eta))$. We refer to this as the probability distribution on *goal realisation*, which describes the realisation of goal G as a result of performing plan P.

[7] The proof for Proposition 2 is straightforward: By applying the rules of Assumptions 1 and 2, Proposition 1 is reduced to Proposition 2.

Finally, we will use the notation $\mathbb{P}^{t}_{\mathrm{C|H|R}}(X|Y)$ to refer to *any* of the probability distributions on compliance, honour, or goal realisation.

2.2.3. Initialisation

In this paper, we provide a method that calculates a probability distribution $\mathbb{P}^{t}_{\mathrm{C|H|R}}(X|Y)$ incrementally, by updating the probability distributions on compliance, honour, and goal realisation for every experience in the history of experiences H. However, whether the update of the probability distributions is triggered every time a new experience is recorded, whether it is triggered on demand every time a trust measure needs to be calculated, or whether it is executed periodically is an implementation detail that we do not discuss in our presentation of the model (although Section 2.4 does provide one approach that we implement and use in our evaluation). What is important though, is that when a probability distribution is updated, it is updated by considering the oldest experiences first. In other words, the temporal order of the experiences decides which experience needs to be considered first for updating the probability distributions.

At the initial time t_I when no experiences have been considered yet in calculating the distribution, the initial value of the probability distribution may be domain dependent. Alternatively, for domain independent values, one can choose the uniform distribution $\mathbb{F} = \{p(x_1) \mapsto 1/n, \ldots, p(x_n) \mapsto 1/n\}$[8] to describe ignorance. That is, $\mathbb{P}^{t_I}_{\mathrm{C|H|R}}(X|Y) = \mathbb{F}$. As new experiences are considered, the probability distribution gets reshaped, or updated, by incorporating the information learned from these experiences. We note that the support of $\mathbb{P}_{\mathrm{C}}(X|[P^{o}_{\eta}]G)$ and $\mathbb{P}_{\mathrm{H}}(X|P_{\eta})$ is the set of all plans $\mathcal{P}$, and the support of $\mathbb{P}_{\mathrm{R}}(X|P)$ is the set of all possible goals $\mathcal{G}$.[9]

2.2.4. Update of the probability distributions on compliance, honour, and goal realisation

To calculate the expected outcome of an advice, as defined by Equation 11, CONSUASOR needs to keep track of the probabilities on compliance, honour, and goal realisation. While the previous section presented the initial values of the probability distributions on compliance, honour, and goal realisation, this section illustrates how these distributions are updated when considering new experiences, as these distributions are learnt from similar past experiences.

Every time an experience μ is considered for updating a probability distribution $\mathbb{P}^{t_{last}}_{\mathrm{C|H|R}}(X|Y)$, the following steps are taken:

1. The relevance of μ w.r.t. $\mathbb{P}_{\mathrm{C|H|R}}(X|Y)$ is calculated.

The relevance is calculated differently for each context:

a) Compliance: Relevance of μ with respect to $\mathbb{P}_{\mathrm{C}}(X|[P^{o}_{\eta}]G)$.

Given a past experience where η committed to P'' when $[P'_{\eta}]G'$ was advised, the question then is: how much relevant is this past experience in informing us about the possible commitment of η given the current advice $[P_{\eta}]G$? To define this measure of relevance, we use analogical reasoning by stating that η might behave similarly to how it behaved in experience μ if μ's advice ($[P'_{\eta}]G'$) is similar to the current advice ($[P_{\eta}]G$). As an advice is composed of both a plan and a goal, we compute the similarity between advice $[P'_{\eta}]G'$ and $[P_{\eta}]G$ by relying on the semantic similarity between the advice's goals ($Sim(G', G)$) and plans ($Sim(P', P)$). For instance, if η has been accepting or rejecting plans that have been recommend for a given goal G, the it will most likely behave similarly when a plan for a similar goal is recommended. Similarly, if η accepts or rejects a plan in the past (whether the reason to accept or reject was based on its willingness or capability of performing the plan),

[8] n is the number of elements in the support of the distribution.

[9] There are drawbacks to setting the support of the distributions to the set of all plans or goals. For example, in some cases, the set of all possible goals/plans might be dynamic. In other cases, the set of all possible goals/plans might be massive in size, which would result in an extremely inefficient algorithm. One approach to address such issues is to have the sets of plans and goals learned from experiences. As such, the initial set of goals would be $\mathcal{G} = \{unknown\}$ and the initial set of plans would be $\mathcal{P} = \{unknown\}$, where the probability of *unknown* is initially 1. This describes that with the lack of any experience, one can expect some unknown thing to happen (specified as *unknown*). Then, as experiences are formed, they add their own accepted and/or performed plans to $\mathcal{P}$, and their realised goals to $\mathcal{G}$. In other words, the sets of possible plans and goals are updated by learning from experiences. We note that while this is the implementation of our choice, this paper assumes fixed sets of plans and goals to keep the presentation of the CONSUASOR model relatively simple.

then it will most likely behave similarly when a similar plan is recommended in the future.

As $\mathbb{P}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G)$ describes the expectation of η's compliance with ρ's recommended advice $[P_{\eta}]G$, we only consider experiences where η has been given advice in the past by similar advisers (where the similarity of advisers is specified by $S_A(\rho, \rho')$, and is defined shortly), in order to learn from η's compliances in similar past scenarios.

We say, given a new experience $\mu = \langle Commit(\rho', [P'_{\eta}]G')_t$, $Commit(\eta, P'')_{t'}, \_, \_\rangle$,[10] such that $S_A(\rho, \rho') \geq \zeta_r$, we calculate μ's relevance to $\mathbb{P}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G)$ as follows:

$$R_{\mu}(\mathbb{P}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G)) = \frac{\zeta_g \cdot Sim(G', G) + \zeta_p \cdot Sim(P', P)}{\zeta_g + \zeta_p} \qquad (12)$$

where $\zeta_g, \zeta_p \in [0, 1]$ are parameters that specify the importance of each similarity measure, and ζ_r of the condition $S_A(\rho, \rho') \geq \zeta_r$ describes the similarity threshold for advisers.

The similarity between two advisers is calculated by comparing their past advice, and it is based on the rationale that two advisers ρ and ρ' are similar if they recommend similar plans for similar goals.[11] As such, we aggregate the similarity of ρ and ρ''s recommended plans, where the weight given to the similarity of each pair of plans is defined by the similarity of the goals for which these plans were recommended:

$$S_A(\rho, \rho') = \frac{\displaystyle\sum_{\forall [P_{\eta'}]G \in \mathcal{A}(\rho), [P'_{\eta''}]G' \in \mathcal{A}(\rho')} Sim(P, P') \cdot Sim(G, G')}{\displaystyle\sum_{\forall [P_{\eta'}]G \in \mathcal{A}(\rho), [P'_{\eta''}]G' \in \mathcal{A}(\rho')} Sim(G, G')} \qquad (13)$$

where $\mathcal{A}(\rho) = \{[P_{\eta}]G \mid \langle Commit(\rho, [P_{\eta}]G)_t, \_, \_, \_\rangle \in H\}$ describes the set of advice that ρ has given in the past. Note that following Equation 13, it is sufficient for one pair of advice to share very similar goals for the similarity of their advisers to be dominated by the similarity of the plans of that specific pair of advice. Alternative approaches for calculating the similarity of advisers may also be adopted, such as the collaborative filtering techniques of recommender systems Linden et al. (2003).

In the general case, we say that if the similarity between the current adviser ρ and the adviser ρ' of μ is greater than a certain threshold ζ_r, specified as $S_A(\rho, \rho') \geq \zeta_r$, then the experience μ is considered. If $\zeta_r = 1$, then we are *only* considering experiences where advice has been suggested to η by ρ, which means that the probability will describe the compliance of η with ρ's advice. If $\zeta_r = 0$, then we are considering experiences where advice has been suggested to η by *any* adviser, which means that the probability will describe the compliance of η in general. If $0 \leq \zeta_r \leq 1$, then we are considering experiences where advice have been suggested to η by advisers that share with ρ a similarity level of ζ_r.

b) Honour: Relevance of μ with respect to $\mathbb{P}_{\mathrm{H}}(X|P_{\eta})$.

Given a past experience where η committed to P' and then performed P'', the question then is: how much relevant is this past experience in informing us about what η will perform if it has committed to P? To define this measure of relevance, we again use analogical reasoning by stating that η might behave similarly to how it behaved in experience μ if the plan it committed to in μ (P') is similar to the plan it currently committed to (P). However, the similarity between P' and P is not based on semantic similarity as above, but on a measure of empowerment. This is because a necessary condition for performing a plan is to be *capable* of performing the plan. As such, it is relevant to know how η behaved when it committed to a plan

[10] In this paper, we use the underscore symbol '_', as in Prolog, to refer to an anonymous variable and it means "any term".

[11] Of course, alternative methods that consider additional information, such as roles and context, may be used for calculating the similarity of advisers. One alternative approach would be to calculate the similarity between the advisers' roles. For instance, if two advisers are music teachers, then they should be considered much more similar than a music teacher and a mechanic. Alternatively, the context may also be informative. For example, someone may be considered a good tutor at one of the top conservatoires, but a poor tutor for children in less-developed countries as the adviser may not be capable of taking into consideration the needs and constraints of those children. To keep this paper simple, however, we define adviser's similarity through the similarity of their past plans and goals only. Although we confirm that Equation 13 may be modified to consider additional information, such as role and/or context similarity. One approach for considering role (or context) similarity is to define a role (or context) ontology and then use semantic similarity for calculating the similarity of roles (or contexts) in that ontology, which we leave for future work.

that empowers the plan under consideration. For instance, if it diverted much from it, then it is highly possible that it will divert from its current plan.

As $\mathbb{P}_{\mathrm{H}}(X|P_\eta)$ describes the expectation of η's honour in fulfilling what it has committed to, we only consider experiences where η has committed and performed plans in order to learn from η's past honour. We say given a new experience $\mu = \langle \_, Commit(\eta, P')_t, Observe(\beta, P''_\eta)_{t'}, \_ \rangle$, we define μ's relevance to $\mathbb{P}_{\mathrm{H}}(X|P_\eta)$ as follows:[12,13]

$$R_\mu(\mathbb{P}_{\mathrm{H}}(X|P_\eta)) = Emp(P', P) \qquad (14)$$

c) Goal realisation: Relevance of μ with respect to $\mathbb{P}_{\mathrm{R}}(X|P)$. Given a past experience where G' was realised as a result of performing plan P', the question then is: how much relevant is this past experience in informing us about what will be realised if the current plan P is performed? To define this measure of relevance, we again use analogical reasoning by stating that the higher the similarity of the plan that was performed in experience μ (P') to the currently performed plan (P), then the higher the probability of G'. Note that unlike the case on honour above, we now use the semantic similarity between plans ($Sim(P', P)$) as opposed to plan empowerment. This is because the capability of performing a plan is no longer an issue as plans have already been performed (and observed).

As $\mathbb{P}_{\mathrm{R}}(X|P)$ describes the expectation of realising a goal given that plan P was performed, we need to focus on the goals that may be realised as a result of an *already performed* plan. We assume the identity of who performed the plan is irrelevant. As such, and unlike the cases on compliance and honour above, we look at all experiences and not just those where η has performed the plan.[14] We say given a new experience $\mu = \langle \_, \_, Observe(\beta, P'_{\eta'})_t, Observe(\alpha, G')_{t'} \rangle$, we calculate its relevance to $\mathbb{P}_{\mathrm{R}}(X|P)$ as follows:

$$R_\mu(\mathbb{P}_{\mathrm{R}}(X|P)) = Sim(P', P) \qquad (15)$$

2. The distribution $\mathbb{P}^{t_{last}}_{\mathrm{C|H|R}}(X|Y)$ is retrieved and decayed. If $\mathbb{P}_{\mathrm{C|H|R}}(X|Y)$ has not been updated by any experiences yet, then the initial distribution is retrieved ($\mathbb{P}^{t_{last}}_{\mathrm{C|H|R}}(X|Y) = \mathbb{P}^{t_I}_{\mathrm{C|H|R}}(X|Y)$). Otherwise, the latest probability distribution is retrieved ($\mathbb{P}^{t_{last}}_{\mathrm{C|H|R}}(X|Y)$) and the distribution is decayed according to the decay approach of Section 2.1.4. This allows us to take into consideration how the information represented by a probability distribution loses its value over time. Note that we use the notation t_{last} to refer to the time when the distribution in question has been updated last, t_μ to refer to the time the experience μ occurred, and $\mathbb{P}^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{C|H|R}}(X|Y)$ to refer to the decayed distribution. Also note that $\mathbb{P}^{t_{last}}_{\mathrm{C|H|R}}(X|Y)$ is not decayed to the current time, but to the time that the experience μ occurred (t_μ), since the objective of this update is to update $\mathbb{P}_{\mathrm{C|H|R}}(X|Y)$ with respect to μ.

Concerning the time the experience μ occurred, which we refer to as t_μ, recall that an experience μ has four timestamps, as illustrated earlier in Section 2.1.1. The timestamp that represents the time μ occurred depends on the probability distri-

[12] If Assumption 1 does not hold — that is, if the probability of performing a plan is *dependent* on the recommended plan — then the relevance of an experience on the probability of honour should take into consideration the recommended plan. One suggested approach to achieve this is to have: $R_\mu(\mathbb{P}_{\mathrm{H}}(X|P_\eta)) = Emp(P''', P) \cdot Emp(P', P)$, where P''' is the recommended plan of experience μ. In other words, if β's performance will be influenced by both the recommended plan P''' and the committed to plan P', then it is important to assess the empowerment of both P''' and P' on the current plan in question, P.

[13] Alternative approaches may be followed when considering the capability of performing a plan, such as having $R_\mu(\mathbb{P}_{\mathrm{H}}(X|P_\eta)) = Emp(P'', P)$. This approach is based on the idea that past *actions* (where P'' is the plan that was performed in the past) provide a more accurate indication of capability. Note that in such a case, Assumption 1 has no impact on this relevance measure, as this measure is only affected by past actions (depicted through P'') as opposed to recommended and committed to plans. Whereas the impact of Assumption 1 on the relevance measure of Equation 14 was discussed above in Footnote [12].

[14] We assume that the causal relationship between performed plans and realised goals to be independent of who performs the plan. In other words, the same plan always results in the same goal, regardless of who performs the plan. This might not always be true, however. For example, one student may need to study one hour a day to pass the exam, while another might need to study much more than that. To address such cases, and in order to learn from similar past experiences, we need to look for experiences where advisees with similar profiles were observed performing some plans. This, however, requires a measure of similarity for advisees' profiles, whose definition is left for future work.

bution in question. In the case of updating the probability distribution on compliance ($\mathbb{P}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G)$), t_μ is the timestamp of the second element of μ, or the time η committed to a plan. In the case of updating the probability distribution on honour ($\mathbb{P}_{\mathrm{H}}(X|P_\eta)$), t_μ is the timestamp of the third element of μ, or the time η performed a plan. In the case of updating the probability distribution on goal realisation ($\mathbb{P}_{\mathrm{R}}(X|P)$), t_μ is the timestamp of the fourth element of μ, or the time a goal was realised.

3. The probability $p^{t_\mu}_{\mathrm{C|H|R}}(X=x|Y)$ is calculated.

The experience μ is used to decide which expectation needs to have its probability updated and how is this new probability calculated. The decision of which expectation should have its probability updated is based on analogical reasoning, as motivated earlier. For instance, if η committed to a very different plan than what was recommended in experience μ, and μ is highly relevant to the current scenario, then one can expect η to commit to a very different plan than what is currently recommended. Similarly, if η strongly diverted from the plan that it has committed to in μ, and μ is highly relevant to the current scenario, then one can expect η to strongly divert from the currently committed plan too, following a similar behaviour to μ. Similarly, if a goal was realised in experience μ, and the performed plan in μ is very similar to the plan under consideration, then the probability of that goal being realised now becomes higher. We implement this analogical reasoning next.

How a probability of an expectation $X=x$ is updated follows a similar approach for all three distributions on compliance, honour, and goal realisation. The probability of an expectation $X=x$ is updated based on the relevance of the experience μ, by applying the probability update of Equation 6 (Section 2.1.3):[15]

$$p^{t_\mu}_{\mathrm{C|H|R}}(X=x|Y) = \\ p^{t_{last}\rightsquigarrow t_\mu}_{\mathrm{C|H|R}}(X=x|Y) + (1 - p^{t_{last}\rightsquigarrow t_\mu}_{\mathrm{C|H|R}}(X=x|Y)) \cdot \epsilon \cdot R_\mu(\mathbb{P}_{\mathrm{C|H|R}}(X|Y))$$

where, following the analogical reasoning presented above, the probability update is carried out for an expectation $X=x$ if:

a) $|Sim(P,x) - Sim(P',P'')| \leq \chi_{\mathrm{C}}$, when updating the probability of committing to a plan ($p^{t_\mu}_{\mathrm{C}}(X=x|[P_\eta]G)$) given an experience $\mu = \langle Commit(\rho', [P'_\eta]G')_t, Commit(\eta, P'')_{t_\mu}, \_, \_\rangle$. That is, we consider an expectation $X=x$ based on its semantic distance to the recommended plan P in such a way that this distance is approximately equivalent (or less than a threshold χ_{C}) to the semantic distance between the plan that was recommended in experience μ and the plan that η committed to then.

b) $|Sim(P,x) - Sim(P',P'')| \leq \chi_{\mathrm{H}}$, when updating the probability of performing a plan ($p^{t_\mu}_{\mathrm{H}}(X=x|P_\eta)$) given an experience $\mu = \langle \_, Commit(\eta, P')_t, Observe(\beta, P''_\eta)_{t_\mu}, \_\rangle$. That is, we consider an expectation $X=x$ based on its semantic distance to the committed plan P in such a way that this distance is approximately equivalent (or less than a threshold χ_{H}) to the semantic distance between what η has committed to in experience μ and what it has performed then.[16]

c) $x = G'$, when updating the probability of a goal being realised ($p^{t_\mu}_{\mathrm{R}}(X=x|P)$) given an experience $\mu = \langle \_, \_, Observe(\beta, P'_{\eta'})_t, Observe(\alpha, G')_{t_\mu}\rangle$. Note that since goal similarity and plan similarity cannot be compared (as goals and plans are two distinct concepts), we cannot state that the semantic distance between the realised goal G' of μ and the currently expected goal need to be approximately

[15] The value of ϵ may be based on considering factors that are not related to the experience μ, and yet they may influence the probability in question. For example, when calculating the probability on compliance, ϵ may depend on social commitments. In other words, the stronger the social commitment between η and the current recommender ρ then the higher the probability that η will commit to ρ's current advice.

[16] If Assumption 1 does not hold — that is, if the probability of performing a plan is *dependent* on the recommended plan — then finding the relevant expectation $X=x$ should take into consideration the recommended plan. One suggested approach to achieve this is to modify the similarity condition accordingly: $|Sim(P'''',x) - Sim(P''',P'')| \cdot |Sim(P,x) - Sim(P',P'')| \leq \chi_{\mathrm{H}}$, where P'''' is the currently recommended plan and P''' is the recommended plan of experience μ. In other words, we consider the conjunction of: (1) the distance between what was committed to and performed in the past to what was committed to and may be performed now, and (2) what was recommended and performed in the past to what was recommended and may be performed now. The conjunction of these distances should then be less than the threshold χ_{H}.

equivalent to the semantic distance between the performed plan P' of μ and the plan P under consideration. Accordingly, we simply state that the higher the relevance of an experience μ (which was based on the similarity between the performed plan P' of μ and the plan P under consideration – Equation 15), the higher the probability of the goal G' of μ.

Note that more than one expectation may satisfy the condition of semantic equivalence, and as a result, the probability of more than one expectation may be updated.

4. The probability distribution $\mathbb{P}^{t_\mu}_{\text{C|H|R}}(X|Y)$ is calculated.

Given the probability $p^{t_\mu}_{\text{C|H|R}}(X = x|Y)$ (or a set of probabilities $\{p^{t_\mu}_{\text{C|H|R}}(X=x|Y), \ldots\}$) , we calculate the new probability distribution $\mathbb{P}^{t_\mu}_{\text{C|H|R}}(X|Y)$ by applying Equation 7 of Section 2.1.3, which follows the minimum relative entropy approach:

$$\mathbb{P}^{t_\mu}_{\text{C|H|R}}(X|Y) = \underset{\mathbb{P}(X)}{\arg\min} \sum_i p^{t_{last}\rightsquigarrow t_\mu}_{\text{C|H|R}}(X=i|Y) \log \frac{p^{t_{last}\rightsquigarrow t_\mu}_{\text{C|H|R}}(X=i|Y)}{p(X=i)}$$
$$\text{such that} \quad \{p(X=x) = p^{t_\mu}_{\text{C|H|R}}(X=x|Y), \ldots\}$$

A note on learning from past experiences. CONSUASOR is a trust model that calculates the probability of accepting a plan, the probability of executing a plan, and the probability of a goal being realised by learning from past experiences. It is true that things may change over time. For example, if η accepted a plan in the past, it may change its mind in the future and reject it. While many actions may be *possible*, some actions will be more *probable* than others. Learning from past experience entails that the probability of a given action (whether it was accepting a plan, executing a plan, or having a goal realised) should increase with the number of times that this same or similar action has occurred in the past. In fact, the proposed CONSUASOR model has been designed in such a way that the probability of an action (say accepting a plan) would increase a 'teeny-tiny bit' if that same action (say the same person has accepted the plan) has occurred 'once' in the past, and it will increase more as the number of times that the action occurred in the past increases. Furthermore, we note that the model also incorporates the notion of decay. In other words, if an action occurred 'a very long time ago', then that action will have much less impact than an action occurring more recently. This allows change to happen with time, such as accommodating for people changing their mind over time and starting to reject a plan they used to accept. The proposed model is designed to adjust to such situations. Additionally, if the action is more random (say the person is more random in accepting and rejecting a plan, that is, he sometimes accepts and sometimes rejects), then the model will learn this information as well from those past experiences, and all possible outcomes (such as accepting and rejecting a plan) will have more or less equal probabilities.

2.3. *Trust Computation*

With updated probability distributions on compliance, honour, and goal realisation, the main probability distribution $\mathbb{P}^t(\mathit{Observe}(\gamma,[P_\eta]X) \mid \mathit{Commit}(\rho,[P_\eta]G))$ is calculated via Equation 11, which we simply refer to as $\mathbb{P}^t(X|[P^\rho_\eta]G)$ for simplification. The question now is: How do we interpret such expectations? In other words: How do we calculate a *trust measure* given an expectation specified as a probability distribution?

Different trust equations can be implemented, depending on the particular context or interest. For example, a trust measure may be based on the preference of outcomes: given the preferences of possible outcomes, the trust measure will be higher when preferred outcomes are more likely to happen than less preferred outcomes. A trust measure may be based on the certainty of the expected outcomes, where entropy may be used to measure uncertainty: the higher the certainty of outcomes, then the higher the trust measure is, and vice versa. Sierra and Debenham (2006) presents a few approaches for calculating trust, including those that are based on preferences or certainty of outcomes.

One alternative straightforward approach, which we present next, is based on the distance between what the advice promised ($\mathbb{P}_{\mathbf{P}}(X|[P^\rho_\eta]G) = \{1, \text{if } X = G; 0, \text{otherwise}\}$) and what is expected to be achieved ($\mathbb{P}^t(X|[P^\rho_\eta]G)$). The trustworthiness of an advice $[P_\eta]G$ is then calculated accordingly:

$$\mathit{trust}^t([P^\rho_\eta]G) = 1 - \mathit{emd}(\mathbb{P}_{\mathbf{P}}(X|[P^\rho_\eta]G), \mathbb{P}^t(X|[P^\rho_\eta]G)) \qquad (16)$$

where the function *emd*, whose range is $[0, 1]$, calculates the earth mover's distance between two probability distributions.

Concerning the trustworthiness of an adviser ρ, we say a good adviser is one who provides good advice. As such, the trustworthiness of an adviser is defined in terms of the trustworthiness of his advice. That is, the trustworthiness of ρ on giving advice becomes an aggregation of $trust^t([P^\rho_\eta]G)$ for every advice $[P_\eta]G$ that ρ has given in the past:

$$trust^t(\rho) = \frac{\sum_{\forall [P_\eta]G \in \mathcal{A}(\rho)} trust^t([P^\rho_\eta]G)}{|\mathcal{A}(\rho)|} \tag{17}$$

where $\mathcal{A}(\rho) = \{[P_\eta]G \mid \langle Commit(\rho, [P_\eta]G)_{t'}, \_, \_, \_\rangle \in H\}$ describes the set of advice that ρ has given in the past ($t' < t$).

The trust equations presented above may be adapted to compute different trust measures. For instance, the trustworthiness of an adviser ρ with respect to a specific goal G may be computed in a similar fashion to Equation 17, where the only difference is in considering ρ's past advice on goal G as opposed to considering ρ's past advice in general. That is:

$$trust^t(\rho, G) = \frac{\sum_{\forall [P_\eta]G \in \mathcal{A}(\rho,G)} trust^t([P^\rho_\eta]G)}{|\mathcal{A}(\rho, G)|} \tag{18}$$

where $\mathcal{A}(\rho, G) = \{[P_\eta]G \mid \langle Commit(\rho, [P_\eta]G)_{t'}, \_, \_, \_\rangle \in H\}$ describes the set of advice that ρ has given in the past ($t' < t$) on goal G.

2.4. Trust Algorithm

In this section, we present one approach for calculating the trustworthiness of advice. To calculate the trustworthiness of ρ's advice $[P_\eta]G$, the relevant probability distributions on compliance, honour, and goal realisation should first be updated in order to update the probability distribution describing the outcome of ρ's advice: $\mathbb{P}^{t_{now}}(X|[P^\rho_\eta]G)$, where t_{now} describes current time. Every time the trustworthiness of the advice needs to be calculated, the relevant probability distributions are updated by calling Algorithm 1. After updating the necessary distributions and calculating $\mathbb{P}^{t_{now}}(X|[P^\rho_\eta]G)$, the final trust measure on ρ's advice ($trust^{t_{now}}([P^\rho_\eta]G)$) is calculated following Equation 16.

In our proposed approach, trust is calculated on demand. Other implementations that call Algorithm 1 to precompute probability distributions are possible. For example, Algorithm 1 may be executed periodically for all possible advice to ensure that when the trustworthiness of a specific advice is requested, minimal time and effort is spent on updating the relevant probability distributions.

To help update a probability distribution $\mathbb{P}(X|[P^\rho_\eta]G)$, originally specified as $\mathbb{P}(Observe(\gamma,[P_\eta]X) \mid Commit(\rho, [P_\eta]G))$, Algorithm 1 updates the relevant probability distributions on compliance, honour, and goal realisation. In Algorithm 1, the similarity between advisers follows Equation 13, the similarity between plans or goals follows Equation 2, and plan empowerment follows Equation 5. Updating the probability of a single expectation follows Equation 6, updating a probability distribution follows Equation 7, the decay follows Equation 9, and the relevance of an experience follows Equations 12, 14, and 15. It also assumes that both the initial distributions as well as the decay limit distributions are equiprobable distributions (i.e. they are equivalent to the uniform distribution $\mathbb{F}$).

The implementation of Algorithm 1 follows an incremental approach were we use 'memoization' techniques, as in dynamic programming, to improve the efficiency of the algorithm. We do this by saving the latest probability distributions and updating older computations. In this way, when a probability distribution needs to be calculated and it has already been calculated in the past, the distribution is modified considering new experiences only.

The algorithm uses parameters ξ_C, ξ_H, and ξ_R to specify the thresholds for considering an experience relevant. By fixing the values of these parameters to high values, we can improve the efficiency of the algorithm even further by saying that experiences that would result in 'small' modifications to past probability distributions are not to be considered. By reducing the values of these parameters progressively, we can have more realistic and fine grained implementations.

Algorithm 1 $update([P^{\rho}_{\eta}]G)$

```
Require: $H$, which specifies the history (or set) of past experiences.
Require: $\mathcal{P}$ and $\mathcal{G}$, which represent the sets of all plans and goals, respectively.
Require: $\mathcal{D}$, which specifies the set of probability distributions that have been updated. Initially, we have $\mathcal{D} = \emptyset$.
Require: $\mathbb{P}^{t_I}_{\mathrm{C}} = \mathbb{P}^{t_I}_{\mathrm{H}} = \mathbb{P}^{t_I}_{\mathrm{R}} = \mathbb{F}$, which describe the initial value of the probability distributions on compliance, honour, and goal realisation, respectively, and sets them all to the uniform distribution $\mathbb{F}$.
Require: $\mathbb{D}_{\mathrm{C}} = \mathbb{D}_{\mathrm{H}} = \mathbb{D}_{\mathrm{R}} = \mathbb{F}$, which describe the decay limit distributions for the probability distributions on compliance, honour, and goal realisation, respectively, and sets them all to the uniform distribution $\mathbb{F}$.
Require: $\xi_{\mathrm{C}}, \xi_{\mathrm{H}}, \xi_{\mathrm{R}} \in [0,1]$, which describe the thresholds for considering an experience relevant when updating the probabilities on compliance, honour, and goal realisation, respectively.
Require: $S_A : \mathcal{R} \times \mathcal{R} \to [0,1]$, which describes the similarity between two advisers, where $\mathcal{R}$ is the set of all advisers. (See Equation 13)
Require: $Sim : \mathcal{T} \times \mathcal{T} \to [0,1]$, which describes the similarity between two plans or goals, respectively, where $\mathcal{T} = \mathcal{P} \vee \mathcal{G}$ is either the set of all plans $\mathcal{P}$ or the set of all goals $\mathcal{G}$. (See Equation 2)
Require: $Emp : \mathcal{P} \times \mathcal{P} \to [0,1]$, which describes how much does one plan empower another. (See Equation 5)
Require: $\zeta_g, \zeta_p \in [0,1]$, which describe the weights for goal and plan similarity when calculating the relevance of an experience with respect to the probability on compliance.
Require: $\zeta_r \in [0,1]$, which describes the threshold for considering two advisers similar.
Require: $\epsilon_{\mathrm{C}}, \epsilon_{\mathrm{H}}, \epsilon_{\mathrm{R}} \in [0,1]$, which describe the percentages by which the probabilities on compliance, honour, and goal realisation may increase.
Require: $\nu \in [0,1]$, which describes the decay rate.
Require: $\Delta_{t,t'} : T \times T \to [0,1]$, which describes the pace of decay, where $T$ represents time. (See Section 2.1.4)
Require: $t_{now} \in T$, where $t_{now}$ represents current time.
Require: $\ominus$ and $\oplus$, which describe functions that remove elements from a set and append elements to a set, respectively. For example, $X \ominus x$ removes the element $x$ from the set $X$, whereas $X \oplus x$ adds the element $x$ to the set $X$.

    ▹ First, the probability distribution on compliance is updated.
    ▹ Get the experiences that have not been taken into account yet.
    if $\mathbb{P}^{t_{last}}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G) \in \mathcal{D}$ then
        $H^* = \{\mu \mid \mu \in H$ and $\mu = \langle \_, Commit(x,y)_{t_\mu}, \_, \_ \rangle$ and $t_\mu > t_{last}\}$
    else
        $H^* = H$
        $\mathbb{P}^{t_{last}}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G) = \mathbb{P}^{t_I}_{\mathrm{C}}$
    end if
for all $\mu \in H^*$ and $\mu = \langle Commit(\rho', [P'_{\eta}]G')_t, Commit(\eta, P'')_{t_\mu}, \_, \_ \rangle$ and $S_A(\rho, \rho') > \zeta_r$ do
    ▹ The relevance of $\mu$ with respect to compliance is calculated.
    $R_\mu(\mathbb{P}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G)) = \frac{\zeta_g \cdot Sim(G', G) + \zeta_p \cdot Sim(P', P)}{\zeta_g + \zeta_p}$
    if $R_\mu(\mathbb{P}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G)) > \xi_{\mathrm{C}}$ then
        ▹ $\mathbb{P}^{t_{last}}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G)$ is decayed to $t_\mu$
        $\mathbb{P}^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G) = \nu^{\Delta_{t_\mu, t_{last}}} \cdot \mathbb{P}^{t_{last}}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G) + (1 - \nu^{\Delta_{t_\mu, t_{last}}})\mathbb{D}_{\mathrm{C}}$
        for all $x \in \{y \mid |Sim(P, y) - Sim(P', P'')| \leq \chi_{\mathrm{C}}\}$ do
            ▹ The probability of the expectation $x$ is calculated.
            $p^{t_\mu}_{\mathrm{C}}(X = x|[P^{\rho}_{\eta}]G) = p^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{C}}(X = x|[P^{\rho}_{\eta}]G) +$
                $(1 - p^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{C}}(X = x|[P^{\rho}_{\eta}]G)) \cdot \epsilon_{\mathrm{C}} \cdot R_\mu(\mathbb{P}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G))$
        end for
        ▹ The probability distribution is updated
        $\mathbb{P}^{t_\mu}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G) = \arg\min_{\mathbb{P}(X)} \sum_i p^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{C}}(X = i|[P^{\rho}_{\eta}]G) \log \frac{p^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{C}}(X = i|[P^{\rho}_{\eta}]G)}{p(X = i)}$
                        such that   $\{p(X = x) = p^{t_\mu}_{\mathrm{C}}(X = x|[P^{\rho}_{\eta}]G), ...\}$
        ▹ $\mathcal{D}$ is updated to contain the latest distribution $\mathbb{P}^{t_\mu}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G)$.
        $\mathcal{D} = \mathcal{D} \ominus \mathbb{P}^{t_{last}}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G) \oplus \mathbb{P}^{t_\mu}_{\mathrm{C}}(X|[P^{\rho}_{\eta}]G)$
    end if
end for

▹ Second, the relevant probability distributions on honour are updated in a similar manner to that on compliance.
for all $P' \in \mathcal{P}$ do
    if $\mathbb{P}^{t_{last}}_{\mathrm{H}}(X|P'_{\eta}) \in \mathcal{D}$ then
        $H^* = \{\mu \mid \mu \in H$ and $\mu = \langle \_, \_, Observe(x,y)_{t_\mu}, \_ \rangle$ and $t_\mu > t_{last}\}$
    else
        $H^* = H$
        $\mathbb{P}^{t_{last}}_{\mathrm{H}}(X|P'_{\eta}) = \mathbb{P}^{t_I}_{\mathrm{H}}$
    end if
    for all $\mu \in H^*$ and $\mu = \langle \_, Commit(\eta, P'')_t, Observe(\alpha, P'''_{\eta})_{t_\mu}, \_ \rangle$ do
        $R_\mu(\mathbb{P}_{\mathrm{H}}(X|P'_{\eta})) = Emp(P', P'')$
        if $R_\mu(\mathbb{P}_{\mathrm{H}}(X|P'_{\eta})) > \xi_{\mathrm{H}}$ then
            $\mathbb{P}^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{H}}(X|P'_{\eta}) = \nu^{\Delta_{t_\mu, t_{last}}} \cdot \mathbb{P}^{t_{last}}_{\mathrm{H}}(X|P'_{\eta}) + (1 - \nu^{\Delta_{t_\mu, t_{last}}})\mathbb{D}_{\mathrm{H}}$
            for all $x \in \{y \mid |Sim(P', y) - Sim(P'', P''')| \leq \chi_{\mathrm{H}}\}$ do
                $p^{t_\mu}_{\mathrm{H}}(X = x|P'_{\eta}) = p^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{H}}(X = x|P'_{\eta}) +$
                    $(1 - p^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{H}}(X = x|P'_{\eta})) \cdot \epsilon_{\mathrm{H}} \cdot R_\mu(\mathbb{P}_{\mathrm{H}}(X|P'_{\eta}))$
            end for
            $\mathbb{P}^{t_\mu}_{\mathrm{H}}(X|P'_{\eta}) = \arg\min_{\mathbb{P}(X)} \sum_i p^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{H}}(X = i|P'_{\eta}) \log \frac{p^{t_{last} \rightsquigarrow t_\mu}_{\mathrm{H}}(X = i|P'_{\eta})}{p(X = i)}$
                            such that   $\{p(X = x) = p^{t_\mu}_{\mathrm{H}}(X = x|P'_{\eta}), ...\}$
            $\mathcal{D} = \mathcal{D} \ominus \mathbb{P}^{t_{last}}_{\mathrm{H}}(X|P'_{\eta}) \oplus \mathbb{P}^{t_\mu}_{\mathrm{H}}(X|P'_{\eta})$
        end if
    end for
end for
```

```
▹ Third, the relevant probability distributions on goal realisation are updated in a similar manner to those on compliance and honour.
for all $P' \in \mathcal{P}$ do
    if $\mathbb{P}_{\mathrm{R}}^{t_{last}}(X|P') \in \mathcal{D}$ then
        $H^* = \{\mu \mid \mu \in H$ and $\mu = \langle \_, \_, \_, Observe(x, y)_{t_\mu} \rangle$ and $t_\mu > t_{last}\}$
    else
        $H^* = H$
        $\mathbb{P}_{\mathrm{R}}^{t_{last}}(X|P') = \mathbb{P}_{\mathrm{R}}^{t_I}$
    end if
    for all $\mu \in H^*$ and $\mu = \langle \_, \_, Observe(\alpha, P''_{\eta'})_t, Observe(\beta, G')_{t_\mu} \rangle$ do
        $R_\mu(\mathbb{P}_{\mathrm{R}}(X|P')) = Sim(P', P'')$
        if $R_\mu(\mathbb{P}_{\mathrm{R}}(X|P')) > \xi_{\mathrm{R}}$ then
            $\mathbb{P}_{\mathrm{R}}^{t_{last} \rightsquigarrow t_\mu}(X|P') = \nu^{\Delta_{t_\mu, t_{last}}} \cdot \mathbb{P}_{\mathrm{R}}^{t_{last}}(X|P') + (1 - \nu^{\Delta_{t_\mu, t_{last}}})\mathbb{D}_{\mathrm{R}}$
            $x = G'$
            $p_{\mathrm{R}}^{t_\mu}(X = x|P') = p_{\mathrm{R}}^{t_{last} \rightsquigarrow t_\mu}(X = x|P') +$
                $(1 - p_{\mathrm{R}}^{t_{last} \rightsquigarrow t_\mu}(X = x|P')) \cdot \epsilon_{\mathrm{R}} \cdot R_\mu(\mathbb{P}_{\mathrm{R}}(X|P'))$
            $\mathbb{P}_{\mathrm{R}}^{t_\mu}(X|P') = \arg\min_{\mathbb{P}(X)} \sum_i p_{\mathrm{R}}^{t_{last} \rightsquigarrow t_\mu}(X = i|P') \log \frac{p_{\mathrm{R}}^{t_{last} \rightsquigarrow t_\mu}(X = i|P')}{p(X = i)}$
                                        such that    $p(X = x) = p_{\mathrm{R}}^{t_\mu}(X = x|P')$
            $\mathcal{D} = \mathcal{D} \ominus \mathbb{P}_{\mathrm{R}}^{t_{last}}(X|P') \oplus \mathbb{P}_{\mathrm{R}}^{t_\mu}(X|P')$
        end if
    end for
end for

▹ Finally, update the probability distribution $\mathbb{P}^{t_{now}}(X|[P_\eta^\rho]G)$.
for all $x \in \mathcal{G}$ do
    $p^{t_{now}}(X = x|[P_\eta^\rho]G) = 0$
    for all $y \in \mathcal{P}$ do
        for all $z \in \mathcal{P}$ do
            if $\mathbb{P}_{\mathrm{H}}^{t_{last}}(Y|z_\eta) \in \mathcal{D}$ and $\mathbb{P}_{\mathrm{R}}^{t_{last}}(X|y) \in \mathcal{D}$ then
                $p^{t_{now}}(X = x|[P_\eta^\rho]G) = p^{t_{now}}(X = x|[P_\eta^\rho]G) +$
                $(p_{\mathrm{C}}^{t_{last} \rightsquigarrow t_{now}}(Z{=}z|[P_\eta^\rho]G) \cdot p_{\mathrm{H}}^{t_{last} \rightsquigarrow t_{now}}(Y{=}y|z_\eta) \cdot p_{\mathrm{R}}^{t_{last} \rightsquigarrow t_{now}}(X{=}x|y))$
            end if
        end for
    end for
end for
```

3. Evaluation

In this Section we provide an empirical evaluation based on a simulation of the interaction between advisers and advisees. We refer to advisers and advisees as recommenders and users, respectively. In the following, we describe our experimental platform, define benchmarks for a music learning domain, and show a comparison between our CONSUASOR algorithm, the well-known EigenTrust algorithm (Kamvar et al., 2003), and a random method. We note that, to our knowledge, CONSUASOR is the only model that takes into consideration the particularities of an advice — plans and goals — instead of considering advice as a single unit — a black box. As such, it is really difficult to compare it to an existing trust model. Nevertheless, for the sake of a more meaningful evaluation, we compare CONSUASOR with EigenTrust, as opposed to other existing and more recent models, as EigenTrust is the most renowned model in terms of its application to real life scenarios (for instance, it has been used in eBay and Amazon).

3.1. Experimental platform

The following sets and functions determine a benchmark:

- A set of actions $\mathcal{A}$, from which plans are composed, and an action meronomy $\mathcal{M}_{\mathcal{A}}$ used to define particular similarity functions (based on the semantic similarity measure of Li et al. (2003)) and empowerment functions (based on the OpinioNet algorithm of Osman et al. (2010)).
- A set of propositional terms $\mathcal{G}$ to define goals and a term ontology $T_{\mathcal{G}}$ used to define particular similarity functions (based on the semantic similarity measure of Li et al. (2003)).
- The set of all plans $\mathcal{P}$ and the set of all goals $\mathcal{G}$.
- A causality function $f : \mathcal{P} \to \mathcal{G}$, which describes whether a plan achieves a goal.
- A set of users $\mathcal{U}$ where every $\eta \in \mathcal{U}$ is defined by the tuple $\langle G^t, d_1, d_2, c_\eta, h_\eta \rangle$ such that $d_1, d_2 \in [0, 1]$ and:
    - $G^t \subseteq \mathcal{G}$ is the set of η's goals at every time instant t.
    - $c_\eta : \mathcal{P} \to \mathcal{P}$ and $h_\eta : \mathcal{P} \to \mathcal{P}$ are functions describing η's compliance and honor, therefore: when a plan P is recommended to η, η commits to $c_\eta(P)$; and when η commits to P, η executes $h_\eta(P)$.
    - d_1 and d_2 describe the level of compliance and honor of η satisfying the following conditions:

$$\begin{aligned} \forall P \in \mathcal{P}: \quad & Sim(c_\eta(P), P) \geq 1 - d_1 \\ & Sim(h_\eta(P), P) \geq 1 - d_2 \end{aligned}$$

Notice that when $d_1 = d_2 = 0$ a user is fully compliant and honourable. When $d_1 > 0$ or $d_2 > 0$, then the user may not commit to the recommended plan or execute its commitment, respectively.

- A set of recommenders $\mathcal{R}$ where every $\rho \in \mathcal{R}$ is defined by the tuple $\langle d_3, d_4, d_5, \{{c_\eta}^{-1}\}_{\eta \in \mathcal{U}}, \{{h_\eta}^{-1}\}_{\eta \in \mathcal{U}}, f^{-1}\}\rangle$ such that $d_3, d_4, d_5 \in [0, 1]$ and:
  - ${c_\eta}^{-1} : \mathcal{P} \to \mathcal{P}$ describes ρ's knowledge about η's compliance, and ρ believes that for η to commit to plan P, ${c_\eta}^{-1}(P)$ should be recommended to η.
  - ${h_\eta}^{-1} : \mathcal{P} \to \mathcal{P}$ describes ρ's knowledge about η's honor, and ρ believes that for η to execute plan P, η should have commited to ${h_\eta}^{-1}(P)$.
  - $f^{-1} : \mathcal{P} \to \mathcal{G}$ describes ρ's knowledge about the causality between a plan and a goal (that is, if the plan achieves the goal), and ρ believes that for goal G to be achieved, plan $f^{-1}(G)$ must be executed.
  - d_3, d_4 and d_5 describe the level of knowledge of ρ about compliance, honor and causality, respectively, satisfying the following conditions:

$$\begin{aligned} \forall P \in \mathcal{P}: \quad & Sim({c_\eta}^{-1}(P), P') \geq 1 - d_3 \text{ and } c_\eta(P') = P \\ & Sim({h_\eta}^{-1}(P), P') \geq 1 - d_4 \text{ and } h_\eta(P') = P \\ & Sim(f(G)^{-1}, P) \geq 1 - d_5 \text{ and } f(G) = P \end{aligned}$$

Notice that when $d_3 = 0$, ρ is accurate about what plan P' must be recommended to η so that η commits to P. When $d_3 > 0$, ρ's knowledge is not accurate and ${c_\eta}^{-1}(P)$ may not lead to η committing to P. The same reasoning follows for d_4, d_5 on honor and causality. A competent recommender is one with $d_3 = d_4 = d_5 = 0$.

We assume determinism and that Assumptions 1 and 2 are satisfied. A recommender ρ suggests a plan P for user η for goal G as follows:

$$P = {c_\eta}^{-1}({h_\eta}^{-1}(f^{-1}(G)))$$

If the recommended plan P is accepted, a new experience μ is generated and added to the history of experiences H. The new generated experience would then be:

$$\mu = \langle [P_\eta]G_{t'}, c_\eta(P)_{t''}, h_\eta(c_\eta(P))_{t'''}, f(h_\eta(c_\eta(P))_{t''''})\rangle$$

where time instants $t' < t'' < t''' < t''''$ are generated with a difference of one (simulation) time-step between each time instant and the following one.

The success of a recommendation $[P]G$ for user η is defined as:

$$Succ_\eta(P, G) = Sim(G, f(h_\eta(c_\eta(P))))$$

And the success of a user η at time t is defined as:

$$Succ^t_\eta = \sum_{\mu = \langle [P_\eta]G_{t'}, \_{t''}, \_{t'''}, \_{t''''}\rangle_{t' < t'' < t''' < t'''' < t} \in H} \frac{Succ_\eta(P,G)}{|\mu|}$$

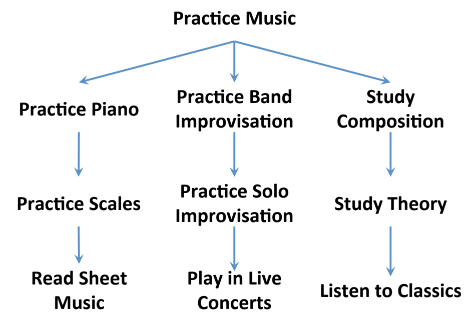
where $[P_\eta]G$ is a recommendation selected by user η at time $t' < t$. A trust strategy is used to decide which recommendation to select from the set of all recommendations suggested by all recommenders. A trust strategy will be considered good for user η if $Succ_\eta(\cdot, \cdot)$ increases over time. That is, if a time frame of size n is considered, then $(Succ^t_\eta - Succ^{t-n}_\eta)/n$ should approach the maximum success possible, where the maximum success is determined by the user's compliance and honour (the less compliant and honourable a user is, the lower the maximum possible success is).

3.2. Benchmarks

In our benchmarks we consider an action meronomy $\mathcal{M}_\mathcal{A}$ and goal ontology $T_\mathcal{G}$ of 10 terms related with music composition, improvisation and instrument performance (Figure 3).

Our set of plans $\mathcal{P}$ contains 10 plans, one for every action in $\mathcal{M}_\mathcal{A}$. Causality function f is built such that every plan in $\mathcal{P}$ achives a goal in $\mathcal{G}$: f(Practice all) = Finest Musician, f(Practice Piano) = Good Piano Performant, f(Practice Band Improvisation) = Good Band Improviser, etc.

In the experiments there is always a single user $\eta \in U$. Compliance and honor functions for this user, $c_\eta(P)$ and $h_\eta(P)$, are built such that for every plan $P \in \mathcal{P}$, a plan P' is picked at a distance d_1 (for compliance) or d_2 (for honor). Notice that c_η



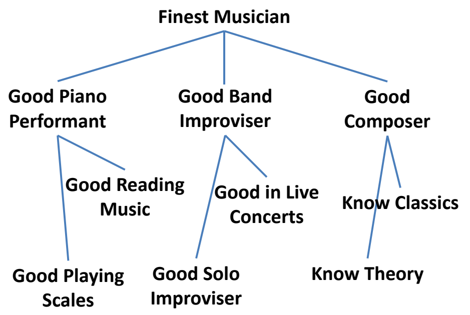


Figure 3: $\mathcal{M}_{\mathcal{A}}$, (up) and $T_{\mathcal{G}}$, (down)

and h_η are not inyective functions, therefore when $d_1 \neq 0$ and $d_2 \neq 0$ it may be the case that η never commits to a plan P ($\nexists P'$ s.t. $c_\eta(P') = P$) or honors a plan P ($\nexists P'$ s.t. $h_\eta(P') = P$).

The number of recommenders varies from 5 to 30. To generate ρ's knowledge about η's compliance ($c_\eta(P)^{-1}$), we first calculate the inverse of $c_\eta(P)$ and then pick a plan in $\mathcal{P}$ at a distance d_3 from the inverse value. The same logic is followed to generate h_η^{-1} and f^{-1}.

Experiments run as follows. The user needs advice to achieve goals and he receives advice from the recommenders. Once an advice is selected, an experience is generated where the user may or may not commit to the recommended plan and achieve the desired goal. As explained in Section 3.1, recommendations are generated according to the recommender's knowledge and experiences are generated according to the user's compliance and honor. In the experiments we perform several iterations and observe the trust evolution of the recommenders and the user's success rate over time. Results are given for different values of d_1, d_2, d_3, d_4, d_5. For convenience we use the notation $\bar{d} = (d_1 + d_2 + d_3)/3$.

We compare 3 trust strategies: selecting advise randomly, selecting the advice whose adviser is ranked top by the EigenTrust algorithm (Kamvar et al., 2003),[17] and selecting the advice/recommender ranked top by CONSUASOR.

The flexibility of the CONSUASOR model allows to consider several applications where we are able to focus on different aspects of advice. We provide different experimental scenarios. In Experiment 1, we calculate the trust on advices given by recommenders, while in Experiment 2 we calculate the trust on recommenders for a specific goal. Experiment 1 addresses the question: How good is ρ recommending $[P_\eta]G$? While Experiment 2 addresses the question: How good is recommender ρ giving advice for goal G? We are interested in showing the flexibility and coherence of the model in cases were different approaches are needed. Finally, Experiment 3 presents a comparison of different CONSUASOR trust evaluations focusing on different aspects of advice. In every case, we present averaged results of 50 experiments with a time frame $n = 5$.

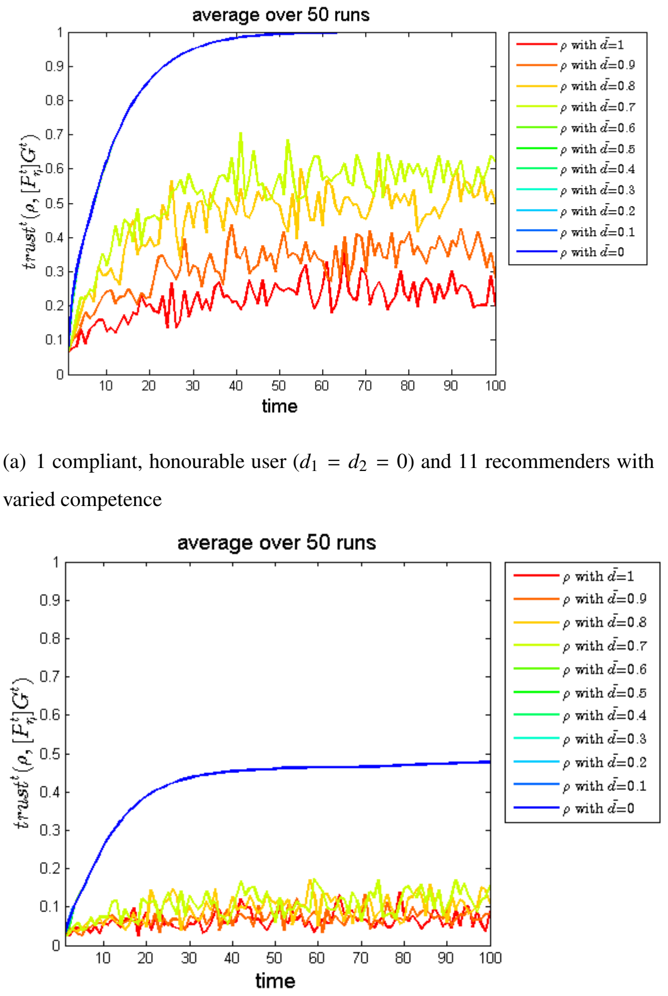
3.3. Experiment 1. Trust on Recommender giving Advice

In these experiments the user η has a single goal (randomly picked from the goals ontology) which is fixed over time. There is a set of recommenders providing advice. Every experiment performs 100 iterations (time instants) of the following steps: (1) The user asks advice to all recommenders for its goal; (2) Each recommender provides a recommendation, as defined in Section 3.1; (3) The user selects one advice (in the random case, randomly, in the EigenTrust and CONSUASOR cases, the most trusted one); and, (4) An experience is generated and stored, the trust of recommenders is updated and the success of the user is updated.

Figure 4 shows the CONSUASOR trust values evolution in time for 11 recommenders (each line represents the trust in a specific recommender). Recommenders knowledge vary from fully knowledgeable (small $\bar{d}$) to ignorant (large $\bar{d}$). Case (a)

[17] In the EigenTrust case, we compute the normalized local trust. The notion of transitive trust is not needed since neither users nor recommenders provide advise among themselves.

present results for a fully compliant and honourable user ($d_1 = d_2 = 0$). In this case, recommenders with $0 \leq \bar{d} \leq 0.6$ overlap. This is an effect of our small ontology: since we have just a few terms, the semantic distance between them is relatively large. We need to increase the distance beyond 0.6 to obtain recommenders that are not competent. The trust on competent recommenders ($0 \leq \bar{d} \leq 0.6$) increases over time and reaches 1, while the trust on less competent recommenders ($0.7 \leq \bar{d} \leq 1$) fluctuates up and down (an incompetent recommender may randomly give a good recommendation, but most of the times he does not). As expected, the worst the recommender, the lower its trust line is. Case(b) present results with a non-compliant and dis-honourable user ($d_1 = d_2 = 1$). We observe the same behavior of Case (a) but trust measures are lower now (trust on competent recommenders reaches 0.5 approximately instead of 1). This is because a non-compliant and dis-honourable user does not necessarily commit to or honor every plan (because c_η and h_η are not inyective functions). In such cases, it is not possible to obtain a plan such that the user is willing to commit to (or execute) and that causes the goal fullfilment, given the recommender's knowledge ($\nexists P''$ s.t $c_\eta(P'') = P'$ and $h_\eta(P') = P$ and $f(P) = G$). In other words, there is no good plan for this non-compliant and dis-honourable user. In such cases, the recommender does not provide any advice to the user and the trust on that recommender/advice is not measured (since the advice does not exist), then we simply put a zero as the trust value in that iteration and when executions are averaged this is reflected in the graphics. If we increase d_1 and d_2 (η's compliance and honor), we see how the trust lines increase until we obtain a graph like (a) for a fully compliant and honourable user.

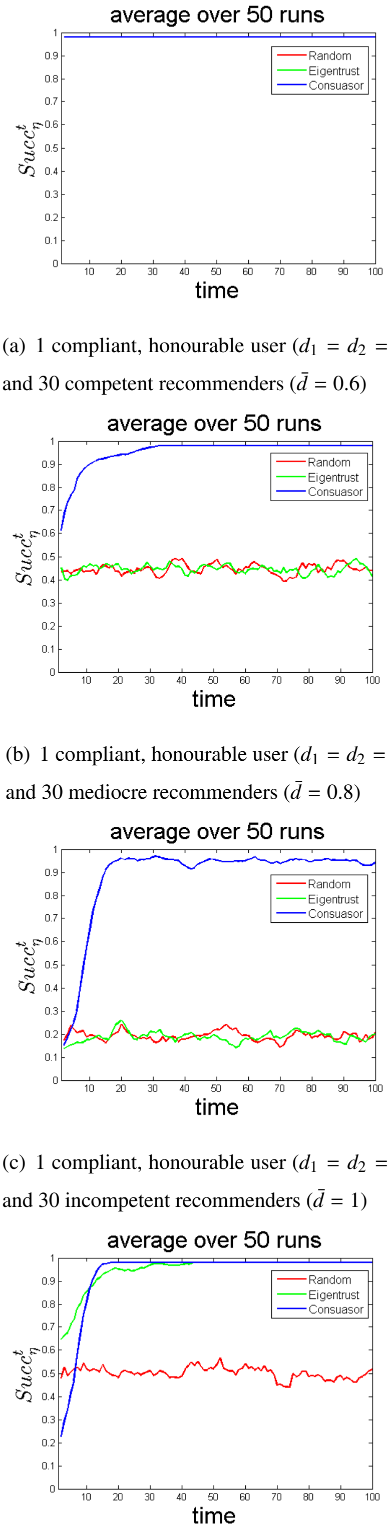


(a) 1 compliant, honourable user ($d_1 = d_2 = 0$) and 11 recommenders with varied competence

(b) 1 non-compliant, dis-honourable user ($d_1 = d_2 = 1$) and 11 recommenders with varied competence

Figure 4: Experiment1. Trust evaluation in time

Figures 5 and 6 show η's success evolution in time ($Succ^t_\eta$) for compliant/honourable and non-compliant/dis-honourable users, respectively, following the three strategies: random, EigenTrust and CONSUASOR. Results in Figure 5 correspond to executions with a compliant and honourable user. Case 5(a) includes 30 competent recommenders. Competent recommenders always give good advice and η always commits and executes that advice, therefore success is always 1. Cases 5(b) and 5(c) include 30 mediocre and incompetent recommenders, respectively, we observe how the CONSUASOR algorithm obtains high levels of success while EigenTrust and the random approach remain low. Case 5(d) includes 5 recommenders with varying competency (from fully competent to fully incompetent), we observe that EigenTrust performs well and that CONSUASOR stabilizes earlier. In Figure 6, the user η is not always compliant and honourable ($d_1 = d_2 = 0.7$). We observe the same behaviour of the previous cases but the success rate is lower now, since sometimes there are no recommendations that allow η's goal to be fulfilled and that η is willing to commit to or execute. For higher d_1 and d_2 values (less compliant/honourable user), we verified experimentally that the suc-



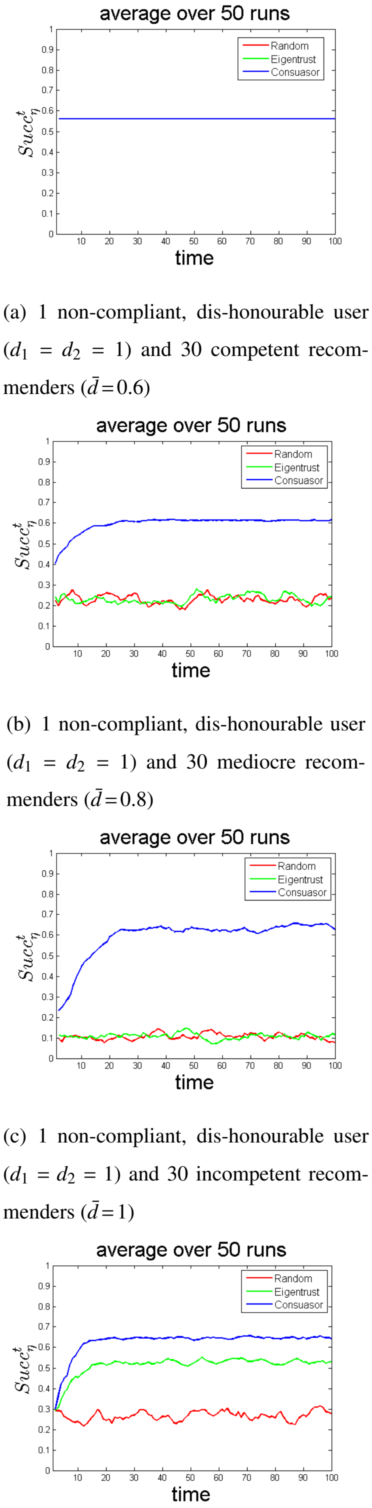
(a) 1 compliant, honourable user ($d_1 = d_2 = 0$) and 30 competent recommenders ($\bar{d} = 0.6$)

(b) 1 compliant, honourable user ($d_1 = d_2 = 0$) and 30 mediocre recommenders ($\bar{d} = 0.8$)

(c) 1 compliant, honourable user ($d_1 = d_2 = 0$) and 30 incompetent recommenders ($\bar{d} = 1$)

(d) 1 compliant, honourable user ($d_1 = d_2 = 0$) and 5 recommenders with varied competence ($\bar{d} = 1 \ldots 0.6$)

Figure 5: Experiment1. Success evaluation in time, for compliant and honourable users



(a) 1 non-compliant, dis-honourable user ($d_1 = d_2 = 1$) and 30 competent recommenders ($\bar{d}=0.6$)

(b) 1 non-compliant, dis-honourable user ($d_1 = d_2 = 1$) and 30 mediocre recommenders ($\bar{d}=0.8$)

(c) 1 non-compliant, dis-honourable user ($d_1 = d_2 = 1$) and 30 incompetent recommenders ($\bar{d}=1$)

(d) 1 non-compliant, dis-honourable user ($d_1 = d_2 = 1$) and 5 recommenders with varied competence ($\bar{d}=1 \ldots 0.6$)

Figure 6: Experiment1. Success evaluation in time, for non-compliant and dishonourable users

cess lines are lower and for lower d_1 and d_2 values the success lines increase.

These results show that CONSUASOR is able to increase η's success rate in a short period of time, when compared with the random and EigenTrust strategies. EigenTrust obtains high levels of success when there is at least one competent recommender, but when recommenders are less competent or users are not fully compliant or honourable, its success diminishes considerably (in fact it acts similarly to the random approach in many cases). CONSUASOR is able to learn which advice is trustworthy and reaches high levels of success in all observed cases.

3.4. Experiment 2. Trust on Recommenders for Goals

In these experiments the user η has 3 goals that will change over time every 40 time instants ($G^{1,\ldots,40}$ =Good Piano Performant, $G^{41,\ldots,80}$ = Good Band Improviser, $G^{81,\ldots,120}$= Good Composer). η is fully compliant and honourable ($d_1 = d_2 = 0$) and there are 11 recommenders providing advice, each spacialized in a different goal. Every experiment performs 100 iterations (time instants) of the following steps: (1) The user calculates the trust on each recommender for its goal G^t; (2) The user selects the most trusted recommender and asks him for advice; (3) The selected recommender provides a recommendation, as defined in Section 3.1; (4) An experience is generated and stored, the trust on the recommender is updated and the success of the user is updated.

Figures 7 and 8 show the recommenders's trust evolution and η's success evolution in time. In Figure 7, each recommender is fully expert in one goal and ignorant in the others ($\bar{d} = 0$ when he is asked for its goal of expertise and $\bar{d} = 1$ otherwise). We observe that every time the user changes its goal, the trust on the recommender which is expert in that goal grows. The trust evolution is slower when the user changes from one goal to the other, since more elements are included in the probability distributions and increments in such distributions require more time. The success evolution of CONSUASOR is better than the EigenTrust case. In Figure 8, each recommender is moderately

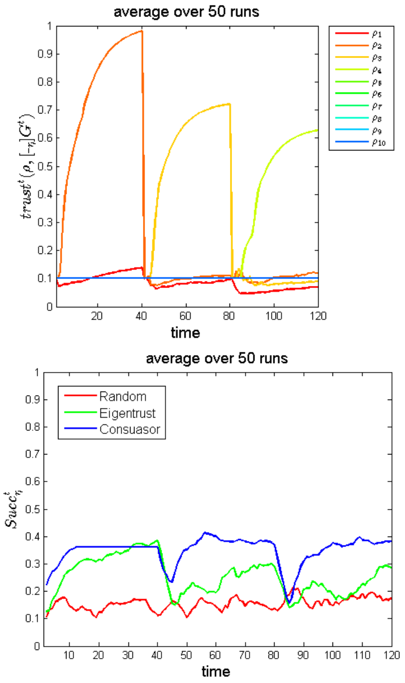


Figure 7: Experiment 3. Trust and success in time, for 1 compliant and honourable user ($d_1 = d_2 = 0$) with 3 different goals, and 11 recommenders, each competent in 1 goal ($\bar{d}=0$) and incompetent in the other 2 ($\bar{d}=1$)

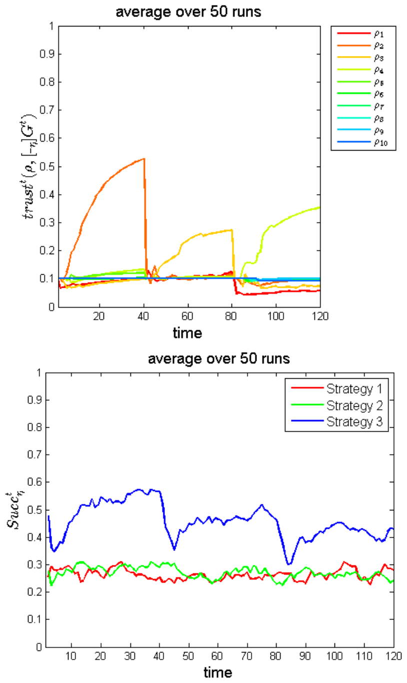


Figure 8: Experiment 3. Trust and success in time, for 1 compliant and honourable user ($d_1 = d_2 = 0$) with 3 different goals, and 11 recommenders, each with mediocre competency in 1 goal ($\bar{d} = 0.7$) and incompetent in the other 2 ($\bar{d}=1$)

knowledgeable in one goal and ignorant in the others ($\bar{d}$ = 0.7 when he is asked for its goal of expertise and $\bar{d}$ = 1 otherwise). As in the previous case, the recommender which is more knowledgeable at every time instant stands above the rest, although the trust measure is lower since the recommender is not fully expert and sometimes he might not give the best advice. Success evolution of the CONSUASOR strategy is also better than EigenTrust in this case.

These results show that CONSUASOR is able to adapt to a different scenario (calculate the most trusted recommender for a given goal) and learn which is the most suitable recommender when a user pursues different goals in time. Experiments where the user first asks for advice from the recommenders and then selects the most trusted advice were also executed, and these showed similar results trends. The adequacy of CONSUASOR to provide reliable trust measures at every time instant results in higher success rates than in the EigenTrust case.

3.5. Experiment 3. Comparison of CONSUASOR strategies

Finally we present different strategies that can be used with CONSUASOR and how these strategies behave in time for a particular context. In these experiments a compliant and honourable user η ($d_1 = d_2 = 0$) has 5 goals that change in time randomly. Recommenders are fully expert in one particular goal and ignorant in the others. In this context, η will focus on three specific questions: (1) How good is recommender ρ giving advice $[P_\eta]G$? (2) How good is recommender ρ giving advice for my goal G? and (3) How good is recommender ρ?

Strategy 1 consists of the following steps: (1) The user asks advice to all recommenders for its goal G^t; (2) Each recommender provides a recommendation; (3) The user calculates the trust on each advice and selects the most trusted one; and, (4) An experience is generated and stored, the trust on recommenders is updated and the success of the user is updated.

Strategy 2 consists of the following steps: (1) The user calculates the trust on each recommender for its goal G^t; (2) The user selects the most trusted recommender and asks him for advice; (3) The selected recommender provides a recommendation; (4) An experience is generated and stored, the trust on the recommender is updated and the success of the user is updated.

Strategy 3 consists of the following steps: (1) The user calculates the general trust on each recommender ρ; (2) The user selects the most trusted recommender and asks him for advice; (3) The selected recommender provides a recommendation; (4) An experience is generated and stored, the trust on recommenders is updated and the success of the user is updated.

Figure 9 shows the success evolution of strategies 1, 2 and 3. We observe that, in this particular context where recommenders are specialized in goals, Strategy 1 is the one that presents a faster learning rate in the first time instants, but eventually Strategy 2 achieves higher levels of success. In this particular context, focusing on a competent recommender for a given goal is more effective than focusing on the most trusted advice in general. Strategy 3 is not effective, which tells us that overgeneralising (focusing on competent recommenders in general) is not useful in this scenario.

These results stress the point that the flexibility of the CONSUASOR algorithm in calculating general trust measures is an advantage. However when making such generalizations, the decision on what to generalize should be carefully made, and it should take into consideration the specific context and the available information.

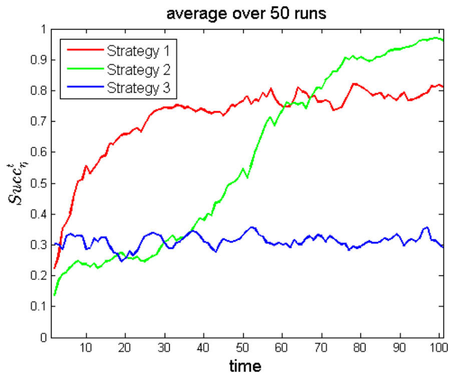


Figure 9: Experiment 3. Success in time, for 1 compliant and honourable user ($d_1 = d_2 = 0$) with 5 different goals, and 11 recommenders, each competent in 1 goal ($\bar{d} = 0$) and incompetent in the other 4 ($\bar{d} = 1$)

4. Background

The literature on trust is vast, with many surveys on the subject (e.g. Pinyol and Sabater-Mir (2013); Ramchurn et al. (2004a); Mui et al. (2002); Sabater and Sierra (2002)), where each survey focuses on different dimensions in their classification of existing trust and reputation models. For instance, Sabater and Sierra (2002), the most cited survey, categorises computational trust and reputation models based on six dimensions: (1) *paradaigm*, which describes whether the model follows a cognitive approach or a numerical one; (2) *information source*, which describes the type of information used, such as direct experiences, witness information, sociological information, and prejudice; (3) *visibility*, which describes whether a trust measure is a global property that may be viewed by all or a private and subjective property that each individual builds; (4) *granularity*, which describes whether a model is context dependent or not; (5) *cheating behaviour*, which describes what type of cheating behaviour is addressed, if any; and (6) *type of exchanged information*, which describes whether exchanged information is described as boolean or continous measures.

According to the above classification, CONSUASOR follows a numerical paradigm where trust is defined based on a probabilistic measure, although the model has been designed based on a cognitive model that assumes a good adviser is one who knows about the compliance and honour of advisees as well as the causal relations between plans and goals. Concerning the information source, we say CONSUASOR is based on direct experiences as the model relies on the experiences of a single entity (that which maintains the history of experiences). However, we note that the 'observers' that help populate the history of experiences may be trusted third-party entities. Concerning visibility, although a trust measure may be shared, we say the trust measure is subjective as it is local to the history of experiences it relies on. Concerning the granularity, CONSUASOR is context dependent, as it learns from past experiences in similar contexts. Concerning cheating behaviour, CONSUASOR does assume that the adviser or advisee might lie. For example, an advisee may say they will perform a plan (i.e. commit to it) when they are in fact not willing to do so. However, CONSUASOR assumes observers to be truthful. Who to trust as an observer is a decision left for the entity maintaining the history of experiences (as it has been discussed in Section 2.1.1).

From the large existing literature, the models that may be classified as similar to ours are those that are context dependent, which we focus on next. One of these models is the early model by Marsh (1994), which proposed an approach for calculating 'situational trust'. In this model, the trust that an agent x has on an agent y in a given situation (or context) α is based on the utility that x gains from situation α, the importance of the situation α for x, and an estimate of the general trust of x on y that takes into account all possible relevant data in similar past situations. Marsh (1994) also has a notion of decay, which is modelled as a time window for experiences.

A more cognitive-based model is that of Castelfranchi and Falcone (1998), where trust is viewed as a mental state and a complex attitude where the trust of x on y regarding goal g depends on whether x believes y is both capable and willing of performing g. x's trust on y regarding goal g also depends on whether g is a goal for x and whether x believes it depends on y in achieving goal g.

The model by Abdul-Rahman and Hailes (2000) makes use of both direct and indirect (or communicated) experiences. The model uses a qualitative degree approach to model trust, taking into account the context as well as the trustworthiness of indirect (communicated) experiences. However, the modelling of uncertainty is somewhat ad-hoc and not based on probabilistic grounds.

In Mui et al. (2001), a context-based personalised reputation measure is inferred from propagated ratings through a peer-to-peer network. The model is based on a Bayesian probabilistic framework.

The Regret model (Sabater and Sierra, 2001) also calculates trust based on learning from similar past experiences. Both direct and indirect experiences are considered, and an incorporated credibility model is used to assess the reputation and trust of an information provider, based on social network analysis and prejudices. However, the overall notion of trust does not have a probabilistic meaning and is based on a utility modelling of the interactions that we depart from (see Debenham and Sierra (2009) for a discussion).

Like Regret, the FIRE model (Huynh et al., 2006) uses a number of information sources to calculate and integrate four different types of trust and reputation measures: (1) interaction trust, which is based on learning from the similar past experience of direct interactions; (2) role-based trust, which is a trust measure that is influenced by role-based relationships between the agents; (3) witness reputation, which is based on learning from the similar past experience of indirect interactions; and (4) certified reputation, which is built from third-party references provided by the agent itself (this is similar to the recommendation letters used when applying for a job position).

Ramchurn et al. (2004b) also based reputation on similarity between new contexts and past ones. However, their approach uses the concept of fuzzy sets to compute one's confidence, based on the notion of assigning utilities to the different aspects of a context. Trust is then built on the concept of the maximum expected loss in utility.

Simari et al. (2008)'s approach compares what the agent has committed to to what was actually delivered (which we refer to as observed). Similar to us, trust is then context dependent and based on past performances, although our similarity measures are more general measures that are based on semantic matching and empowerment.

Tavakolifard et al. (2008) illustrates how trust information in one context can affect trust information in other contexts. The model uses case-based reasoning to assess trustworthiness in new situations by relying on past similar experiences.

In the domain of recommendations, Nakatsuji et al. (2010) illustrate how recommendations may be made across multiple interrelated domains such as music and movies. Their similarity is based on comparing users who share (and rate) similar items or who share social connections. Such a similarity measure can then help provide recommendation chains (sequences of transitively associated edges) to items in other domains.

Finally, as illustrated by the introduction of Section 2, Osman et al. (2014) uses the same philosophy as this paper. In Osman et al. (2014), one compares what has been committed to to what is actually delivered (or observed), and the model is based on probability and information theory (for updating the probabilities of expectations) and semantic similarity (for comparing contexts). The equations on semantic similarity and information decay have first been introduced in Osman et al. (2014), whereas the notion of updating a probability distribution following the minimum relative entropy approach has first been introduced in Sierra and Debenham (2005).

However, differently from all the above models, CONSUASOR takes into account the specifics of an advice. For computing trust on advice, the model does not treat advice as a single entity, but assesses each of its components: the recommended plan and the intended goal. Carefully analysing an advice's constituents is what makes this model stand out from the rest. As a result, we say an adviser is a good adviser if the adviser takes into consideration the compliance and honour of the advisee when making his advice, in addition to the causality relations between plans and goals. Additionally, CONSUASOR introduces the notion of empowerment for comparing the similarity of contexts when the *capability* of performing actions (or plans) is important. Given these significant differences between CONSUASOR and existing models, for our evaluation, we have chosen to compare CONSUASOR to the popular EigenTrust model, which has been used in the industry by eBay and Amazon, to name a few. EigenTrust (Kamvar et al., 2003) is based on the notion of having each peer building a local trust value for other peers based on direct past experiences. A transitive trust measure is then computed for other peers based on the idea that if a peer trusts another peer x, then it will also trust every other peer trusted by x.

5. Conclusions

A good adviser needs to certainly know the causal relationship between performing a plan and realising a goal. However, when suggesting a plan to someone, the adviser also needs to know how compliant and honourable that person is, in order to personalise his advice accordingly. An adviser may need to be more demanding to those who are less compliant or honourable, for instance, in order to guarantee that the target plan is actually performed. Currently existing trust algorithms do not take into account the specifics of plan recommendation. We have proposed a trust model for advice based on the compliance and honour of advisees, as well as the causality between plans and goals. All these models are learned from similar past experiences using probability and information theory. The similarity of experiences is based on semantic similarity and the notion of empowerment, which this paper introduces to help compute the similarity of actions (or plans) when the capability of performing these actions (or plans) is important. Finally, we have proposed a benchmark for the evaluation of trust algorithms on advisers that may be used in the future to compare with other trust algorithms modelling plan recommendations, if they become available. Empirical evaluation shows that our model assesses coherent trust measures and reach high levels of success in short periods of time. We leave for future work a more detailed analysis of the sensitivity of the algorithm's parameters and further experiments on complex scenarios.

Concerning impact and applicability, we note that the proposed trust model may be used in a number of existing platforms where *advice* may be given to help achieve a certain *goal*. For example, in wikiHow (`wikihow.com`), which is a web-based and wiki-based community, how-to guides are proposed by members, where *guides* may be understood as *plans* that help achieve the *goal* defined by the title of the how-to guide (e.g. in 'how to boil an egg', the goal is 'boil and egg'). In Stack Overflow (`stackoverflow.com`), which is a question and answer website focusing on computing programming, *answers* may be interpreted as *plans* that are recommended by their author to fulfil the *goal* presented by the *question*. In complaint resolution technologies (e.g. CogniCor, `cognicor.com`), suggested *resolutions* may be understood as *plans* for the *goal* of *solving a given complaint*. These are sample scenarios where CONSUASOR can help select the best *advise* (whether it was a wikihow plan, a stack overflow answer, or a suggested complaint resolution). In any scenario, a number of requirements need to be fulfilled for the CONSUASOR model to be applicable. These may be summarised as follows: (1) the system needs to keep track of who is suggesting what plan for what goal; (2) the system needs to keep track of the results of following an advice (what plan was committed to and executed, and which goal was fulfilled) which usually relies on feedback from the user attempting to follow the advice; and (3) the system needs to be capable of computing the similarity of plans/goals. We note that some simplifications may be applied to increase the applicability of the CONSUASOR model. For example, in some scenarios, one may assume that executed plans are those that are committed to, and in others, one may assume that committed to plans are those that are recommended. Simplifications depend on how much detail does the system expect from the user's feedback. For instance, can the user state his executed plan? To simplify keeping track of the outcome, one may also assume that only two outcomes are possible: the goal was fulfilled and the goal was not fulfilled. Additionally, if a numerical rating can be provided by the user as feedback on an advice's outcome, this rating may be used to indicate the 'similarity' between the intended goal and the achieved goal. Finally, computing the similarity of plans/goals will very much depend on the specific context. For example, tags in Stack Overflow may be used to represent plans (in the case of answers) and goals (in the case of questions). Then, either the semantic similarity between tags can be used to compute the similarity between plans/goals, or the intersection between tags may be used to indicate the similarity between plans/goals (this implies that Equation 2 may be redefined to consider the intersection between keywords, as opposed to considering semantic similarity). In the case of domain specific conflict resolution technologies, where there usually is a predefined set of plans and goals, similarity between plans/goals may be computed in advance using semantic similarity, manual assessments, or any alternative method that seems most applicable to the scenario in question.

Acknowledgments

This work is supported by the PRAISE project (funded by the European Commission under the FP7 STREP grant number 318770), and the Agreement Technologies project (funded by CONSOLIDER CSD 2007-0022, INGENIO 2010).

References

Abdul-Rahman, A., Hailes, S., 2000. Supporting trust in virtual communities. In: Proceedings of the 33rd Hawaii International Conference on System Sciences. HICSS '00. IEEE Computer Society, Washington, DC, USA, pp. 6007–6016.

Castelfranchi, C., Falcone, R., Jul 1998. Principles of trust for mas: cognitive anatomy, social importance, and quantification. In: Proceedings of the International Conference on Multi Agent Systems. pp. 72–79.

Debenham, J., Sierra, C., 2009. An agent supports constructivist and ecological rationality. In: Baeza-Yates, R., Lang, J., Mitra, S., Simon (Eds.), Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 02. WI-IAT '09. IEEE Computer Society, Washington, DC, USA, pp. 255–258.

Huynh, T., Jennings, N., Shadbolt, N., 2006. An integrated trust and reputation model for open multi-agent systems. Autonomous Agents and Multi-Agent Systems 13 (2), 119–154.

Kamvar, S. D., Schlosser, M. T., Garcia-Molina, H., 2003. The eigentrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th International Conference on World Wide Web. WWW '03. ACM, New York, NY, USA, pp. 640–651.

Li, Y., Bandar, Z. A., McLean, D., Jul. 2003. An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on Knowledge and Data Engineering 15 (4), 871–882.

Linden, G., Smith, B., York, J., Jan 2003. Amazon.com recommendations: item-to-item collaborative filtering. Internet Computing, IEEE 7 (1), 76–80.

Marsh, S., 1994. Formalising trust as a computational concept. Ph.D. thesis, Department of Mathematics and Computer Science, University of Stirling.

Mui, L., Mohtashemi, M., Ang, C., Szolovits, P., Halberstadt, A., 2001. Ratings in distributed systems: A bayesian approach. In: Workshop on Information Technologies and Systems.

Mui, L., Mohtashemi, M., Halberstadt, A., 2002. Notions of reputation in multi-agents systems: A review. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1. AAMAS '02. ACM, New York, NY, USA, pp. 280–287.

Nakatsuji, M., Fujiwara, Y., Tanaka, A., Uchiyama, T., Ishida, T., 2010. Recommendations over domain specific user graphs. In: Proceedings of the 19th European Conference on Artificial Intelligence. IOS Press, Amsterdam, The Netherlands, The Netherlands, pp. 607–612.

Osman, N., Sierra, C., McNeill, F., Pane, J., Debenham, J., 2014. Trust and matching algorithms for selecting suitable agents. ACM Transactions on Intelligent Systems and Technology 5.

Osman, N., Sierra, C., Sabater-Mir, J., 2010. Propagation of opinions in structural graphs. In: Proceedings of the 19th European Conf. on Artificial Intelligence. IOS Press, Amsterdam, The Netherlands, pp. 595–600.

Pinyol, I., Sabater-Mir, J., Jun. 2013. Computational trust and reputation models for open multi-agent systems: A review. Artif. Intell. Rev. 40 (1), 1–25.

Ågotnes, T., van der Hoek, W., Wooldridge, M., 2008. Quantified coalition logic. Synthese 165 (2), 269–294.

Ramchurn, S. D., Huynh, D., Jennings, N. R., Mar. 2004a. Trust in multi-agent systems. Knowledge Engineering Review 19 (1), 1–25.

Ramchurn, S. D., Jennings, N. R., Sierra, C., Godo, L., 2004b. Devising a trust model for multi-agent interactions using confidence and reputation. Applied Artificial Intelligence 18 (9-10), 833–852.

Rubner, Y., Tomasi, C., Guibas, L. J., 1998. A metric for distributions with applications to image databases. In: Proceedings of the Sixth International Conference on Computer Vision (ICCV 1998). ICCV '98. IEEE Computer Society, Washington, DC, USA, pp. 59–66.

Sabater, J., Sierra, C., 2001. Regret: Reputation in gregarious societies. In: Proceedings of the Fifth International Conference on Autonomous Agents. AGENTS '01. ACM, New York, NY, USA, pp. 194–195.

Sabater, J., Sierra, C., 2002. Reputation and social network analysis in multi-agent systems. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1. AAMAS '02. ACM, New York, NY, USA, pp. 475–482.

Sierra, C., Debenham, J., 2005. An information-based model for trust. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems. AAMAS '05. ACM, New York, NY, USA, pp. 497–504.

Sierra, C., Debenham, J., 2006. Trust and honour in information-based agency. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. AAMAS '06. ACM, New York, NY, USA, pp. 1225–1232.

Simari, G. I., Broecheler, M., Subrahmanian, V. S., Kraus, S., 2008. Promises kept, promises broken: An axiomatic and quantitative treatment of fulfillment. In: Brewka, G., Lang, J. (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference. AAAI Press, pp. 59–69.

Tavakolifard, M., Knapskog, S. J., Herrmann, P., 2008. Trust transferability among similar contexts. In: Proceedings of the 4th ACM symposium on QoS and security for wireless and mobile networks. Q2SWinet '08. ACM, New York, NY, USA, pp. 91–97.