D3.3
Trust and Reputation in Online Social Learning Communities

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core authors</td>
<td>Nardine Osman, Patricia Gutierrez, and Carles Sierra</td>
</tr>
<tr>
<td>Maintainers/Editors</td>
<td>Nardine Osman</td>
</tr>
<tr>
<td>PRAISE research group leaders</td>
<td>Mark d’Inverno, François Pachet, Carles Sierra, Luc Steels</td>
</tr>
<tr>
<td>PRAISE project leader</td>
<td>Carles Sierra</td>
</tr>
</tbody>
</table>

| Grant agreement no.               | 318770                                                    |
| Project acronym                   | PRAISE - Practice and peRformance Analysis Inspiring Social Educa-   |
|                                    | tion                                                      |
| Version                           | v1.0                                                      |
| Date                              | August 28, 2013                                           |
| Distribution                      | Confidential                                              |
Abstract

Peers in online social learning communities—whether humans, autonomous agents, or web services—can provide feedback when observing the activities performed by others. This feedback may vary in its form. In this paper, we divide feedback into three different types: (1) as opinions which assess the quality of observed activities from the point of view of the opinion holder, (2) as advice which is specified as plans of actions that are proposed by advisers intending to help peers improve their performance, and (3) as information which simply attempt to introduce new information. An example of an opinion would be a statement such as: I do not like how you play ‘My Funny Valentine’. An example of a plan of action would be a statement such as: To get better at playing ‘My Funny Valentine’ on time, you will have to practice it twice a week over a period of four months. An example of information would be a statement such as: Miles Davis played ‘My Funny Valentine’. This paper visits these three forms of feedback and proposes trust models that help assess each of them.

Keyword list: trust and reputation, experience, semantic similarity, information decay, opinions, plans, information
Executive Summary

Trust and reputation are major issues that arise in online communities. Online communities are usually open societies with participants that may not all share the same goals or standards of behaviour. As such, there is a need to assess the trustworthiness of others.

One of the main uses of the trust and reputation measures in open online communities is their help in ranking, and hence aiding in the selection process, of both content and other community members. The ranking is usually based on feedback. For instance, providing feedback on uploaded instructional material could help teachers (and students) choose which class material is more suitable for a given class, providing feedback on uploaded student performances could help in the evaluation of students and best students can then be recommended automatically for gigs and concerts, providing feedback on members could help evaluate who is best at teaching a given subject, or who is best at playing at tomorrow’s gig, and so on.

In this research, we introduce the TrustIt model, a model that assesses the trustworthiness of feedback and feedback providers. We argue that since feedback is the basis of computational trust and reputation models, then assessing the trustworthiness of feedback provides the basis for assessing the trustworthiness of other entities; hence the general name TrustIt.

We divide feedback into three categories: (1) opinions, (2) advice, specified as plans that aim at fulfilling given goals, and (3) general information. As such, the TrustIt model is also divided into three modules accordingly: (1) trust on opinion holders, (2) trust on advisers, and (3) trust on information providers.

The TrustIt model is based on a simple concept. We say one can learn from similar past experiences in order to predict the outcome of future experiences, and we use this predicted outcome to measure the trustworthiness of members and entities in general. For example, Emily could learn that since Mike has been providing her with good advice on Jazz improvisation in the past, then he can be considered trustworthy enough in the future in providing advice on both Jazz improvisation and similar issues. As such, the core of the TrustIt model is based on the following three technologies: similarity measures, probability theory, and information theory.

As for the novelty of our proposed work, concerning opinions, we say that instead of simply providing the average of opinions when assessing the trustworthiness of some entity, we propose aggregating opinions through a weighted arithmetic average, where the weights describe how much trusted is the opinion holder. The trust on a given opinion holder represents how much reliable are the opinions of the opinion holder with respect to a given context. We address this issue by calculating this trust measure in terms of how far were the past opinions of the same opinion holder from that of the group opinion.

Another novelty of the trust on opinion holders module lies in analysing the character of opinion holders (such as labelling them as decisive if they usually never change their opinions, or indecisive if they are always changing their opinions, or persuaders if they usually convince the group to move towards their opinion, and so on). We illustrate how such characters can be accounted for when assessing the trust in an opinion holder.
Concerning the trust on advisers, we believe this is a new concept in the field of trust and reputation. Usually either the trust or reputation of items is assessed, or the trust on (or reputation of) people in the context of performing a simple task is assessed. For example, one may assess the trust on sellers, buyers, teachers, doctors, etc. However, the trust on advisers module computes the trust on advisers (where an adviser can either be a person, or peer, or even an algorithm, say a recommender system) in the context of performing more complex tasks. The task of advisers is to recommend a given plan for a given person so that a given goal can be achieved. Hence, the trust on advisers should consider all these four orthogonal aspects: the adviser, the plan, the goal, and the person this plan is recommended to. To our knowledge, we believe we are one of the few to address the issue of trust on advisers.

Concerning the trust on information providers, we present our view of what is the current problem with the presentation of online information, which we believe does not address the most crucial issue of figuring out what information is truly valuable. Accordingly, we propose an approach for addressing this problem by assessing the reliability, relevance, redundancy, and significance of information.
## Contents

1 Introduction 1

2 Background 2
   2.1 Trust & Matching Algorithms for Selecting Suitable Agents 3
   2.2 OpinioNet: The propagation of Opinions in Meronomies 5

3 TrustItO: Trust on Opinion Holders 7
   3.1 Opinion Change over Time 9
   3.2 Opinion Change and the Trust Measure 10

4 TrustItA: Trust on Advisers 11
   4.1 Preliminaries 12
      4.1.1 Experiences 12
      4.1.2 Similarity Measures 13
      4.1.3 Decay of Information 17
   4.2 Probability of a Proposed Plan Fulfilling its Goal 18
      4.2.1 Probability of Committing to a Proposed Plan 20
      4.2.2 Probability of Enacting a Committed-To Plan 22
      4.2.3 Probability of Realising a Goal after Enacting a given Plan 23
   4.3 Trust Computation 24
      4.3.1 Trust Measures 24
      4.3.2 General Trust Measures 25
   4.4 Trust Algorithm 26

5 TrustItI: Trust on Information Providers 26
   5.1 Reliability Measure 30
   5.2 Relevance Measure 31
   5.3 Redundancy Measure 32
   5.4 Significance Measure 32
   5.5 Trust versus Reputation 33

6 PRAISE: A use case in music learning 34
   6.1 Trust on Opinion Holders 34
   6.2 Trust on Advisers 37
   6.3 Trust on Information Providers 39

7 Conclusion 42
1 Introduction

What is feedback? Feedback is information about actions returned to the source of the actions.\footnote{http://en.wikipedia.org/wiki/Feedback\,(disambiguation)} For example, after listening to a piece of music, one can provide feedback on whether they enjoyed the piece or not, provide advice on how this performance may be improved, provide information about this piece, such as when and where was this piece performed, and so on. Feedback is usually intended to be used by the source of the actions to help them improve future actions. For example, if the community does not appreciate one’s music compositions, then it would be useful to pinpoint what aspects exactly the community does not like, or how can the performance be improved. Feedback is also commonly used to help others assess the trustworthiness of the source. For example, opinions (a special type of feedback) can be used to help assess the trustworthiness of a given issue in question. Say one is looking for a new saxophone player for his band, and he is doubtful about the performance of a player new to him. In such a case, peer opinions can help him make a decision on whether this new saxophone player is to be trusted or not. This second use of feedback is becoming extremely popular online, where feedback is crucial for the success of online systems such as Amazon and eBay. However, a crucial question that arises is how can feedback be trusted? What if a rating was biased because it was provided by a competitor? What if a piece of information provided is false because it was provided by a competitor to simply manipulate the community’s opinion? In this paper, we study how feedback itself may be assessed, or how to evaluate the trustworthiness of a given feedback.

Furthermore, we argue that in computational trust models, feedback is usually the main component in assessing the trustworthiness of various entities. Without any form of feedback, it would be difficult to assess these entities. In this paper, we illustrate how, by assessing the trust on feedback and feedback providers, we can generalise to assess the trust of any entity in the system, such as performances in PRAISE, teachers, good advisers, informative people, and so on. Hence the general name of the proposed model: TrustIt.

We categorise feedback into three different types: (1) opinions, (2) advice, specified as plans of actions, and (3) general information. An example of an opinion would be a statement such as: \textit{I do not like how you play ‘My Funny Valentine’}. An example of an advice would be a statement such as: \textit{To get better at playing ‘My Funny Valentine’ on time, you will have to practice it twice a week over a period of four months}. An example of information would be a statement such as: \textit{Miles Davis played ‘My Funny Valentine’}. The question then is how does one assess whether a given feedback is to be trusted or not.

The proposed trust model, TrustIt, follows an empirical approach where the trustworthiness of a given feedback is learnt from the history of past experiences. The proposal relies on probability theory as well as information theory. The basic idea is that past experiences (or past feedback) will help predict the quality of future experiences (or future feedback). The prediction is based on a probability measure: past experiences help build a probability distribution that would describe the possible outcomes of a future experience. For example, if one peer has always provided good plans in the past on issues related to jazz music, then one can expect this peer’s current proposed plan on improving one’s performance in jazz improvisation to be trustworthy enough, but maybe not his current proposed plan on how to train one’s soprano voice. We say that the more similar a past experience is to the current issue in question, then the more impact it would have when assessing
the issue in question. As such, and in addition to the probability theory and information theory foundations, similarity measures become of utmost importance to the proposed trust models, to which a decent effort is dedicated for specifying these similarity measures.

The rest of this paper is divided as follows. Section 2 provides a background summarising the existing trust models that are most relevant to the proposed work; Section 3 presents the proposed trust model for assessing the trustworthiness of opinion holders; Section 4 presents the proposed trust model for assessing the trustworthiness of advisers; Section 5 presents the proposed trust model for assessing the trustworthiness of information providers; Section 6 demonstrates the application of the proposed trust model in real life use cases; and Section 7 concludes with a brief summary of our proposal, its novelty, our plans for evaluating it, along with our plans for future research work.

2 Background

Trust is a mechanism for managing the uncertainty about autonomous entities and the information they deal with. As computer systems have become increasingly distributed, and control in those systems has become more decentralised, trust has become an increasingly important concept in computer science [4, 11]. For example, there have been studies on the development of trust in e-commerce [22], on mechanisms to determine which sources to trust when faced with multiple conflicting information sources [30], and mechanisms for identifying which individuals to trust based on their past activity [3]. Similarly, researchers have investigated trust and reputation specifically in peer-to-peer networks; for instance, proposing frameworks to prevent peers manipulating their trust values to obtain preferential downloads [2], or frameworks to assess the trustworthiness of a peer by the reputations of the other peers associated with, or connected to, that peer [13]. An interesting development is the idea of having individuals indentify each other by placing some form of financial guarantee on transactions that others enter into [8].

The propagation of trust, the transitivity of trust relations [29], and more complex relationships, such as ‘co-citation’ [12] have all been studied, and in many cases empirically validated [12].

Trust is an especially important issue from the perspective of autonomous agents and multiagent systems [27]. A key premise behind the multiagent systems field is that of developing software agents that will work in the interests of their owners, carrying out their owners’ wishes while interacting with other entities. In such interactions, agents will have to reason about the amount that they should trust those other entities, whether they are trusting those entities to carry out some task, or whether they are trusting those entities to not misuse crucial information. As a result we find much work on trust in agent-based systems. We list some of these below.

Several approaches, such as those of [7], [28], and [21], use a ‘contextual’ approach when dealing with trust, which is a central theme of this paper. The early model by Marsh [16] follows a utilitarian approach and time decay is modelled as a time window for experiences. The model proposed in [11] uses a qualitative degree approach to model trust and takes into account the context as well. The modelling of uncertainty is somewhat ad-hoc and not based on probabilistic grounds. The REGRET model [24] has some notions of time decay and follows a subjective modelling of the experiences. The overall notion of trust does not have a probabilistic meaning and is based on a utility modelling of the interactions (see [9] for a discussion). AFRAS [5] offers a model based...
on fuzzy sets with an entropic-like notion of uncertainty on the behaviour of other agents, which is similar to some parts of this proposal. [20] also based reputation on similarity between new contexts and past ones. They used 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. [25] distinguished between trust, which measures the expected deviations of behaviour in the execution of commitments, and honour, which measures the expected integrity of the arguments exchanged. They also distinguished between capabilities and trust. [26]’s approach compares what the agent has committed to to what is actually delivered, which this paper’s trust on advisers module refers to as observed. Although TrustIt achieves its goals through similarity measures, which are based on semantic matching and measures of action empowerment. Lastly, [10]’s approach is a more general approach that tries to understand the application domain and the structure of the trust model in order to match trust models with application domains.

In what follows we present two specific trust and reputation models that are most relevant to our proposed model. The first (Section 2.1) provides the basis of the TrustIt model that addresses the trust on advisers, whereas the second (Section 2.2) is used by the TrustIt model to help calculate specific similarity measures (as illustrated later on by Section 4.1.2).

2.1 Trust & Matching Algorithms for Selecting Suitable Agents

The trust model proposed by [18] addresses the problem of finding suitable collaborators in open distributed systems. The model essentially calculates the expectation about an agent’s future performance in a given context by assessing both the agent’s willingness and capability (as suggested by Castelfranchi and Falcone [6]) through the semantic comparison of the current context in question with the agent’s performance in past similar experiences. In what follows, we present the basic ideas behind this model, without going into much detail.

In [18], trust is based on the expectation of a particular observation given a commitment represented as a conditional probability:

\[ p(\text{Observe}(\alpha, \phi') \mid \text{Commit}(\rho, \phi)) \]

where the term \textit{Commit} has two arguments — the peer making the commitment (\rho) and the action it was committing to (\phi) — and the term \textit{Observe} has two arguments — the peer observing the outcome of the commitment (\alpha) and the outcome of the commitment which described \rho’s actual execution (\phi’). The idea is that past commitments can help 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.

This conditional probability is defined as the product between the capability of an agent and its actual willingness to achieve \phi:

\[ p(\text{Observe}(\alpha, \phi) \mid \text{Commit}(\rho, \phi)) = p(\text{Can}(\rho, \phi)) \cdot p(\text{Does}(\rho, \phi) \mid \text{Commit}(\rho, \phi)) \]

The capability of \rho (p(\text{Can}(\rho, \phi))) is assessed by checking the history of all past experiences of \rho and comparing the similarity between the actions performed in the past and those committed to now. The willingness (p(\text{Does}(\rho, \phi) \mid \text{Commit}(\rho, \phi))) is assessed by comparing with past experiences (as opposed to checking all past experiences when computing capabilities). The model
essentially goes over each past experience of $\rho$, and based on the similarity of that experience to the current commitment in question, the probability distribution describing the potential outcomes is modified (by learning from that experience). The similarity is mainly computed in this model based on the semantic similarity between the terms describing the commitments and observations of past experiences and comparing them to the current commitment in question.

Experiences are specified as records of pairs of the type:

$$\langle \text{Commit}(\rho, \phi), \text{Observe}(\alpha, \phi') \rangle$$

The agent assessing the trust in another ($\alpha$ in the case above) is the one observing the actions of $\rho$, and thus verifying if the expectations are met, since the executed/observed action might not be exactly the same as the one the agent committed to. Sometimes, it is assumed that an oracle (institution) existed to verify that the actions took place. For instance, a money transaction leaves a trace that can be followed.

Different definitions of trust are then proposed for translating the probability distribution $\mathbb{P}(\text{Observe}(\alpha, X)|\text{Commit}(\rho, \phi))$ into a trust measure. For instance, one approach is to compute the relative entropy with respect to an ideal distribution $\mathbb{P}_I(\text{Observe}(\alpha, X)|\text{Commit}(\rho, \phi))$, which would represent the ideal behaviour of the agent making the commitment from the point of view of the agent observing the commitment:

$$\text{Trust}(\alpha, \rho, \phi) = 1 - \sum_{\phi'} p_I(\text{Observe}(\alpha, \phi')|\text{Commit}(\rho, \phi)) \cdot \log \frac{p_I(\text{Observe}(\alpha, \phi')|\text{Commit}(\rho, \phi))}{p(\text{Observe}(\alpha, \phi')|\text{Commit}(\rho, \phi))}$$

Finally, the model proposes the notion of the decay of the value of information over time, which was later used in subsequent trust and reputation models like OpinioNet and the proposed model of this paper. In other words, the information provided by any probability distribution would lose its value over time and decay towards a default value.

We do not dwell further on this model in this paper, and we refer the interested reader to [18]. However, we note that this model has provided the basis for our trust on advisers module. The basic idea of using similar past experiences to predict the outcome of future experiences, the notion of decay of the value of information, some similarity measures of this model, along with the trust equations, have been reused in our trust on advisers module. Nevertheless, the trust on advisers module remains substantially different from this model. For instance, the action committed to is much more elaborate: a commitment does not represent a single action but a plan recommended to a given peer for fulfilling a given goal. As such, experiences are different in our model. The definition of the conditional probability is also different: it does not rely on the capability and willingness of the peer making the commitment, but on whether the peer the plan is recommended to is willing to adopt and execute the plan, and whether the execution of the plan actually fulfils the intended goals. In our model, comparing past experiences to a current commitment uses more sophisticated similarity measures which not only rely on semantic similarity but on measures of empowerment, which describe how much does one plan empower another. Similarly, our proposed model uses a substantially different approach in assessing how the probability distribution describing the expected outcome is modified, when learning from similar past experiences.
2.2 OpinioNet: The propagation of Opinions in Meronomies

Opinions and ratings, which are the basis of numerous trust and reputation mechanisms, are not always abundant. Their abundancy differs from one field to another. For example, while tons of data may be available on Amazon or eBay, very little information is available in the publications field. OpinioNet illustrates how opinions may be deduced in areas where such information is scarce. The main goal of OpinioNet is to provide the means for allowing a peer to deduce opinions about new entities, by learning from their own opinions on other related entities in a meronomy. In other words, given one peer’s opinions on a set of nodes of a meronomy, what can the peer deduce about its opinions concerning the remaining nodes? To achieve this, OpinioNet essentially provides an algorithm based on the propagation of opinions in meronomies.

The basic idea of the OpinioNet algorithm is that if a node in a meronomy does not receive a direct opinion, then its evaluation may be deduced from its children nodes’ evaluations. This is because the parent node is structurally composed of its children nodes. Hence, the evaluations on children nodes must necessarily influence the deduced evaluation on a parent node. OpinioNet refers to the direct opinion on a node or an opinion that is deduced from evaluations of the parts that compose that node as the ‘acquired opinion’. Additionally, an ‘inherited opinion’ is an opinion that is propagated down from parent nodes to their children. In the absence of information about the node itself, or the parts that compose it, information may be inherited from what one belongs to (the parents’ nodes). As an example of how the OpinioNet algorithm propagates evaluations through a meronomy, consider that if one performed well at the PreparingMeal activity of the meronomy of Figure 1, this will affect his evaluation for the activity of BabySitting. This would be an acquired evaluation. An inherited evaluation would be to say that performing well at BabySitting carries over to a good evaluation for the subactivity of ChangingNappies, if there is no direct evaluation of performing the ChangingNappies activity.

For further details on OpinioNet, we refer the interested reader to [19]. However, we summarise below the main equations describing the OpinioNet model, where OpinioNet’s meronomy is defined as the tuple \( M = (N, \mathcal{E}, D, I, A) \), where \( N \) is a set of nodes, \( \mathcal{E} : N \times N \) describes the

![Figure 1: A meronomy describing actions about caring for a child](image)
edges of the graph (in other words, \((n, c) \in E\) may be viewed as describing that \(c\) is a child node of \(n\)), \(D\) is the set of direct opinions \(o_n\) that nodes \(n \in N\) receive, \(I\) is the set of deduced inherited opinions \(h_n\) of the nodes \(n \in N\), and \(A\) is the set of deduced acquired opinions \(a_n\) of the nodes \(n \in N\). We note that opinions in OpinioNet (like opinions in our model) are specified as probability distributions over an evaluation space, where the evaluation space could be something like \{"i love it, i like it, i hate it\}.

One basic idea behind OpinioNet is that it relies on the attention that a node receives (where the attention is described through the direct opinions received) to assess the impact of the node on its neighbouring nodes. This approach is adopted because it is usually difficult to know what is the exact weight to assign to each child node when assessing its impact on its parent nodes, and vice versa. For this reason, the impact of a given node \(n\) at time \(t\) is then based on the proportion of nodes that have received a direct opinion in the structural sub-tree of \(n\), and it is defined accordingly:

\[
\pi_n = \begin{cases} 
1, & \text{if } \exists o_n \in D \\
0, & \text{if } \not\exists o_n \in D \land \not\exists c \in N \cdot (n, c) \in E \\
\frac{\sum_{(n, c) \in E} \pi_c}{|\{c \mid (n, c) \in E\}|}, & \text{otherwise}
\end{cases}
\] (1)

Another basic idea behind OpinioNet is that the direction of propagation in the structural graph is crucial, as illustrated earlier. It views the ‘downward’ propagation to provide the inherited opinion \(h_n\), while the ‘upward’ propagation to provide the acquired opinion \(a_n\). Then, each time a new opinion is added to a node in the graph, the \(a_n\) and \(h_n\) values of its neighbouring nodes are updating accordingly:

\[
a_n = \frac{1}{\sum_{(n, c) \in E} \pi_c} \cdot \sum_{(n, c) \in E} \pi_c \cdot O_c
\] (2)

\[
h_n = \frac{1}{\sum_{(p, n) \in E} \pi_p} \cdot \sum_{(p, n) \in E} \pi_p \cdot O_p
\] (3)

where, \(O_n\) describes the general deduced opinion of a node, and it is either equal to the acquired opinion \(a_n\) or the inherited opinion \(h_n\), based on which is considered more informative.

\[
O_n = \begin{cases} 
a_n, & \text{if } H(a_n) < H(h_n) \\
h_n, & \text{otherwise}
\end{cases}
\] (4)

where \(H(X)\) describes the entropy of a distribution \(X\), which describes how much informative a distribution is, and it is defined as:

\[
H(X) = -\sum X_i \cdot \log(X_i)
\]

Another basic idea behind OpinioNet is the notion of the decay of the value of information over time. In other words, the information provided by any opinion (specified as a probability distribution) would lose its value over time and decay towards a default value.
Finally, we note that the update of one node’s values triggers the update of its neighbouring nodes, resulting in a propagation wave throughout the structural graph.

We do not dwell further on OpinioNet in this paper, and we refer the interested reader to [19]. However, we note that OpinioNet is used by the trust on advisers module of our proposal to help compute how much does one action empower, or impact, another.

3 TrustIt\textsuperscript{O}: Trust on Opinion Holders

What is an opinion holder? An opinion holder is a peer who forms a view, or judgement, about something, not necessarily based on fact or knowledge. This ‘thing’ that the opinion is formed about could essentially be anything. It could describe other peers and their performances, products on the market, a piece of music, etc.

We define the opinion \( o_{tn}^\phi(\phi) \) that \( \alpha \) forms at time \( t_n \) about an issue \( \phi \) as a probability distributions as follows:

\[
o_{tn}^\phi(\phi) = \{e_1 \mapsto v_1, \ldots, e_n \mapsto v_n\}
\]

where, \( \{e_1, \ldots, e_n\} = E \) describes the evaluation space over which opinions are defined (e.g. \( E = \{i\text{ love } t, i\text{ like } t, i\text{ hate } t\}\)), and \( v_j \in [0, 1] \) represents the value assigned to each element \( e_i \in E \), with the condition that \( \sum v_j = 1 \). We note that probability distributions subsume classical approaches and are more informative.

We say the trust on a given opinion holder represents how much reliable are the opinions of the opinion holder with respect to a given context. We propose to calculate this trust measure in terms of how far were the past opinions of the same opinion holder from that of the group. As such, \( \alpha \)'s trust on \( \beta \) at time \( t_n \) with respect to its opinion on issue \( \phi \) is defined accordingly:

\[
\text{trust}^t(\alpha, \beta, o_\phi(\phi)) = \frac{\sum S_t(\phi, \phi') \cdot \text{emd}(o_{tn}^{i=0,\phi}(\phi'), o_{tn}^{\phi}(\phi'))}{\sum S_t(\phi, \phi')}
\]

where \( \phi' \sim_\phi \phi' \) specifies that \( \phi' \) is semantically similar to \( \phi \) with a minimum similarity level of \( \nu \), \( o_{tn}^{i=0,\phi}(\phi') \) describes \( \beta \)'s opinion on issue \( \phi' \) at time \( t_n \) (note that this opinion undergoes a decay from the time it was introduced \( t_i \) until the time the trust measure is calculated \( t_n \), which is based on the idea that the integrity of information decreases with time and, as such, the information provided by a probability distribution should also lose its value over time; Section 4.1.3 specifies in detail how probability distributions undergo a decay), \( o_{tn}^{\phi}(\phi') \) describes the group’s opinion on that same issue \( \phi' \) at time \( t_n \), and \( \text{emd} \) is the earth mover’s distance that calculates the distance (whose range is \([1, 0]\)) between two probability distributions [23].

\footnote{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 \( R \). The range of \( \text{emd} \) is \([0, 1]\), where 0 represents the minimum distance and 1 represents the maximum possible distance. However, we note that to use the earth mover’s distance, one also needs to determine what the distance between the terms of the ordered probability distribution’s support \( R \) is. In other words, one needs to define the matrix \( \delta = \{d_{ij}\}_{I,J\in R} \), where \( d_{ij} \) represents the distance between the elements \( i \) and \( j \) of the support \( R \).}
Concerning the semantic similarity between two issues, we say if an issue is defined by a set of keywords, then \( \phi' \sim_\nu \phi \equiv S_\nu(\phi, \phi') \geq \nu \), where \( S_\nu(\phi, \phi') \) describes the semantic similarity between sets of terms, as it is defined later on by Equation 15.

As for the group opinion \( o_{G_{\alpha}(\phi)}^{t_n}(\phi') \), it is calculated accordingly:

\[
o_{G_{\alpha}(\phi)}^t(\phi') = \sum_{\gamma \in G_{\alpha}(\phi)} o_{\gamma}^{t_n}(\phi') \bigg/ \left| G_{\alpha}(\phi) \right|
\]

where \( G_{\alpha}(\phi) \) describes the trusted group whose opinion matters at time \( t_n \) when calculating \( \alpha \)'s trust on \( \beta \) with respect to issue \( \phi \). Note that the individual opinions also undergo a decay from the time they were introduced \( t_i \) until the time the group opinion is calculated \( t_n \).

Of course, it is up to \( \alpha \) to decide who is the group \( G_{\alpha}(\phi) \) whose opinion matters at time \( t_n \) when assessing the trust on \( \beta \)'s opinion on issue \( \phi \). For instance, group \( G_{\alpha}(\phi) \) may represent the entire community, it may be composed of the single member \( \alpha \), or it may be \( \alpha \)'s selection of trusted community members. \( G_{\alpha}(\phi) \) may be manually defined by \( \alpha \) when needed.

One approach is to define group \( G_{\alpha}(\phi) \) accordingly:

\[
G_{\alpha}(\phi) = \{ \gamma \mid \gamma \in C \land \text{trust}_{\gamma}^{t_n}(\alpha, \gamma, o_\gamma(\phi)) \geq \chi_\alpha^\phi \}
\]

where \( C \) describes the set of community members, and \( \chi_\alpha^\phi \) describes \( \alpha \)'s threshold for selecting trusted members with respect to issue \( \phi \). If \( \chi_\alpha^\phi = 0 \), then \( \alpha \) is essentially trusting all community members; and if \( \chi_\alpha^\phi = 1 \), then \( \alpha \) is at least trusting itself. Note that \( G_{\alpha}(\phi) \), which is required for calculating the trust measure on an opinion holder, is built by relying on the same trust measure, but for other opinion holders. As such, the set \( G_{\alpha}(\phi) \) will evolve over time as new trust measures are computed.

In the PRAISE community, we consider community members to be human users, software agents, or web services. The set \( C \) of community members may then be defined as the union of the set of human users \( HU \), the set of software agents \( SA \), and the set of web services \( WS \): \( C = HU \cup SA \cup WS \). As such, the definition of \( G_{\alpha}(\phi) \) may also be modified in such a way that one can state that only specific web services are trusted to give opinions on a given issue \( \phi \) (in other words, \( \gamma \in C \) may be replaced with \( \gamma \in WS \) in the definition of \( G_{\alpha}(\phi) \) above), since web services’ opinions may be viewed to be more objective that human users’ opinions. For example, when assessing whether one member is good at giving opinions about ‘playing on time’, this member’s past opinions will be compared to those of the OnSetCalc web service, which assesses how well one plays on time. Then, it may be said that the closer the member’s opinions are to that of the web service’s, then the more trusted is this member at giving opinions about ‘playing on time’.

Finally, we note that our view of trust is that it describes one party’s assessment of another party’s trustworthiness, whereas reputation describes the group’s opinion of the party in question. The measure calculated by Equation 5 may be viewed as a reputation-based trust measure, since it relies on the group’s opinion, yet the group is defined by the peer assessing the trustworthiness of another.
3.1 Opinion Change over Time

An interesting aspect that may be assessed is the change of one’s opinion over time. Peers may then be categorised according to their personality that describes the pattern of opinion change. In what follows, we recognise a few personalities that might be of interest to pinpoint.

- **Indecisive.** A peer $\alpha$ is considered indecisive if $\alpha$’s opinion changes a lot over time (and discussions):

$$\sqrt{\frac{\sum_{t_i \in [t_i, t_{i+n}]} \text{emd}(o_{t_i}^{\alpha}(\phi), o_{t_{i+1}}^{\alpha}(\phi))^2}{n}} > \xi_i$$  \hspace{1cm} (7)

Note that we use the standard deviation to calculate how much $\alpha$’s opinion about a given issue $\phi$ changes over the time interval $[t_i, t_{i+n}]$, where $\xi_i$ specifies the threshold at which the peer $\alpha$ is considered indecisive.

- **Decisive.** A peer $\alpha$ is considered decisive if $\alpha$’s opinion does not change with time (and discussions):

$$\sqrt{\frac{\sum_{t_i \in [t_i, t_{i+n}]} \text{emd}(o_{t_i}^{\alpha}(\phi), o_{t_{i+1}}^{\alpha}(\phi))^2}{n}} < \xi_d$$  \hspace{1cm} (8)

where, $\xi_d$ specifies the threshold at which the peer $\alpha$ is considered decisive.

- **Adamant.** A decisive peer $\alpha$ is considered adamant if $\alpha$’s opinion at the end of the discussion is far from the group’s opinion:

$$\text{emd}(o_{t_j}^{\alpha}(\phi), o_{t_j}^G(\phi)) > \xi_a$$  \hspace{1cm} (9)

where, $\phi$ describes the issue that is being discussed, $G$ describes the group whose opinion is compared to that of $\alpha$’s, $t_j$ describes the discussion’s end time, and $\xi_a$ specifies the threshold at which the peer $\alpha$ is considered adamant.

- **Insightful.** A decisive peer $\alpha$ is considered insightful if $\alpha$’s opinion at the end of the discussion is close to the group’s opinion:

$$\text{emd}(o_{t_j}^{\alpha}(\phi), o_{t_j}^G(\phi)) < \xi_l$$  \hspace{1cm} (10)

where, $\xi_l$ specifies the threshold at which the peer $\alpha$ is considered a leader.

- **Persuader.** A leader $\alpha$ is considered a persuader if $\alpha$’s opinion initially was far from the group’s opinion, before the group finally converges towards $\alpha$’s opinion:

$$\text{emd}(o_{t_i}^{\alpha}(\phi), o_{t_i}^G(\phi)) > \xi_p$$  \hspace{1cm} (11)

where, $t_i$ describes the discussion’s start time, and $\xi_p$ specifies the threshold at which the peer $\alpha$ is considered a persuader.

Naturally, one may think of defining more personalities, but we list above only a few of those that we believe might be of interest to pinpoint.
To be able to identify such personalities, this requires the history of opinions. In some cases, such as assessing the adamant, leadership, and persuasive nature of peers, time needs to be divided into intervals that describe discussions, and the relevant opinions of peers should be captured before and after the discussion, with the assumption that any opinion change will be the result of that discussion. Figure 2 illustrates a discussion on a given issue $\phi$. It shows how a group of peers $G = \{\alpha, \beta, \ldots\}$ enter the discussion with initial opinions at time $t_i$, discuss the issue in question (possibly by providing arguments for or against $\phi$), and exit the discussion with updated opinions at time $t_j$.

In online scenarios, there would be peers providing opinions on a variety of issues over time, and discussions taking place in the form of peers posting comments. Identifying a concrete discussion and its relevant opinions is a challenge. Several simplifications may be made to address this issue. For instance, either each uttered statement (or comment) may be viewed as a discussion, or all statements may be grouped under one discussion. However, the challenge will be identifying which opinions are changing as the result of which statements, or discussions.

3.2 Opinion Change and the Trust Measure

From the point of view of calculating the trust in an opinion holder, one may say that Equation 5 may be enriched further by considering whether the opinion holder is usually decisive, a leader, or even persuasive. For example, we illustrate below how an opinion holder is considered trusted, not only if its past opinions were close to that of the group, but also if the opinion holder has historically been decisive.

$$
\text{trust}^n(\alpha, \beta, o_\beta(\phi)) = \sum_{\phi' \sim o_\phi} S_s(\phi, \phi') \cdot V^{|t_i-t_j|}(\phi') \cdot \text{emd}(o_\beta^{t_i \rightarrow t_j}(\phi'), o_{G(\phi)}^n(\phi'))
$$

$$
\text{sum}(\phi, \phi') = \frac{\sum_{\phi' \sim o_\phi} S_s(\phi, \phi') \cdot V^{|t_i-t_j|}(\phi') \cdot \text{emd}(o_\beta^{t_i \rightarrow t_j}(\phi'), o_{G(\phi)}^n(\phi'))}{\sum_{\phi' \sim o_\phi} S_s(\phi, \phi')}
$$
where $V^{[t_i,t_{i+n}]}_{\beta} (\phi)$ describes how much decisive $\beta$ has been in the past with respect to $\phi$:

$$V^{[t_i,t_{i+n}]}_{\beta} (\phi) = 1 - \sqrt{\sum_{t_x \in [t_i,t_{i+n}]} \text{emd}(o_{\beta}^{0}(\phi), o_{\beta}^{n+1}(\phi))^2}$$

4 TrustIt^A: Trust on Advisers

What is an adviser? An adviser is a peer that proposes plans for achieving goals. One could think of an adviser to represent planners or recommenders. Examples of advisers’ proposals would be: ‘If you go to Ferran Adria’s restaurant you will have the time of your life!’ or ‘If you study everyday for two hours you will get very good marks next semester’. These proposals, or advice, have two components: a plan to execute and a goal to achieve. In dynamic logic, these proposals could be formalised as: $[P_\eta]G$. That is, if $\eta$ performs plan $P$, then goal $G$ will be achieved. An adviser is a peer who recommends such proposals. Advisers may recommend generic plans that can be personalised for particular agents by adapting $[P_\eta]G$ to $[P_\eta]G$.

The question that this section addresses is: How much should one trust an adviser? In more precise terms, how much should $\alpha$ trust $\rho$ when $\rho$ recommends a plan $[P_\eta]G$? We get inspiration from previous work [18], where trust was based on the expectation of a particular observation given a commitment, which was specified as a conditional probability:

$$p(\text{Observe}(\alpha, \phi') | \text{Commit}(\rho, \phi))$$

where the term Commit had two arguments — the peer making the commitment ($\rho$) and the action it was committing to ($\phi$) — and the term Observe had two arguments — the peer observing the outcome of the commitment ($\alpha$) and the outcome of the commitment which described $\rho$’s actual execution ($\phi'$). 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 trust on an adviser, this expectation is specified as a conditional probability of observing a goal $G$ given $\rho$’s proposed plan $[P_\eta]G$ for achieving this goal:

$$p(\text{Observe}(\alpha, G) | \text{Commit}(\rho, [P_\eta]G))$$

where $P$ is the plan recommended by $\rho$ for $\eta$ in order to fulfil goal $G$, and $\alpha$ represents the party that observes the fulfilment of the goal $G$.

We note, however, that $G$ is only one of the possible outcomes that may be observed. As such, we define the probability distribution which describes the probabilities of all possible outcomes of a given proposed plan accordingly:

$$\mathbb{P} (\text{Observe}(\alpha, X) | \text{Commit}(\rho, [P_\eta]G))$$

The remainder of this section is divided as follows. Section 4.1 presents the preliminaries needed for understanding the proposed model, Section 4.2 presents how the probability distri-
D3.3 Trust and Reputation in Online Social Learning Communities

4.1 Preliminaries

The proposed model is an experienced-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 4.1.1) and similarity measures (Sections 4.1.2). Additionally, this section also presents the general concept of information decay (Section 4.1.3), which is a basic notion that underlies our work, as it describes how information loses its value over time.

4.1.1 Experiences

A Single Experience. When we talk about the trust on advisers, we are implicitly talking about the trust on their advice, or recommended plans. However, the recommended plans that we are interested in assessing are conditional statements of the form: ‘if the recommended plan is executed, then the intended goal will be achieved’. As such, past experiences should not only keep track of recommended plans and their achieved goals, but of the fulfilment of the conditional part of these plans as well. This is because the adviser might have good advice, but the peer \( \eta \) has not been fulfilling its duties in carrying out the recommended plans. An experience should keep note of several issues (as motivated by Section 4.2):

- **The adviser’s recommended plan.** We interpret recommended plans as commitments made by the adviser that the goal \( G \) will be achieved if \( \eta \) executes plan \( P \). A recommended plan is specified as \( \text{Commit}(\rho, [P_\eta|G])_{t_i} \), where \( t_i \) specifies the time at which the plan \( [P_\eta|G] \) was recommended by \( \rho \).

- **The adopted plan.** An adopted plan is then interpreted as a commitment made by the peer adopting the plan and it describes the peer’s willingness to execute this plan. It is specified as \( \text{Commit}(\beta, P')_{t_j} \), where \( t_j \) describes the time at which the peer adopts the plan.

- **The outcome of executing the adopted plan.** This outcome needs to be observed by some peer \( \gamma \), and this observation is specified as \( \text{Observe}(\gamma, P''_{\beta})_{t_m} \), where \( t_m \) describes the time at which \( \gamma \) observed the execution of the adopted plan.

- **The realised goal.** The realised goal needs to be observed by some peer \( \alpha \), and this observation is specified as \( \text{Observe}(\alpha, G')_{t_n} \), where \( t_n \) describes the time at which \( \alpha \) observed the realised goal.

A single experience \( \mu \) is then specified as follows:

\[
\mu = (\text{Commit}(\rho, [P_\eta|G])_{t_i}, \text{Commit}(\beta, P')_{t_j}, \text{Observe}(\gamma, P''_{\beta})_{t_m}, \text{Observe}(\alpha, G')_{t_n})_{t_i < t_j < t_m < t_n}
\]
Different performers can pickup \( \rho \)'s plan: while the plan may be recommended to \( \eta \), \( \beta \) may decide to adopt it. Additionally, the performer \( \beta \) may commit to a variation of the plan: \( P' \neq P \). For example, assume a recommended plan that states that one should “practice his piano twice a day” and a performer who decides to adopt a variation of this plan by committing to “practicing his piano once a day”. Finally, 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 as well, and different from the adviser and the one being advised. Although there may arise cases where \( \gamma = \alpha \), or \( \rho = \gamma \), or \( \beta = \alpha \), and so on.

Each element of the experience should have a different time-stamp. An integrity constraint is then needed to check that a plan is adopted (by committing to it) after it has been recommended by an adviser, and that the plan’s execution has been observed after it has been adopted, and that the goal’s fulfilment has been assessed after the plan has been executed. This integrity constraint is specified by the condition \( t_i < t_j < t_m < t_n \).

**History of Experiences.** Each single peer will have its own history of experiences, and we use the notation \( H_\alpha = \{ \mu_1, \ldots, \mu_n \} \) to describe \( \alpha \)'s history of experiences.

Populating the history of experiences needs to address numerous issues, such as how is information collected, how do peers register their commitments to plans, how is the causality between elements recognised (for example, recognising that observing goal \( G \) is the result of observing the execution of plan \( P \), or that observing plan \( P' \) is the result of committing to \( P' \), or that committing to \( P' \) is the result of following the recommended plan \( [P|G] \), whose duty is it to observe certain outcomes, how are observers guaranteed (or incited) to fulfil their duties correctly, and so on.

We do not dwell much on how a peer’s history of experience is populated, as this needs to be specified by the peer itself. For instance, the peer will specify how indirect experiences are incorporated, and which observers are trusted. And in the case of a centralised system with a centralised history of experiences, we say the community’s charter is responsible for specifying how the issues related to populating the history \( H \) are to be addressed.

### 4.1.2 Similarity Measures

When assessing the level of similarity between past experiences and a current experience, there is a variety of relevant similarity measures that address different aspects of experiences, such as the similarity of plans, the similarity of goals, or the similarity of peer capabilities with respect to plans. This section is dedicated to defining these similarity measures.

**Plan and Goal Similarity.** We assume plans are sets of actions, where the set off all actions \( U \) is the set of nodes of the action taxonomy \( T_A \). And we say that there is a semantic similarity relationship between actions \( S_t : U \times U \rightarrow [0, 1] \) that shows the degree of relationship between the actions. We also assume goals to be a conjunction of terms, where the set of all terms describing goals \( V \) is the set of nodes of the goal taxonomy \( T_G \). And we say that there is a semantic similarity relationship between terms describing goals \( S_t : V \times V \rightarrow [0, 1] \) that shows the degree of relationship between the goals.
Plan similarity and goal similarity are then computed in the same manner following the following equation:

\[
S_x(Q, Q') = \frac{1}{2} \left( \min_{\phi \in Q, \phi' \in Q'} \{ \max \{ S_x(\phi, \phi') \} \} + \min_{\phi' \in Q, \phi' \in Q'} \{ \max \{ S_x(\phi, \phi') \} \} \right)
\]

(15)

where \( \phi \in Q \) either describes an action \( \phi \) of the plan \( Q \)'s set of actions (if \( Q \) was a plan), or it describes a term \( \phi \) in the goal \( Q \)'s conjunction of terms (if \( Q \) was a goal), and \( S_x \) describes the semantic similarity between terms, or actions.

But what is the motivation behind choosing this approach for calculating \( S_x \)? 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, as illustrated by Figure 3, there are a number of conjunctive operators that may be used, such as the product operator \((\prod)\); and the minimum operator \((\min)\) separates between conjunctive operators and the average ones. We adopt the minimum operator, which describes an optimistic approach. For instance, if we are comparing \( a \land b \) to \( c \) and \( S_x(a, c) = 0.3 \) and \( S_x(b, c) = 0.2 \), then we have \( \min \{ S_x(a, c), S_x(b, c) \} = 0.2 \). For a more pessimistic approach, one can even replace the minimum operator \((\min)\) with the product operator \((\prod)\). In this case, \( \prod S_x(a, c), S_x(b, c) \) = 0.06, which is drastically smaller than considering the minimum. We note that the choice of operator will be domain dependant. We also note that in this example, we are only considering the minimum from the point of view of the first set \((a \land b)\). To maintain symmetry (i.e. \( S_x(Q, Q') = S_x(Q', Q) \)), and as illustrated by Equation [15] above, one also needs to consider the minimum from the point of view of the second set \((c)\). The function \( S_x \) is symmetric, and its range is \([0, 1]\).

![Figure 3: The classification of operators as conjunctive, average, and disjunctive](image)

Finally, we adopt the following definition of semantic similarity [15]:

\[
S_x(\phi, \phi') = e^{-\kappa_1 l} \cdot \frac{e^{\kappa_2 h} - e^{-\kappa_2 h}}{e^{\kappa_2 h} + e^{-\kappa_2 h}}
\]

(16)

where \( e \) is Euler’s number, \( l \) is the length (i.e. number of hops) of the shortest path between the terms \( \phi \) and \( \phi' \) in a taxonomy, \( h \) is the depth of the deepest concept subsuming both concepts, and \( \kappa_1 \) and \( \kappa_2 \) are parameters scaling the contribution of shortest path length and depth, respectively. Essentially, \( \kappa_1 \) and \( \kappa_2 \) are parameters that \( \alpha \) could use to customise the weight given to \( l \) and \( h \), respectively. The function \( S_x \) is symmetric (i.e. \( S_x(\phi, \phi') = S_x(\phi', \phi) \)), 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 is-a graph. The measure we use 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 16 just as an example. As such, we refer the interested reader to [15] for further details on Equation 16 and we stress that alternative approaches can easily be used to replace this equation. There is no universal measure for semantic similarity, and this usually depends on the structure of the taxonomy amongst other things. Different contexts and different taxonomies may require different approaches and equations. Similarly, different agents may also prefer different equations for their own taxonomies. This is outside the scope of this paper and deserves a dedicated line of work.

Plan Empowerment. When considering the capabilities of performing ‘similar’ actions, we are interested in measures of empowerment as opposed to semantic similarity measurements. For example, driving a truck and driving a car may be similar. However, if \( \alpha \) 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 or not. As illustrated above in the driving example, empowerment measures are not usually symmetric. While similarity measures are computed by considering taxonomies (based on the \textit{is-a} relation), empowerment measures will be computed by considering meronomies (based on the \textit{empowered-by} relation). Note that we argue that the \textit{empowered-by} relation is just another form of the \textit{part-of} relation. Figure 1 provides a sample meronomy for actions about caring for a child [14]. The arrows describe the \textit{part-of} relations, which may also be interpreted to represent \textit{empowered-by} relations. For instance, the action GiveALift is composed of two sub-actions: PickingUp and DroppingOff. These sub-actions are part-of the GiveALift parent action, and we say they are empowered by that parent action. In other words, if someone is capable of GivingALift for a child, then they will be capable of both PickingUp and DroppingOff a child.

To compute the empowerment measure between two nodes of a meronomy, we make use of the OpinioNet algorithm. OpinioNet essentially illustrates how opinions may propagate from one node in a meronomy to another, losing some of their value as they move farther away from the original node. The result of this propagation may be viewed as describing the level of influence of one node on another. We argue that this level of influence may be understood in terms of the empowerment of one action node on another.

To calculate the empowerment measure from node \( \phi \) to \( \phi' \), we assume that node \( \phi \) is the only node in the graph that receives a direct opinion, and the opinion it receives is \( T \) (where \( T \) is the highest opinion possible, for instance \( T = \{ T(1) \mapsto 0, \ldots, T(n - 1) \mapsto 0, T(n) \mapsto 1 \} \)). We then propagate this opinion in the graph, which would result in a deduced opinion about \( \phi' \). The difference between this deduced opinion and \( T \) describes the empowerment measure from node \( \phi \) to \( \phi' \). Formally, we say:

\[
E(\phi, \phi') = |emd(T, oNet(M, T, \phi, \phi'))|
\]

where \( oNet \) computes (and returns) the deduced opinion at \( \phi' \) by propagating \( T \) from \( \phi \) in meronomy \( M \) following the OpinionNet propagation algorithm, and \( emd \) is the earth mover’s distance that calculates the distance (whose range is \([1, 0]\)) between two probability distributions [23].

If probability distributions are viewed as piles of dirt, then the earth mover’s distance measures the minimum cost
Algorithm 1  

\[ oNet(M, T, \phi, \phi') \]

\textbf{Require:} \( M = (N, E, D, A, I) \) \( \triangleright \) This describes a meronomy \( M \), where \( N = \{ \phi, \phi', \ldots \} \) is the set of nodes, \( E : N \times N \) describes the edges of the graph, \( D \) is the set of direct opinions, \( I \) is the set of inherited opinions, and \( A \) is the set of acquired opinions

\textbf{Require:} \( F = \{ F(1) \mapsto \frac{1}{n}, \ldots, F(n) \mapsto \frac{1}{n} \} \) \( \triangleright \) This is a probability distribution describing complete ignorance, and we choose \( n = 5 \)

\textbf{Require:} \( T = \{ T(1) \mapsto 0, \ldots, T(n - 1) \mapsto 0, T(n) \mapsto 1 \} \) \( \triangleright \) This is a probability distribution describing the highest opinion possible, and again, we choose \( n = 5 \)

\textbf{Require:} \( \text{OpinioNet}(M) \) \( \triangleright \) This initiates the propagation of opinions in \( M \) via OpinioNet

\[ D = A = I = \emptyset \]

\textbf{for all} \( \phi \in N \) \textbf{do}

\[ a_{\phi} = h_{\phi} = F; \]

\textbf{end for}

\[ a_{\phi} = T; \]

\text{OpinioNet}(M); 

\textbf{return} \( O(\phi') \); 

algorithm is described by Algorithm 1, which essentially sets the acquired and inherited opinions of all nodes of meronomy \( M \) to the default uniform probability distribution \( F \), and introduces the single direct opinion \( T \) at node \( \phi \), before calling OpinioNet to propagate this direct opinion.

We note that although OpinioNet also has the notion of the decay of the value of information over time, our use of OpinioNet is not to maintain the reputation of nodes of a meronomy over time, but to calculate the impact of one node on another. As such, in our use of OpinioNet, the notion of time does not exist, and the notion of decay becomes irrelevant.

As long as the empowerment meronomy does not change, empowerment measures will be fixed. Since such meronomies seldom evolve, empowerment measures may then be computed in advance (or when the meronomy changes) for every pair of nodes.

Based on Equation [17] we define the empowerment measure of plan \( P \) on plan \( P' \) accordingly:

\[
E_p(P, P') = \min_{a \in P} \{ \max_{a' \in P} \{ E(a, a') \} \}
\]  (18)

Note that we use the ‘min-max’ operators, following the same reasoning as Equation [15].

\textbf{Similarity of Peer Capabilities with respect to Plans.} We believe it is important to compare the capability of peers with respect to plans. For example, assume a past proposed plan recommended that \( \alpha \) drives a car, and a current proposed plan recommends \( \beta \) to drive a car and park it. To assess

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 \( R \). The range of \(emd\) is \([0, 1]\), where 0 represents the minimum distance and 1 represents the maximum possible distance. However, we note that to use the earth mover’s distance, one also needs to determine what the distance between the terms of the ordered probability distribution’s support \( R \) is. In other words, one needs to define the matrix \( \delta = \{ d_{ij} \}_{i,j \in R} \), where \( d_{ij} \) represents the distance between the elements \( i \) and \( j \) of the support \( R \).

\( ^4 \) We choose \( n = 5 \) following the common online approaches of providing opinions based on 5-star ratings, which also keeps us in line with [17]’s recommendations of limiting the size of the evaluation space between 5 and 9.
whether the current intended goal will be achieved, it is not only important to compare the current plan with the past plan, but also to compare the peers’ capabilities with respect to their proposed plans. For instance, \( \alpha \) might know how to drive a car, and hence the goal was reached in the past experience; whereas, \( \beta \) does not know how to drive a car, and hence the goal might not be reached in the current experience.

We define the similarity measure of peer capabilities with respect to plans accordingly:

\[
S_c(\alpha, \beta) = | p(Can(\alpha, P)) - p(Can(\beta, P')) | \tag{19}
\]

where \( p(Can(\alpha, P)) \) describes the capability of \( \alpha \) executing plan \( P \), and it is specified as a probability learned from past experiences, as follows:

\[
p(Can(\alpha, P)) = \min_{a \in P} \max_{a' \in P'} \{ E(a', a) \} \tag{20}
\]

where, \( a \in P \) describes an action \( a \) of the set of actions of plan \( P \), \( P' \) describes the past plans that \( \alpha \) has performed, \( P'' = \bigcup_{\langle \langle \alpha, \text{Observe}(\alpha, P''') \rangle \in H} P''' \) and \( E(a', a) \) describes how much does action \( a' \) empower action \( a \). Note that when assessing capabilities, the empowerment measure is used as opposed to the semantic similarity measure. Again, we use the ‘min-max’ operators, following the same reasoning as Equation 15.

### 4.1.3 Decay of Information

An important notion to our proposal is the notion of the decay of information. We say the integrity of information decreases with time. In other words, the information provided by any probability distribution \( P(\text{Observe}(\alpha, X) \mid \text{Commit}(\rho, [P]G)) \) as well as others 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 \( t_n \to \infty \). Such background knowledge may be expressed in terms of an individual’s knowledge, and is represented as a decay limit distribution \( D_x \), where \( x \) describes the specific context. If the background knowledge is incomplete then one possibility is to assume that \( D \) has maximum entropy whilst being consistent with the data.

In summary, given a distribution, \( P \), and a decay limit distribution \( D_x \), \( P \) decays by:

\[
P^{t_n} = \Lambda(D_x, P^{t_{n-1}}) \tag{21}
\]

where \( \Lambda \) is the decay function satisfying the property: \( \lim_{t_n \to \infty} P^{t_n} = D_x \).

\[3\]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.
One possible definition for $\Lambda$ could be:

$$\mathbb{E}^{t_n} = \nu^{\Delta t_n} \cdot \mathbb{E}^{t_{n-1}} + (1 - \nu^{\Delta t_n}) D_x$$

where $\nu$ is the decay rate, and:

$$\Delta t_n = \begin{cases} 
0, & \text{if } t_n - t_{n-1} < \omega \\
1 + \frac{t_n - t_{n-1}}{t_{\max}}, & \text{otherwise}
\end{cases}$$

The definition of $\Delta t_n$ above serves the purpose of establishing a minimum grace period during which the information does not decay, and that once reached the information starts decaying. The period of grace is determined by the parameter $\omega$. We then introduce the parameter $t_{\max}$, which may also be defined in terms of multiples of $\omega$, to control the pace of decay. The main idea behind this is that after the grace period, the decay happens very slowly; in other words, $\Delta t_n$ 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: $\Lambda_{tn}$ and $D_{tn}$.

### 4.2 Probability of a Proposed Plan Fulfilling its Goal

The main question to answer when defining the trust of $\alpha$ on $\rho$ with respect to a proposed plan $[P_\eta]|G$ is how to build the following probability distribution from the memory of previous experiences:

$$\mathbb{P}(\text{Observe}(\alpha,X)|\text{Commit}(\rho,[P_\eta]|G))$$

We argue that building such a probability distribution requires more than a simple check of what goals have similar proposed plans fulfilled in the past. For instance, if we are interested in calculating the probability of observing that goal $G$ has been achieved, given that $\rho$ has recommended plan $[P_\eta]|G$, a number of things must happen:

- **$\eta$ must be willing to perform $P$.** We say first of all and foremost, $\eta$ must commit to the execution of plan $P_\eta$. This describes $\eta$’s willingness to follow $\rho$’s recommendation. Several motivations may exist for $\eta$ to be willing to perform $P$. For example, $\eta$ might be interested in achieving goal $G$ and it trusts $\rho$’s advice that $P$ achieves $G$; or $\eta$ might be interested in executing $P$ for other personal reasons; or $\eta$ might have been instructed by its supervisor to execute plan $P$ and it cannot afford the ramifications of not following its supervisor’s instructions. In this proposal, we do not dwell on the motivations for $\eta$ to be willing to perform a given plan, as this is outside the scope of this research line. Instead, we assess $\eta$’s willingness as a probability distribution (conditional to $\rho$ recommending plan $[P_\eta]|G$) that is learnt from past experiences, by studying what has $\eta$ committed to in the past and how similar are these past commitments to the current commitment in question.

- **$\eta$ must be capable of performing $P$.** In addition to willing to perform $P$, we say $\eta$ needs to be capable of performing $P$. We assume peers may be capable of performing actions they
are not willing to perform and vice versa. This is because we do not only consider rational peers that only make commitments that they know they can fulfil, but we also consider fraudulent and malicious peers that may lie about their capabilities, or peers that might be mistaken about their own capabilities who realise after making their commitments that such performances are not actually possible. In such systems, capabilities are independent from willingness. Again, we assess η’s capability as a probability distribution (conditional to η willing to perform P) that is learnt from past experiences, by studying what plans has η actually executed in the past and how similar are these to the current plan in question.

- **P must be effective in realising G**. Last, but not least, we need to assess whether a plan reaches the objective the recommender claims it will achieve. This means we need to assess whether the α observes G when η actually performs P_η. Again, we assess this causality as a probability distribution (conditional to η actually performing P_η) that is learnt from past experiences.

To generalise, we say the probability of α observing a goal G’, given that ρ recommended plan [P_η]G, is assessed by the product of the probability of some peer β committing to ρ’s recommended plan (or a variation of it) with the probability of β executing the plan it committed to (or a variation of it) and the probability of the goal G’ being achieved as the result of the executed plan:

\[ p(\text{Observe}(\alpha, G') \mid \text{Observe}(\gamma, P_\beta'')) \cdot p(\text{Observe}(\gamma, P_\beta'') \mid \text{Commit}(\beta, P')) \cdot p(\text{Commit}(\beta, P') \mid \text{Commit}(\rho, [P_\eta]G)) \]

Note that although plan P was recommended to η, an alternative peer β may pick up this recommendation and commit to a variation of the recommended plan P (P’). β might also execute a variation of the plan P’ it committed to (P’’). Finally, a variation of the goal G might eventually be achieved (G’).

As such, the probability of observing G’, given that ρ recommended plan [P_η]G, is then specified as the summation of the product above for all possible committed to or executed plans L (i.e. ∀P’ ∈ L, P’’ ∈ L).

\[
p(\text{Observe}(\alpha, G') \mid \text{Commit}(\rho, [P_\eta]G)) = \sum_{P_\beta, P_\beta' \in L} p(\text{Observe}(\alpha, G') \mid \text{Observe}(\gamma, P_\beta'')) \cdot p(\text{Observe}(\gamma, P_\beta'') \mid \text{Commit}(\beta, P')) \cdot p(\text{Commit}(\beta, P') \mid \text{Commit}(\rho, [P_\eta]G)) \quad (23)
\]

We note that \( p(\text{Observe}(\alpha, G') \mid \text{Commit}(\rho, [P_\eta]G)) \) describes the probability of observing G’. To calculate all possible observed outcomes of ρ recommending plan [P_η]G, then the probability distribution \( \mathbb{P}(\text{Observe}(\alpha, X) \mid \text{Commit}(\rho, [P_\eta]G)) \) needs to be built accordingly:

\[
\mathbb{P}(\text{Observe}(\alpha, X) \mid \text{Commit}(\rho, [P_\eta]G)) = \left\{ \begin{array}{l} p(\text{Observe}(\alpha, G') \mid \text{Commit}(\rho, [P_\eta]G)), \\ p(\text{Observe}(\alpha, G'') \mid \text{Commit}(\rho, [P_\eta]G)), \\ \cdots \end{array} \right\} \quad (24)
\]

where \{G’, G’’, …\} = G describes the set of all potential outcomes.

The question that arises then is how are the sets of all potential outcomes G and all potential committed to or executed plans L calculated? Several approaches may be followed. For example, semantic similarity may be used to calculate potential outcomes, plans, or performers. For
instance, for a given recommended plan $[P_\eta]|G$, one can say that all goals that are semantically similar to the intended goal $G$ can be included in $G$, or all plans that are semantically similar to plan $P$ can be included in $\mathcal{L}$. However, in this paper we follow a simpler approach such that for a given recommended plan $[P_\eta]|G$, we have the initial set all potential outcomes $G = \{G, other\}$ and the initial set of all potential committed to or executed plans $\mathcal{L} = \{P, other\}$. Then, with every new experience, a new achieved goal may appear, which is then added to the set $G$; or a new committed to or executed plan may appear, which is then added to the set $\mathcal{L}$. In other words, the sets $G$ and $\mathcal{L}$ grow with the experiences.

In what follows, we present our proposed approach for calculating the different probability distributions required by Equation 23 above for predicting the outcome of the recommendation $[P_\eta]|G$ provided by $\rho$.

4.2.1 Probability of Committing to a Proposed Plan

For simplification, we use the notation $P_{\eta}^t(X_t | [P_\eta]|G)$ to describe the probability distribution of committing to a plan $X$ at time $t$, given that the recommended plan $[P_\eta]|G$ has been recommended. In other words, $P_{\eta}^t(X_t | [P_\eta]|G) = P(\text{Commit}(X_t | [P_\eta]|G))$.

Essentially, $P_{\eta}^t(X_t | [P_\eta]|G)$ is calculated incrementally. At time $t_0$, and with the lack of any experiences, we say the initial value of such a probability distribution ($P_{\eta}^0(X_t | [P_\eta]|G)$) should be domain dependent. Alternatively, for domain independent values, one can choose $P_{\eta}^0(X_t | [P_\eta]|G) = \mathbb{F}$, the uniform distribution, to describe ignorance. As new experiences are encountered, this probability starts to get reshaped by incorporating the information learned from these new experiences.

We say, given a new experience $\mu = \langle \text{Commit}(\rho', [P_\eta']|G'), \text{Commit}(\beta, P') \rangle \in \mathbb{R}$, we calculate its relevance to $P_{\eta}^t(X_t | [P_\eta]|G)$ as follows:

$$R_{\mu}^{\rho_{\eta}} = \frac{\zeta_e \cdot S_e(G', G) + \zeta_p \cdot E_p(P', P) + \zeta_r \cdot S_r(P_\eta', P_\eta) + \zeta_c \cdot S_c(\rho', \rho) + \zeta_r \cdot S_r(\eta', \eta)}{\zeta_e + \zeta_p + \zeta_r + \zeta_c} \tag{25}$$

where $\zeta_e$, $\zeta_p$, $\zeta_r$, $\zeta_c$, and $\zeta_e$ are parameters that help specify the impact of (or weight given to) each similarity measure. The basic idea behind Equation 25 is that an experience is considered relevant if the goals are similar (specified by $S_e(G', G)$) and the past plan empowers the newly recommended plan (specified by $E_p(P', P)$). Note that we use plan empowerment as opposed to plan similarity because the capability of performing a plan is relevant here. Plan empowerment illustrates the level to which one plan may imply another. In addition to comparing plans and goals, we say that peers’ capabilities are also crucial (specified by $S_r(P_\eta', P_\eta)$). For example, $\eta'$ might be capable of executing $P'$, but $\eta$ might not be capable of executing $P$. And even if all the other components of the experiences are extremely similar, if the degrees of the peers’ capabilities each with respect to their suggested plans are very different, then one can expect the outcome to be very different. Finally, we say that the similarity of the peers’ roles, whether the advisers ($\rho'$ and $\rho$) or the peers expected to execute the plans ($\eta'$ and $\eta$), may be important in some cases, as discussed later by Section 4.3.2.

Additionally, we say that if one is interested in calculating how well $\rho$ performed in the past, then one must set the condition $S_e(p', \rho) = 1$. However, if one is interested in calculating how well did similar roles (say the ‘teacher’, the ‘mentor’, the ‘tutor’, and so on) perform in the past, then one must have $S_e(p', \rho) < 1$. The same reasoning is followed for the peers expected to execute...
the plans; comparing their roles might also be of interest in some cases. For example, one might be interested in calculating how well \( \rho \) performed in the past when recommending plans for a specific role (in which case, the condition \( S_t(\eta', \eta) = 1 \) must hold), or similar roles (in which case, \( S_t(\eta', \eta) < 1 \)).

Note that the proposed approach assumes that peers will be part of a taxonomy of ‘roles’. We argue that this is a natural assumption as the need for roles in communities comes hand in hand with the need for community norms or rules. For instance, one need for defining roles for a community is to specify the rights and duties of a set of community members. Online communities are no exception: online forums have administrators, moderators, common users, etc.; Wikipedea has administrators, reviewers, registered users, unregistered users, etc.; an online music class may have teachers, students, etc.; and so on. Instantiations of roles, for example the specific teacher ‘Alicia’, will be leaf nodes in the taxonomy of roles. As such, the similarity between peers \( S(\rho', \rho) \) and \( S(\eta', \eta) \) of Equation \( 25 \) will be based on the similarity between nodes of a taxonomy, which was defined earlier by Equation \( 16 \).

Also note that we use the weighted arithmetic average in Equation \( 25 \) to aggregate these five measures of similarity as opposed to conjunctive operators such as the product (see Figure 3 and its discussion for a brief classification of these operators), since we believe conjunctive operators could be too extreme. For instance, assuming \( \zeta = \zeta_p = \zeta_c = \zeta_t = 1 \), if all similarity measures and very high and equal to 0.9, then the product will result in the relevance measure to be much lower and equal to 0.59!

We then calculate the probability of having the outcome where \( \beta \) commits to \( P'' \), based on the relevance of the experience \( \mu \):

\[
p_c^\beta(X = P'' | \{P_n\} | G) = p_c^{\beta-1}(X = P'' | \{P_n\} | G) + (1 - p_c^{\beta-1}(X = P'' | \{P_n\} | G)) \cdot \varepsilon_c \cdot R_{\mu}^{\text{mre}}
\]  

where \( \varepsilon_c \) describes how much should the probability of a given point increase (assuming the experience in question was totally relevant: \( R_{\mu}^{\text{mre}} = 1 \)), and it is specified as a percentage of the maximum amount that this probability can increase \((1 - p_c^{\beta-1}(X = P'' | \{P_n\} | G))\).

Note that if \( P''_\beta \) was not already in the support of the distribution \( P_c^{\beta-1}(X | \{P_n\} | G) \), then that would be because the probability \( p_c^{\beta-1}(X = P'' | \{P_n\} | G) = 0 \). In such a case, the equation above will result in introducing \( P''_\beta \) as a new element in the support of the distribution \( P_c^{\beta}(X | \{P_n\} | G) \). Also note that initially, the support is set to \{\( P_n, \text{other} \)\}, and it increases incrementally based on new experiences.

\[ p_c^\beta(X = P'' | \{P_n\} | G) \] essentially specifies the probability at a given point \( X = P'' | \{P_n\} | G \) of the probability distribution \( P_c^{\beta}(X | \{P_n\} | G) \). The probability distribution \( P_c^{\beta}(X | \{P_n\} | G) \) is then calculated accordingly:

\[
P_c^{\beta}(X | \{P_n\} | G) = \text{mre}(p_c^\beta(X = P'' | \{P_n\} | G), p_c^{\beta-1}(X | \{P_n\} | G))
\]  

where \( \text{mre} \) is a function that modifies the previous probability distribution (in this case \( P_c^{\beta-1}(X | \{P_n\} | G) \)) in such a way that the new distribution is a distribution with minimum relative entropy and it satisfies the constraint imposed by the new probability \( p_c^\beta(X = P'' | \{P_n\} | G) \).

Finally, we remind the reader that the information provided by the probability distribution \( P_c^{\beta}(X | \{P_n\} | G) \), like any other piece of information, will all be subject to decay towards a predefined decay limit distribution.
4.2.2 Probability of Enacting a Committed-To Plan

For simplification, we use the notation \( P^\circ_e (X_\beta | P_\beta) \) to describe the probability distribution of \( \beta \) executing a plan \( X \) at time \( t_0 \) given that it has committed to the plan \( P \). In other words, \( P^\circ_e (X_\beta | P_\beta) = \mathbb{P}(\text{Observe}(\gamma, X_\beta) | \text{Commit}(\beta, P)) \).

Again, \( P^\circ_e (X_\beta | P_\beta) \) is calculated incrementally. At time \( t_0 \), and with the lack of any experiences, we say the initial value of such a probability distribution \( P^\circ_e (X_\beta | P_\beta) \) is domain dependent. Alternatively, for domain independent values, one can choose \( P^\circ_e (X_\beta | P_\beta) = \mathbb{P} \), the uniform distribution, to describe ignorance. As new experiences are encountered, this probability starts to get reshaped by incorporating the information learned from these new experiences.

We say, given a new experience \( \mu = \langle \text{Commit}(\beta', P') \rangle_{\text{obs}}, \text{Observe}(\gamma', P''_\beta) \rangle_{\text{obs}} \), we calculate its relevance to \( P^\circ_e (X_\beta | P_\beta) \) as follows:

\[
R^\circ_e = \frac{\zeta_u \cdot E_p (P', P) + \zeta_v \cdot S_e (P', P_\beta) + \zeta_w \cdot S_i (\beta', \beta)}{\zeta_u + \zeta_v + \zeta_w} \tag{28}
\]

where \( \zeta_u, \zeta_v, \) and \( \zeta_w \) are parameters that help specify the impact of (or weight given to) the similarity measures. Equation\[28\] follows a similar reasoning to that of Equation\[25\]. The basic idea behind Equation\refeq:relevanceEnact is that an experience is considered relevant if the past plan empowers the newly recommended plan (specified by \( E_p (P', P) \)). In other words, if one was capable of driving a truck in the past, then he will most probably be capable of driving a car now, but not vice versa. In addition to comparing plans, we say that peers’ capabilities are also crucial \( (S_i (\beta', \beta)) \). For example, \( \beta' \) might be capable of executing \( P' \), but \( \beta \) might not be capable of executing \( P \). And even if the plans are extremely similar, if the degrees of the peers’ capabilities each with respect to their suggested plans are very different, then one can expect the outcome to be very different. Finally, we say that the similarity of the peers’ roles, may be important \( (S_i (\beta', \beta)) \).

For example, in some cases, peers with similar roles may behave similarly: all mathematicians provide their proofs in a deductive manner. Additionally, if one is interested in knowing how \( \beta \) alone has performed in the past, then one should set the condition \( S_i (\beta', \beta) = 1 \).

We then calculate the probability of having the outcome \( P''_\beta \) observed (i.e. \( \beta \) executing \( P'' \)) based on the relevance of the experience \( \mu \):

\[
p^\circ_e (X_\beta = P''_\beta | P_\beta) = p^\circ_e - 1 (X_\beta = P''_\beta | P_\beta) + |1 - p^\circ_e - 1 (X_\beta = P''_\beta | P_\beta)| \cdot \epsilon_e \cdot R^\circ_e \tag{29}
\]

Note that if \( P''_\beta \) was not already in the support of the distribution \( P^\circ_e (X_\beta | P_\beta) \), then that would be because the probability \( p^\circ_e (X_\beta = P''_\beta | P_\beta) = 0 \). In such a case, the equation above will result in introducing \( P''_\beta \) as a new element in the support of the distribution \( P^\circ_e (X_\beta | P_\beta) \). Also note that initially, the support is set to \( \{P_\beta, \text{other}\} \), and it increases incrementally based on new experiences.

\[
p^\circ_e (X_\beta = P''_\beta | P_\beta) \text{ essentially specifies the probability at a given point } X_\beta = P''_\beta \text{ of the probability distribution } P^\circ_e (X_\beta | P_\beta). \text{ The probability distribution } P^\circ_e (X_\beta | P_\beta) \text{ is then calculated accordingly:}
\]

\[
P^\circ_e (X_\beta | P_\beta) = \text{mre} (p^\circ_e (X_\beta = P''_\beta | P_\beta), P^\circ_e (X_\beta | P_\beta)) \tag{30}
\]

where \text{mre} is a function that modifies the previous probability distribution (in this case \( P^\circ_e (X_\beta | P_\beta) \)) in such a way that the new distribution is a distribution with minimum relative entropy and it satisfies the constraint imposed by the new probability \( p^\circ_e (X_\beta = P''_\beta | P_\beta) \).
Finally, we remind the reader that the information provided by the probability distribution \( P_{\alpha}^{\prime}(X | \beta) \), like any other piece of information, will all be subject to decay towards a predefined decay limit distribution.

### 4.2.3 Probability of Realising a Goal after Enacting a Given Plan

For simplification, we use the notation \( P_{\alpha}^{\prime}(X | \beta) \) to describe the probability distribution of observing goal \( X \) at time \( t_n \) given that \( P_{\beta} \) has been observed. In other words, \( P_{\alpha}^{\prime}(X | \beta) = \mathbb{P}(\text{Observe}(\alpha, X) | \text{Observe}(\gamma, P_{\beta})) \).

Again, \( P_{\alpha}^{\prime}(X | \beta) \) is calculated incrementally. At time \( t_0 \), and with the lack of any experiences, we say the initial value of such a probability distribution (\( P_{\alpha}^{\prime}(X | \beta) \)) should be domain dependent. Alternatively, for domain independent values, one can choose \( P_{\alpha}^{\prime}(X | \beta) = \mathbb{I} \), the uniform distribution, to describe ignorance. As new experiences are encountered, this probability starts to get reshaped by incorporating the information learned from these new experiences.

We say, given a new experience \( \mu = \langle \omega -, \text{Observe}(\gamma', P_{\beta}')_n, \text{Observe}(\alpha', G')_n \rangle \), we calculate its relevance to \( P_{\alpha}^{\prime}(X | \beta) \) as follows:

\[
R_{\mu}^{P_{\beta}} = \frac{\zeta_m \cdot S_{\mu}(P', P) + \zeta_o \cdot S_{\mu}(\beta', \beta)}{\zeta_m + \zeta_o}
\]  

(31)

where \( \zeta_m \) and \( \zeta_o \) are parameters that help specify the impact of (or weight given to) the similarity measures. Note that unlike Equations 25 and 28, Equation 31 uses the similarity between plans as opposed to plan empowerment. This is because peers’ capabilities are no longer an issue when observing whether certain actions results in certain goals. The actions have already been executed, and the capability of executing actions is not longer relevant. For the same reason, and unlike Equations 25 and 28, Equation 31 does not make use of similarity of peers’ capabilities. However, the similarity between peers is still crucial. That is because certain goals cannot be achieved unless the plans have been executed by specific pears (or, in most cases, roles). For example, if an experienced piano player practices one hour every week over a period of three months, then he will be playing on time by the end of the three months period. However, if a novice piano student practices one hour every week over a period of three months, then he will most probably not be playing on time by the end of the three months period, as novice students require much more practice than experienced players.

We then calculate the probability of having the outcome \( G' \) observed (i.e. goal \( G \) fulfilled) based on the relevance of the experience \( \mu \):

\[
p_{\alpha}^{\prime}(X=G' | P_{\beta}) = p_{\alpha}^{n-1}(X=G' | P_{\beta}) + (1 - p_{\alpha}^{n-1}(X=G' | P_{\beta})) \cdot \varepsilon_o \cdot R_{\mu}^{P_{\beta}}
\]  

(32)

Note that if \( G' \) was not already in the support of the distribution \( P_{\alpha}^{\prime}(X | \beta) \), then that would be because the probability \( p_{\alpha}^{\prime}(X=G' | \beta) = 0 \). In such a case, the equation above will result in introducing \( G' \) as a new element in the support of the distribution \( P_{\alpha}^{\prime}(X | \beta) \). Also note that initially, the support is set to \( \{ G, \text{other} \} \), and it increases incrementally based on new experiences.

\( p_{\alpha}^{\prime}(X=G' | \beta) \) essentially specifies the probability at a given point \( X=G' \) of the probability distribution \( P_{\alpha}^{\prime}(X | \beta) \). The probability distribution \( P_{\alpha}^{\prime}(X | \beta) \) is then calculated accordingly:

\[
P_{\alpha}(X | \beta) = mre(p_{\alpha}^{\prime}(X=G' | \beta), p_{\alpha}^{n-1}(X | \beta))
\]  

(33)
where \( mre \) is a function that modifies the previous probability distribution (in this case \( \mathbb{P}^{t_n-1}(X \mid P_\beta) \)) in such a way that the new distribution is a distribution with minimum relative entropy and it satisfies the constraint imposed by the new probability \( p_0^t(X=G' \mid P_\beta) \).

Finally, we remind the reader that the information provided by the probability distribution \( \mathbb{P}^t(X \mid P_\beta) \), like any other piece of information, will all be subject to decay towards a predefined decay limit distribution.

### 4.3 Trust Computation

#### 4.3.1 Trust Measures

After calculating \( \mathbb{P}(\text{Observe}(\alpha, X) \mid \text{Commit}(\rho, [P_\eta]G)) \) at a given time \( t_n \), which we simply refer to as \( \mathbb{P}^t(X \mid [P_\eta]G) \), 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?

In what follows, we define three different trust equations that can be implemented, or chosen, depending on the particular context or interest. In the first, the expected performance, which we refer to as the expected enactment, is compared to an ideal performance (or ideal enactment), which simply specifies what the ideal outcome would be. In the second, the expected performance is compared to what other outcomes (or enactments) are preferred. In the third, the focus is on the certainty of the new expectation.

1. **Ideal enactments.** Consider a distribution of enactments that represent \( \alpha \)'s “ideal” in the sense that it is the best that \( \alpha \) could reasonably expect to happen: \( \mathbb{P}^t_I(X \mid [P_\eta]G) \). For example, even if \( \beta \) has proposed a plan for fulfilling the goal of ‘improving performance by 75%’, \( \alpha \)'s ideal outcome would be ‘improving performance by 100%’. Trust is then computed by measuring the relative entropy between this ideal distribution, \( \mathbb{P}^t_I(X \mid [P_\eta]G) \), and the distribution of expected enactments, \( \mathbb{P}^t(X \mid [P_\eta]G) \):

\[
\text{trust}^t_\alpha(\alpha, \rho, [P_\eta]G) = 1 - \sum_{X_i} p^t_I(X=X_i \mid [P_\eta]G) \cdot \log \frac{p^t_I(X=X_i \mid [P_\eta]G)}{p^t(X=X_i \mid [P_\eta]G)}
\]

where “1” is an arbitrarily chosen constant being the maximum value that this measure may have.

It may be questioned whether it is fair to allow peers to set ideal enactments that are different from the goal \( G \) committed to, resulting in one having its trust measure be less than perfect even when their suggested plans for achieving goals always succeed. This may easily be solved by restricting peers from defining their own ideal enactments and assuming an ideal enactment to describe the case when the peer in question simply fulfils its promises. Nevertheless, we say it is also possible (if needed) to permit peers to describe their own ideal enactments because it may also be argued that one can always do better than what they promise, and there should be means for capturing that.

Last, but not least, we note that although we use the equation of relative entropy (Equation 34) to measures the distance between two probability distributions, we note that alternative methods for calculating this distance may also be considered. For instance, Equation 34 may use the earth mover’s distance instead:

\[
\text{trust}^t_\alpha(\alpha, \rho, [P_\eta]G) = 1 - \text{emd}(\mathbb{P}^t_I(X \mid [P_\eta]G), \mathbb{P}^t(X \mid [P_\eta]G))
\]
where the earth mover’s distance (specified through the function *emd*) is a measure of the distance between two probability distributions.

2. **Preferred enactments.** This measures the extent to which the enactment $X_i$ is preferable to the commitment $G$. It requires $\alpha$ to specify its preferences with respect to enactments via $X_i \succ_\alpha G$, which described how much $\alpha$ prefers $X_i$ to $G$ (where the range of $\succ_\alpha$ is $\mathbb{N}$). The set of preferences ($\{X_1 \succ_\alpha G, \ldots, X_n \succ_\alpha G\}$) is then transformed into a normalised probabilistic measure, defined as $\mathbb{P}^\alpha_{\succ_\alpha}(X, G)$. As such, the final trust measure is then simply an aggregation of the various expected outcomes ($p^\alpha(X=X_i \mid [P_\eta]G)$), where the weight of each is decided by the peer’s predefined preferences. This is expressed accordingly:

$$trust^\alpha(\alpha, \rho, [P_\eta]G) = \sum_{X_i} \mathbb{P}^\alpha_{\succ_\alpha}(X=X_i, G) \cdot p^\alpha(X=X_i \mid [P_\eta]G)$$  \hspace{1cm}  (35)

3. **Certainty in enactment.** This measures the consistency in expected acceptable enactments, or in other words, the lack of expected uncertainty in those possible enactments that are better than the original goal. We use entropy to measure uncertainty, and we note that the minimal the uncertainty (or the minimal the entropy) then the maximal trust is. As such, we say let $X(G, \sigma) = \{X_i \mid \mathbb{P}^\alpha_{\succ_\alpha}(X=X_i, G) \geq \sigma\}$ describe the set of acceptable expected enactments with respect to $G$ (where $\sigma$ identifies what is considered “acceptable”), and:

$$trust^\alpha(\alpha, \rho, [P_\eta]G) = 1 + \frac{1}{B^\sigma} \sum_{X_i \in X(G, \sigma)} \mathbb{P}^\alpha_{\succ_\alpha}(X=X_i \mid [P_\eta]G) \cdot \log \mathbb{P}^\alpha_{\succ_\alpha}(X=X_i \mid [P_\eta]G)$$  \hspace{1cm}  (36)

where $\mathbb{P}^\alpha_{\succ_\alpha}(X \mid [P_\eta]G)$ is the normalisation of $\mathbb{P}^\alpha_{\succ_\alpha}(X \mid [P_\eta]G)$ for $X_i \in X(G, \sigma)$, and:

$$B^\sigma = \begin{cases} 1, & \text{if } |X(G, \sigma)| = 1 \\ \log |X(G, \sigma)|, & \text{otherwise} \end{cases}$$

4.3.2 **General Trust Measures**

In the previous section, we have illustrated how trust measures may be computed for specific commitments by calculating $\mathbb{P}(\text{Observe}(\alpha, X) \mid \text{Commit}(\rho, [P_\eta]G))$ for a particular $\rho$, $\eta$, $G$, and $P$. If we are interested in trust measures for more general cases, then the parameters of Equations 25, 28, and 31 need to be set accordingly.

For instance, if we are comparing past plans in general, regardless of the advisers that suggested them, then we set $\zeta_r = 0$, which essentially states that all advisers with previous plans are considered relevant, and not specific advisers only. Similarly, when we are comparing $\rho'$’s past plans without considering who these plans were designed for, then we set $\zeta_r = 0$, which essentially states that all peers who have executed plans in the past are considered relevant, and not specific peers only. And so, when we are comparing $\rho'$’s past plans without considering the goal of the plan, then we set $\zeta_r = 0$; and when we are comparing $\rho'$’s past recommendations without considering the specific recommended plan, then we set $\zeta_p = 0$.

The same reasoning is followed to compute even more general trust measures by combining the generalisations above, such as having $\zeta_r = 0 \land \zeta_p = 0$. And so on.
4.4 Trust Algorithm

We now give Algorithm 2 as an example of a default trust algorithm. This algorithm uses the "preferred enactments" trust equation, assumes the decay to follow Equation 22, and assumes the decay limit distributions are equiprobable distributions (i.e., they are equivalent to the uniform distribution).

Other algorithms may be similarly defined. For instance, a generic version of the algorithm where functions like $emd$, $oNet$, or $mre$ are parameters is also straightforward.

Trust is calculated on demand following Algorithm 2. Other implementations that precompute probability distributions are possible but not considered here. However, we note that although trust is computed on demand, we still use memorising techniques to help improving the efficiency of the algorithm. We do this by allowing the latest probability distributions to be saved by replacing the older computation. This way, when a probability distribution needs to be calculated when it has already been calculated in the past, the past distribution is modified by considering only the new experiences.

The algorithm uses parameters $\xi_c$, $\xi_e$, and $\xi_o$ to specify the thresholds on when to consider an experience relevant. By fixing the values of these parameters to high values, we can have 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.

5 TrustIt 1: Trust on Information Providers

What is an information provider? An information provider is a peer who provides general information on a given issue. The information provided is defined as a statement that the information provider is conveying to the community as true. For example, after a student plays a trumpet solo, a teacher may say: 'Chet Baker used similar chord progressions'.

There is a crucial need in pinpointing valuable information in online communities. This is because online discussions are becoming more and more loaded with noise, making it harder to find the valuable comments. We categorise the noise in discussions, or what we refer to as "valueless" information, accordingly:

- **Unreliable Information.** Unreliable information is information whose truth value is questionable, if not false. We distinguish between two types of unreliable information.
  - **Bias.** Bias can take a variety of forms, such as prejudice, racism, sexism, etc. The truth value of biased information is questionable, deeming the information as unreliable.
  - **Lie.** A lie is a statement that is false.

- **Irrelevant Information and Spam.** Irrelevant information is information that is outside the context it pretends to address. For example, in a discussion on one’s music performance, the comment ‘It is raining today’ will simply be considered irrelevant. Spam is a type of irrelevant information, usually conveyed for advertising purposes. For example, when asking peers to comment on one’s music performance, one peer may simply leave links to his own music.
Algorithm 2 \( \text{trust}^t_h(\alpha, p, [P_h][G]) \)

Require: \( G = \{G, \text{other}\} \) \hspace{1em} ▷ The initial support for \( P^\alpha_h \) is \( \{G, \text{other}\} \)

Require: \( L = \{P_h, \text{other}\} \) \hspace{1em} ▷ The initial support for \( P^\alpha_c \) and \( P^\alpha_o \) is \( \{P_h, \text{other}\} \)

Require: \( H_\alpha \) \hspace{1em} ▷ This is \( \alpha \)'s history of past experiences

Require: \( H^\delta_c \) \hspace{1em} ▷ This is \( \alpha \)'s history of past experiences with a strict total order over the the \( n \)th element’s time-stamps, and the time-stamp of the \( n \)th element of each experience is greater than \( t_i \); for example, \( H^\delta_c = \{\langle \text{Commit}(\ldots), \ldots \rangle | \langle \text{Commit}(\ldots), \ldots \rangle \in H_\alpha \land t_j > t_i \} \)

Require: \( P_\alpha \) \hspace{1em} ▷ This is \( \alpha \)'s database of previously calculated probability distributions

Require: \( T_h, T_g, \) and \( T_R \) \hspace{1em} ▷ These describe the taxonomies of actions (\( T_h \)), goals (\( T_g \)), and user roles (\( T_R \))

Require: \( S_i, S_e, E_p, \) and \( S_c \) \hspace{1em} ▷ These describe the similarity measures of Equations \([16][15][18]\) and \([19]\)

Require: \( M \) \hspace{1em} ▷ This describes a meronomy \( M \) of actions

Require: \( M_M \) \hspace{1em} ▷ This specifies the set of empowerment measures between actions of \( M \)

Require: \( \text{omNet}(M, T, \phi, \phi') \) \hspace{1em} ▷ This calculates the empowerment measure of \( \phi \) on \( \phi' \) (Alg. \([1]\))

Require: \( \text{emd}(P, P') \) \hspace{1em} ▷ This is the earth mover’s distance that calculates the distance between two probability distributions \( P \) and \( P' \)

Require: \( T = \{T(1) \mapsto 0, \ldots, T(n-1) \mapsto 0, T(n) \mapsto 1\} \) \hspace{1em} ▷ This is a probability distribution describing the highest opinion possible, where we choose \( n = 5 \)

Require: \( \xi_c, \xi_p, \xi_e, \xi_z, \xi_c, \xi_z, \xi_o, \xi_s, \xi_o, \) and \( \xi_0 \) \hspace{1em} ▷ These specify the weights given to the different similarity measures

Require: \( \xi_c, \xi_p, \) and \( \xi_o \) \hspace{1em} ▷ These describe the thresholds of when to consider an experience relevant to \( P_c, P_p, \) and \( P_o \), respectively

Require: \( \epsilon_c, \epsilon_p, \) and \( \epsilon_o \) \hspace{1em} ▷ This describes the percentage of increase in the probabilities \( p_c, p_p, \) and \( p_o \), respectively

Require: \( \text{mre}(p(X=X_i), P) \) \hspace{1em} ▷ This modifies the probability distribution \( P \) in such a way that the new distribution is a distribution with minimum relative entropy and it satisfies the constraint imposed by the new probability \( p(X=X_i) \).

Require: \( D_x = D_c = D_o = \emptyset = \{F(X_1) \mapsto \frac{1}{n}, \ldots, F(X_n) \mapsto \frac{1}{n}\} \) \hspace{1em} ▷ These are the decay limit distributions, which are set to the uniform distribution \( P \), where \( n \) is the size of the support of the probability distribution \( P \) that needs to decay

Require: \( \Lambda(D_x, P^{n-1}) = \nu^{\Delta_n} \cdot P^{n-1} + (1 - \nu^{\Delta_n})D_x \) \hspace{1em} ▷ This decays the distribution \( P^{n-1} \) towards \( D_x \)

Require: \( \nu \) \hspace{1em} ▷ This is the maximum possible decay rate

Require: \( \omega, t_{\max}, \) and \( \Delta_n = \{0, \text{if } t_n - t_{n-1} < \omega; 1 + (t_n - t_{n-1})/t_{\max}, \text{otherwise}\} \) \hspace{1em} ▷ \( \Delta_n \) establishes a minimum grace period (\( \omega \)) during which the information does not decay, and that once reached the information starts decaying; and \( t_{\max} \) describes the pace of decay

Require: \( \nu \) and \( \Delta_n \) \hspace{1em} ▷ This is the grace period of decay

Require: \( \mathbb{P}_t(X | [P_h][G]) = \{1, \text{if } X = G; 0, \text{otherwise}\} \) \hspace{1em} ▷ This is the ideal enactment of realising goal \( G \)

Require: \( \text{calcTrust}(\mathbb{P}^t_n(X | [P_h][G]), \mathbb{P}_t(X | [P_h][G])) = 1 - \sum_{X_i} p_t(X=X_i | [P_h][G]) \cdot \log \frac{p_t(X=X_i | [P_h][G])}{p_H^t(X=X_i | [P_h][G])} \) \hspace{1em} ▷ This calculates the trust measure from the probability distribution \( P^t_n(X | [P_h][G]) \) based on the ideal enactment method of Section \([4,3]\) Equation \([34]\)

if \( M_h^c \neq M^{h-n} \) then \hspace{1em} ▷ \( M_M \) is calculated every time meronomy \( M \) is modified
\qquad for all \( \phi \in M \) do
\qquad \qquad \\( E(\phi, \phi') = |\text{emd}(T, \text{omNet}(M, T, \phi, \phi'))| \)
\qquad \qquad \( M_M = M_M + E(\phi, \phi') \)
\qquad end for
\end if
\[ \begin{align*}
&\text{if } P_{c}^{n}(X | P_{\eta}) \in P_{\alpha} \text{ then} \quad \triangleright \text{Check if the probability distribution has been previously calculated} \\
&H_{c}^{n} = H_{a}^{n} \quad \triangleright \text{Consider only the new elements of the history of experiences} \\
&\text{else} \\
&\quad \text{\( P_{c}^{n-1}(X | P_{\eta}) = \mathcal{F} \quad \triangleright \text{Set the initial distribution, with support } \mathcal{L}, \text{ to the uniform distribution } \mathcal{F} \)} \\
&H_{c}^{n} = H_{a}^{n} \quad \triangleright \text{Consider the entire history of experiences} \\
&\text{end if} \\
&\text{for all } \langle \text{Commit}(\beta', [P_{\eta}']) | [G']_{\alpha}, \text{Commit}(\beta, [P_{\eta}'])_{\alpha}, \ldots \rangle \in H_{c}^{n} \text{ do} \quad \triangleright \text{Calculate } P_{c}^{n}(X | P_{\eta}) \\
&\quad \text{Note that the oldest experience is chosen first from the totally ordered set } H_{c}^{n} \\
&\quad R_{\mu}^{\text{ot}} = S_{c}(G', [G'])_{\text{ot}} \cdot E_{p}(P', [P])_{\text{ot}} \cdot S_{c}(P_{\eta}', P_{\eta})_{\text{ot}} \cdot \left( P_{c}(P_{\eta}', P_{\eta})_{\text{ot}} \cdot S_{c}(\eta', \eta)_{\text{ot}} \right) \\
&\quad \text{if } R_{\mu}^{\text{ot}} \geq \xi_{\text{ot}} \text{ then} \\
&\quad \quad \mathcal{L} = \mathcal{L} \ominus P_{\beta} \\
&\quad \quad P_{c}^{\mu-1-\mu}(X_{\beta} | P_{\eta}) = \Lambda_{\mathcal{L}} \cdot P_{c}^{n-1}(X_{\beta} | P_{\eta}) \\
&\quad \quad p_{c}^{\mu}(X_{\beta} = P_{\beta} | P_{\eta}) = p_{c}^{\mu-1-\mu}(X_{\beta} = P_{\beta} | P_{\eta}) + (1 - p_{c}^{\mu-1-\mu}(X_{\beta} = P_{\beta} | P_{\eta})) \cdot \epsilon_{c} \cdot R_{\mu}^{\text{ot}} \\
&\quad \quad P_{c}^{\mu}(X_{\beta} | P_{\eta}) = \text{mre}(p_{c}^{\mu}(X_{\beta} = P_{\beta} | P_{\eta}), P_{c}^{n-1-\mu}(X_{\beta} | P_{\eta})) \\
&\quad \quad P_{\mathcal{L}} = (\mathcal{L} \backslash \{ P_{c}^{\mu}(X_{\beta} | P_{\eta}) \}) \oplus P_{c}^{\mu}(X_{\beta} | P_{\eta}) \\
&\quad \text{end if} \\
&\quad \text{end for} \\
&\quad \text{for } P_{\beta} \in \mathcal{L} \backslash \{ \text{other} \} \text{ do} \\
&\quad \quad \text{if } P_{c}^{n}(X_{\beta} | P_{\beta}) \in P_{\alpha} \text{ then} \quad \triangleright \text{If the probability distribution has been previously calculated, then only consider the new elements of the history of experiences} \\
&\quad \quad \quad H_{c}^{n} = H_{a}^{n} \quad \triangleright \text{The initial support is } \{ P_{\beta}, \text{other} \} \\
&\quad \quad \quad \text{else} \\
&\quad \quad \quad \quad \text{\( P_{c}^{n-1}(X_{\beta} | P_{\beta}) = \{ \mathcal{F} \rightarrow 0.5, \mathcal{F}(\text{other}) \rightarrow 0.5 \} \) \quad \triangleright \text{The initial support is } \{ P_{\beta}, \text{other} \} \\
&\quad \quad \quad \quad \text{The entire history of experiences is used} \\
&\quad \quad \quad \quad \text{end if} \\
&\quad \quad \text{for all } \langle \text{Observe}(\gamma', [P_{\beta}'])_{\alpha}, \ldots \rangle \in H_{c}^{n} \text{ do} \quad \triangleright \text{Calculate } P_{c}^{n}(X_{\beta} | P_{\beta}) \\
&\quad \quad \quad \text{Note that the oldest experience is chosen first from the totally ordered set } H_{c}^{n} \\
&\quad \quad \quad R_{\mu}^{\text{ot}} = E_{p}(P', [P])_{\text{ot}} \cdot \left( P_{c}(P_{\eta}', P_{\eta})_{\text{ot}} \cdot S_{c}(\beta', \beta)_{\text{ot}} \right) \\
&\quad \quad \quad \text{if } R_{\mu}^{\text{ot}} \geq \xi_{\text{ot}} \text{ then} \\
&\quad \quad \quad \quad \mathcal{L} = \mathcal{L} \ominus P_{\beta} \\
&\quad \quad \quad \quad P_{c}^{\mu-1-\mu}(X_{\beta} | P_{\beta}) = \Lambda_{\mathcal{L}} \cdot P_{c}^{n-1}(X_{\beta} | P_{\beta}) \\
&\quad \quad \quad \quad p_{c}^{\mu}(X_{\beta} = P_{\beta} | P_{\beta}) = p_{c}^{\mu-1-\mu}(X_{\beta} = P_{\beta} | P_{\beta}) + (1 - p_{c}^{\mu-1-\mu}(X_{\beta} = P_{\beta} | P_{\beta})) \cdot \epsilon_{c} \cdot R_{\mu}^{\text{ot}} \\
&\quad \quad \quad \quad P_{c}^{\mu}(X_{\beta} | P_{\beta}) = \text{mre}(p_{c}^{\mu}(X_{\beta} = P_{\beta} | P_{\beta}), P_{c}^{n-1-\mu}(X_{\beta} | P_{\beta})) \\
&\quad \quad \quad \quad P_{\mathcal{L}} = (\mathcal{L} \backslash \{ P_{c}^{\mu}(X_{\beta} | P_{\beta}) \}) \oplus P_{c}^{\mu}(X_{\beta} | P_{\beta}) \\
&\quad \quad \quad \text{end if} \\
&\quad \quad \text{end for} \\
&\quad \text{end for} \\
&\quad \text{for } P_{\beta} \in \mathcal{L} \backslash \{ \text{other} \} \text{ do} \\
&\quad \quad \text{if } P_{c}^{n}(X | P_{\beta}) \in P_{\alpha} \text{ then} \quad \triangleright \text{If the probability distribution has been previously calculated, then only consider the new elements of the history of experiences} \\
&\quad \quad \quad H_{c}^{n} = H_{a}^{n} \quad \triangleright \text{The initial support is } \{ G, \text{other} \} \\
&\quad \quad \quad \text{else} \\
&\quad \quad \quad \quad \text{\( P_{c}^{n-1}(X | P_{\beta}) = \{ \mathcal{F} \rightarrow 0.5, \mathcal{F}(\text{other}) \rightarrow 0.5 \} \) \quad \triangleright \text{The initial support is } \{ G, \text{other} \} \\
&\quad \quad \quad \quad \text{end if} \\
&\quad \quad \text{end if} \\
&\text{end for} \\
&\text{end for} \\
&\text{end for} \\
&\text{end for}
\end{align*} \]

D3.3 Trust and Reputation in Online Social Learning Communities
D3.3 Trust and Reputation

FP7-ICT-2011- 8 Collaborative project 318770 PRAISE

for all \( \langle \cdot, \cdot \rangle \) Observe\( (\gamma', P^o_{\beta}) \rangle_m, \) Observe\( (\alpha', G') \rangle_m \) do

\[ R^o_{\mu_p} = S_{\gamma}(P', P)^{\gamma_m} \cdot S_{\gamma}(\beta', \beta)^{\gamma_m} \]

\[ \text{if } R^o_{\mu_p} \geq \xi_o \text{ then } \]

\[ \begin{align*}
\mathcal{P}^{o-1}_{\rho}(X | P_{\beta}) &= \lambda(\mathcal{D}_{\rho}, \mathcal{P}^{o-1}_{\rho}(X | P_{\beta})) \\
p^o_{\rho}(X = G' | P_{\beta}) &= p^o_{\rho-1}(X = G' | P_{\beta}) + (1 - p^o_{\rho-1}(X = G' | P_{\beta})) \cdot \epsilon_{\alpha} \cdot R^o_{\mu_p} \\
\mathcal{P}^{o}_{\alpha} &= (\mathcal{P} \backslash \mathcal{P}^{o}(X | P_{\beta})) \oplus \mathcal{P}^{o}_{\rho}(X | P_{\beta}) \\
\end{align*} \]

\[ \text{end if} \]

end for

end for

\[ \mathcal{T} \]

\[ \text{Build the main probability distribution } \mathcal{P}^{o}(X | [P_{\eta}]G) \]

\[ p(G' | [P_{\eta}]G) = 0 \]

\[ \text{for } P^o_{\beta} \in \mathcal{L} \text{ do} \]

\[ \text{if } \mathcal{P}(X | P^o_{\beta}) \notin \mathcal{P}_{\alpha} \text{ then} \]

\[ p(G' | P^o_{\beta}) = 0 \]

\[ \text{end if} \]

end for

end for

\[ \text{Calculate the final trust measure} \]

\[ \text{trust}(\alpha, \rho, [P_{\eta}]G) = \text{calcTrust}(\mathcal{P}^{o}(X | [P_{\eta}]G), \mathcal{P}_1(X | [P_{\eta}]G)) \]

\[ \text{return } \text{trust}(\alpha, \rho, [P_{\eta}]G) \]

Redundant Information. Redundant information is usually common in online discussions as it is the result of presenting information in a temporal manner, such as presenting the latest post or comment first. As such, many peers tend to repeat the same information over and over again, just to ensure their information is presented first. Of course, the information itself may or may not be valuable, but what is clear is that redundancy does not actually bring in new information. Furthermore, users tend to distrust peers who post redundant information.

Insignificant Information. Insignificant information is information that is neither categorised as unreliable nor as noise, yet it fails to leave a significant impact on the community. This information is conveyed to. For example, if a student asks for feedback on its performance, and one replies with ‘Chet Baker used similar chord progressions’, but no one else interacts with such information, then we categorise such information as insignificant.
We note that insignificant information is the least problematic, since what is considered insignificant one day may change and become highly significant another day. Whereas the reliability and relevance aspects of information seldom change with time (we say ‘seldom’ because circumstances may arise, for example, where what is considered true may change and be considered a lie, or vice versa; recall that the statement ‘the earth is flat’ was once considered true).

Valuable information is then the opposite of valueless information. It is reliable, relevant, non-redundant information with significant impact. The goal of the trust model is to pinpoint trustworthy information providers whose information is considered valuable. But how can we assess the reliability, relevance, redundancy, or significance of information?

First, we define the information that \( \beta \) provides at time \( t_n \) on an issue \( \phi \) as \( i^\beta_n(\phi) \), and we say information is specified as free text. We note that for simplification, we sometimes drop off the time \( t_n \), and simply refer to a piece of information as \( i^\beta(\phi) \) when the time is not relevant.

We then say the trust on information \( i^\beta(\phi) \) becomes an aggregation of the measures describing the above four aspects:

\[
\text{trust}^\phi(\alpha, \beta, i^\beta(\phi)) = \frac{\lambda_b \cdot \text{relb}_\alpha(i^\beta_n(\phi)) + \lambda_r \cdot \text{relv}_\alpha(i^\beta_n(\phi)) + \lambda_d \cdot \text{red}_\alpha(i^\beta_n(\phi)) + \lambda_s \cdot \text{sig}_\alpha(i^\beta_n(\phi))}{\lambda_b + \lambda_r + \lambda_d + \lambda_s}
\]

(37)

where \( \text{relb}_\alpha(i^\beta_n(\phi)) \) describes the reliability of \( \beta \)’s information in the context of \( \phi \), \( \text{relv}_\alpha(i^\beta_n(\phi)) \) describes the relevance of \( \beta \)’s past information in the context of \( \phi \), \( \text{red}_\alpha(i^\beta_n(\phi)) \) describes the redundancy of \( \beta \)’s past information in the context of \( \phi \), and \( \text{sig}_\alpha(i^\beta_n(\phi)) \) describes the significance of \( \beta \)’s past information in the context of \( \phi \). The four measures are then aggregated in Equation 37 with the option of providing each of those measures with a different weight: \( \lambda_b \), \( \lambda_r \), \( \lambda_d \), and \( \lambda_s \).

Note that we use the weighted arithmetic average in Equation 37 to aggregate these four aspects as opposed to conjunctive operators such as the product (see Figure 3 and its discussion for a brief classification of these operators) since we believe conjunctive operators could be too extreme. For instance, if one’s reliability, relevance, redundancy and significance measures are all very high and equal to 0.9, then the product will result in the final trust measure decreasing dramatically to 0.6561!

Similar to the previous trust models of this paper, we argue that the context of the information \( \phi \) is crucial. For instance, we do not assess whether \( \beta \)’s information is reliable in general, but whether \( \beta \)’s information in the context of \( \phi \) is reliable. Similarly, relevance, redundancy, and significance of information is also measured in the context of the information \( \phi \). In what follows, we propose methods for calculating the measures for each of the above four aspects of valuable information.

### 5.1 Reliability Measure.

We say reliability measures will need to rely on the peer opinions, as there is no automated way to assess the truth value of statements in general. We assume that the more positive opinions a piece of information receives, then the more reliable it is. The prediction of the reliability of \( \beta \)’s new piece of information \( i^\beta(\phi) \) is then based on aggregating the reliability of \( \beta \)’s past information on issues similar to \( \phi \):

\[
\text{relb}_\alpha(i^\beta_n(\phi)) = \frac{\sum_{i^\beta_n(\phi') \in V^\phi_n} \text{reliability}_\alpha(i^\beta_n(\phi'))}{|V^\phi_n|}
\]

(38)
where $V_{\beta,\phi}^n = \{i_\beta^n(\phi') \mid \phi' \sim_v \phi \land t_i \leq t_n\}$ describes the set of information provided by $\beta$ by time $t_n$ on issues similar to $\phi$ (where $\phi' \sim_v \phi$ describes that $\phi'$ is considered semantically similar to $\phi$ — we say if an issue is defined by a set of keywords, then $\phi' \sim_v \phi \equiv S_v(\phi, \phi') \geq v$, where $S_v(\phi, \phi')$ describes the semantic similarity between sets of terms, as it is has been defined by Equation 15), and $\text{reliability}_{\alpha}^{\text{relv}}(i_\beta^n(\phi'))$ describes the reliability of the information $i_\beta^n(\phi')$ based on peer opinions obtained by time $t_n$.

The reliability of $i_\beta^n(\phi)$ is defined as an aggregation of the opinions that this information has received, and the weight given to each of those opinions is the trust measure on the opinion holder that provided these opinions, which has been defined earlier by Section 3. This is specified accordingly:

$$
\text{reliability}_{\alpha}^{\text{relv}}(i_\beta^n(\phi)) = \frac{\sum_{o_j(i_\beta^n(\phi)) \in Y_{\beta,\phi}^n} \text{trust}^\text{tn}(\alpha, \gamma, o_\gamma(i_\beta^n(\phi))) \cdot o_j^\text{tn}(i_\beta^n(\phi))}{\sum_{o_j(i_\beta^n(\phi)) \in Y_{\beta,\phi}^n} \text{trust}^\text{tn}(\alpha, \gamma, o_\gamma(i_\beta^n(\phi)))}
$$

(39)

where $Y_{\beta,\phi}^n = \{o_j(i_\beta^n(\phi)) \mid o_j(i_\beta^n(\phi)) \in \mathcal{O} \land t_i \leq t_n\}$ describes the subset of opinions (where $\mathcal{O}$ describes the set of all opinions) that have been made by community members by time $t_n$ about the piece of information $i_\beta^n(\phi)$.

### 5.2 Relevance Measure.

We say relevance may rely on peers reporting irrelevant information. The prediction of the relevance of $\beta$’s new piece of information $i_\beta^n(\phi)$ is then based on aggregating the relevance of $\beta$’s past information on issues similar to $\phi$:

$$
\text{relv}_{\alpha}^{\text{relv}}(i_\beta^n(\phi)) = \frac{\sum_{\phi' \in Y_{\beta,\phi}^n} \text{relevance}_{\alpha}^{\text{relv}}(i_\beta^n(\phi'))}{|Y_{\beta,\phi}^n|}
$$

(40)

where $Y_{\beta,\phi}^n$, as illustrated earlier, describes the set of information provided by $\beta$ by time $t_n$ on issues similar to $\phi$, and $\text{relevance}_{\alpha}^{\text{relv}}(i_\beta^n(\phi'))$ describes the relevance of the information $i_\beta^n(\phi')$ based on the reporting of irrelevant information done by the peers by time $t_n$.

Reporting a piece of information may conceptually be viewed as the peer providing its opinion on the relevance of this information. Accordingly, relevance needs to consider how much trusted is a peer $\gamma$ in reporting a piece of information $i_\beta^n(\phi)$ (specified as $\text{trust}^\text{tn}(\alpha, \gamma, o_\gamma(\text{report}(i_\beta^n(\phi))))$, which has been defined earlier by Section 3). The relevance measure is then an aggregation of the trust measures of the peers that have reported the information in question:

$$
\text{relevance}_{\alpha}^{\text{relv}}(i_\beta^n(\phi)) = 1 - \frac{\sum_{\gamma \in U_{\beta,\phi}^n} \text{trust}^\text{tn}(\alpha, \gamma, o_\gamma(\text{report}(i_\beta^n(\phi))))}{|U_{\beta,\phi}^n|}
$$

(41)

where $\text{report}(\gamma, i_\beta^n(\phi))$ states that $\gamma$ has reported the information $i_\beta^n(\phi)$, and $U_{\beta,\phi}^n = \{\gamma \mid \text{report}(\gamma, i_\beta^n(\phi))\}$ describes the set of peers who have reported this information.
5.3 Redundancy Measure.

We say redundancy may be picked up automatically. The more redundant one’s posts are, the higher their redundancy measure. The prediction of the redundancy of $\beta$’s new piece of information $i_\beta(\phi)$ is then based on aggregating the redundancy of $\beta$’s past information on issues similar to $\phi$:

$$
red^v(i_\beta(\phi)) = \frac{\sum_{i_\beta(\phi') \in V^v_{\beta,\phi}} \text{redundancy}^v(i_\beta(\phi'))}{|V^v_{\beta,\phi}|} \quad (42)
$$

where $V^v_{\beta,\phi}$, as illustrated earlier, describes the set of information provided by $\beta$ by time $t_n$ on issues similar to $\phi$, and $\text{redundancy}^v(i_\beta(\phi'))$ describes whether the information $i_\beta(\phi')$ is redundant or not.

Note that $\text{redundancy}$ is a binary function whose range $\{0, 1\}$ specifies whether a specific piece of information is redundant or not. As such, the redundancy measure of Equation 42 describes the percentage of redundant past posts. As for the $\text{redundancy}$ function, it is defined accordingly:

$$
\text{redundancy}^v(i_\beta(\phi')) = \begin{cases} 
1 & \text{if } \exists \ i_\beta(\phi') \cdot t_i \neq t_j \land i_\beta(\phi') \approx_{\upsilon} i_\beta(\phi) \\
0 & \text{otherwise}
\end{cases} \quad (43)
$$

Equation 43 states that a piece of information $i_\beta(\phi')$ is considered redundant if there exists another piece of information $i_\beta(\phi)$ on the same issue $\phi$ that is semantically similar to $i_\beta(\phi')$: $i_\beta(\phi') \approx_{\upsilon} i_\beta(\phi)$. Note that we define redundancy as the repetition of the same information on the same issue $\phi$. However, if information is repeated for two different issues (say two different discussions), then the repetitive information is not considered redundant.

Recall that we assume information to be provided by members as free text. We argue that information is redundant not only if the set of words in the free text is semantically similar to the set of words in another existing free text (or piece of information), but if the two texts (whether composed of sentences or paragraphs) as a whole are semantically similar. As such, we do not rely on the $\sim_{\upsilon}$ operator for calculating the semantic similarity between sets of terms, and we define the redundancy operator $\approx_{\upsilon}$ which calculates the semantic similarity between texts. In its simplest form, one can say two texts are semantically similar if they are identical: $(i_\beta(\phi) \approx_{\upsilon} i_\beta(\phi)) \equiv (i_\beta(\phi) = i_\beta(\phi))$. However, for a more complex and realistic definition of semantic similarity of texts, we leave the definition of the $\approx_{\upsilon}$ operator, which is outside the scope of this paper, to the linguistic analysis research line.

5.4 Significance Measure.

We say the significance measure of a statement may rely on the level of interactions it stirs, such as the number of opinions it receives, the number of comments it incites, the number of times this information was shared, and so on. The prediction of the significance of $\beta$’s new piece of information $i_\beta(\phi)$ is then based on aggregating the significance of $\beta$’s past information on issues similar to $\phi$:

$$
sig^v(i_\beta(\phi)) = \frac{\sum_{i_\beta(\phi') \in V^v_{\beta,\phi}} \text{significance}^v(i_\beta(\phi'))}{|V^v_{\beta,\phi}|} \quad (44)
$$
where $V_{\beta, \phi}$, as illustrated earlier, describes the set of information provided by $\beta$ by time $t_n$ on issues similar to $\phi$, and $\text{significance}^{\phi_n}(i_{\beta}(\phi'))$ describes the significance of the information $i_{\beta}(\phi')$ based on the level of interaction it has stirred by time $t_n$.

The basic idea in defining the $\text{significance}$ measure is to have this measure increase with the number of interactions: the more interaction a piece of information receives, then the higher its significance. However, we note that the number of interactions may go up to infinity, whereas to be coherent with other measures, the range of this significance measure should be $[0, 1]$. As such, we say that the significance measure should increase from 0 to 1 as the number of interactions increases increases from 0 to $\infty$. Formally, we define the significance measure of a single piece of information $i_{\beta}(\phi)$ accordingly:

$$\text{significance}^{\phi_n}(i_{\beta}(\phi)) = 1 - e^{-\tau I_{t_n}(i_{\beta}(\phi))}$$  \hspace{1cm} (45)

where $e$ is Euler’s number, $I_{t_n}(i_{\beta}(\phi))$ describes the number of interactions that the information $\phi$ has stirred in the community by time $t_n$, and $\tau$ is a parameter used to control how the significance increases with the number of interactions. Figure 4 provides a visual illustration on how the significance measure increases for different values of $\tau$.

![Figure 4: The increase of the significance measure for different values of $\tau$](image)

5.5 Trust versus Reputation.

As illustrated earlier, our view of trust is that it describes one party’s assessment of another party’s trustworthiness, whereas reputation describes the group’s opinion of the party in question. The measure calculated by Equation 37 may seem at a first glance to describe a reputation measure. This is because it is based on four measures, two of which (redundancy and significance) are generic measures that do not describe a specific party’s view as they are based on group information. The reliability and relevance measures, however, rely on the reputation-based trust measure $\text{trust}^{\beta}(\alpha, \gamma, o_{\gamma}(\omega))$ (as illustrated in Section 3) this trust measure relies on the group’s opinion, yet
the group is defined by the peer assessing the trustworthiness of another). As such, similar to trust measures on opinion holders, trust measures on information providers may also be categorised as reputation-based trust measures.

6 PRAISE: A use case in music learning

The PRASIE use cases revolve around the Music Circle system\footnote{http://goldsmiths.musiccircleproject.com/}, an online system that allows peers to be members of more than one community (or group), upload tracks and discussing them by asking and answering questions.

In one use case, there will be a community of school children aged 11–18 and their teachers using the Music Circle system where students can upload and discuss their music practices and both teachers and other students can suggest lesson plans. The system will also be used in a similar manner in higher education teaching. Another use case will integrate the Music Circle system with music composition tools, allowing a community to compose music collaboratively.

In what follows, we illustrate how the TrustIt model can be used by such online music communities. However, we emphasise that the impact of the trust model that we discuss below provides only examples on how can trust measures be used by the community to improve the community’s performance. Naturally, communities might find different uses, based on their ever evolving needs.

6.1 Trust on Opinion Holders

Over time, and with more opinions provided, the trust on opinion holders can help assess the weight given to each opinion. Instead of simply providing the average of opinions (whether the opinions were on a track, or a performance, a lesson plan, or some comment), the trust on opinion holders can aggregate opinions through a weighted arithmetic average, where the weights (described by $\text{trust}^\text{tsw}(\alpha, \beta, o_\beta(\phi))$) are obtained by Equation 5:

$$
\text{trust}^\text{tsw}(\alpha, \phi) = \sum_{\beta} o_\beta(\phi) \cdot \text{trust}^\text{tsw}(\alpha, \beta, o_\beta(\phi))
$$

(46)

For instance, assume the student Mike uploads his performance and two of his fellow colleagues provide their opinions: Kate gives it $5/5$ while Alex gives it $1/5$. When Matt the teacher logs into the system, the final opinion viewed is then not the average $3/5$, but a weighted arithmetic mean based on trust:

$$
\left( \frac{5 \cdot \text{trust}^\text{now}(\text{Matt}, \text{Kate}, o_{\text{Kate}}(\text{Mike’s piece })) + 1 \cdot \text{trust}^\text{now}(\text{Matt}, \text{Alex}, o_{\text{Alex}}(\text{Mike’s piece }))}{\text{trust}^\text{now}(\text{Matt}, \text{Kate}, o_{\text{Kate}}(\text{Mike’s piece })) + \text{trust}^\text{now}(\text{Matt}, \text{Alex}, o_{\text{Alex}}(\text{Mike’s piece }))} \right) / 5
$$

where $\text{trust}^\text{now}(\text{Matt}, \text{Kate}, o_{\text{Kate}}(\text{Mike’s piece }))$ describes how much does Matt at the time being (now) trust Kate’s opinion on Mike’s performance, and $\text{trust}^\text{now}(\text{Matt}, \text{Alex}, o_{\text{Alex}}(\text{Mike’s piece }))$ describes how much does Matt at the time being (now) trust Alex’s opinion on Mike’s performance. Both trust measures are calculated following Equation [5], which essentially check’s the
person’s history of opinions and how far were they from the group’s opinion. In other words, if in the past, Matt’s opinions were very far away from the group’s opinion, then the trust on his current opinion will be low. And if in the past, Alex’s opinions were close to the group’s opinion, then the trust on his current opinion will be high. Such a case would result in a final rating that is lower than the average 3/5.

We believe that the similarity of context when comparing past context to current context is crucial, and we say only past opinions on similar issues are relevant. Not everyone is good (or bad) at giving opinions about everything. For example, one’s opinions on Jazz improvisation might be great, but his opinions on Soprano singing could be terrible. As such, the context is crucial, which we try to capture through the semantic similarity measure.

As for the group opinion, it is simply the arithmetic average of the opinion of each member of the group (Equation 6). And the group itself would describe Matt’s trusted group $G_{Matt}$. A variety of approaches may exist for defining the group $G_{Matt}$:

1. The trusted group contains only one member, and that member is the person calculating the trust measure. In this case, Matt will be the only person in the trusted group ($G_{Matt} = \{Matt\}$), which implies that his trust on others will depend on how close were their past opinions to his own opinions. This case is ideal when the user, Matt in this case, is usually opinionated, i.e. he states his opinions on the majority of issues.

2. The trusted group is composed of all community members ($G_{Matt} = \{everyone\}$). Such a group will naturally be less reliable than the above. However, such a definition is usually useful when the user, Matt in this case, is not opinionated, i.e. he barely forms and states his opinions on many issues.

3. The trusted group is defined by the person calculating the trust measure, Matt in this case. This case requires the system to be a bit more sophisticated, permitting Matt to specify his group of trusted peers. In such a case, Matt can either hand pick his trusted peers, or he can provide constraints on who can be trusted and the peers that satisfy such constraints are automatically added to $G_{Matt}$.

The example presented above illustrates how the trust on opinion holders can be used in the PRAISE Music Circle community of teachers and students, where members may give their opinions on each others’ performances. We stress however that the same approach may be used by any community where members may provide opinions about different entities.

**Requirements on the PRAISE System.** We list below the requirements of the trust on opinion holders module on the PRAISE system:

- Allow members to provide their opinions on the various elements of the system, such as tracks (or performances), discussions, questions, answers, and/or comments.

- Provide a taxonomy of terms that describe the context of the entities that may be rated. As the model relies on past experiences to predict future experiences, there is a need to assess whether a past experience is relevant or not. In the case of the context of the entities being rated, we say one entity is considered relevant to another if the contexts of both are
D3.3 Trust and Reputation in Online Social Learning Communities

semantically similar, and the semantic similarity computation is based on the taxonomy of the terms. We note that this requirement is dependent on the community and the context of its information.

• For a more sophisticated approach, the system may allow members to specify their own trusted groups. Again, there is at least two different methods for implementing this: (1) either allow peers to manually specify their trusted groups, or (2) allow peers to specify some constraints that define whom they believe can be trusted, and the system can then automatically compute the group of trusted members based on who satisfies the constraints.

• For even a more sophisticated approach, the system may also allow opinion change over time. Section 3.2 illustrates how the trust on an opinion holder may be modified to accommodate the character of the opinion holder in giving opinions and how much does he change his opinions. If we were to accommodate such information, then the system should provide means that help extract information about opinion change over time. For instance, the system could allow members to change their opinions, and keep track of all their past opinions. Additionally, the system should be capable of identifying a concrete discussion and its relevant opinions. As discussed in Section 3.2, several simplifications may be made to address this issue. For instance, either each uttered statement (or comment) may be viewed as a discussion, or all statements may be grouped under one discussion.

Impact on the PRAISE Community. Traditionally, opinions have been relied upon heavily as means for measuring reputation of various entities. Auction sites, like eBay, rely on the opinions to help users identify the good sellers or buyers. E-Commerce sites, like Amazon, rely on opinions to help users identify the good products it sells. Social networking sites provide users with the option of giving their opinion on whether they like or not some content (usually with the thumbs up/down). Question-and-answer websites, like StackExchange, also rely on opinions to rank questions, answers, and users.

The PRAISE system will be no different in relying on the opinions of its community members. However, it will aggregate user opinions not using the arithmetic mean, but a weighted arithmetic mean that takes into consideration how much trusted is each community member in giving its opinion on a given issue. The trust on opinion holders module is what provides this functionality. Furthermore, and depending on the definition of the group that will be adopted (see the discussion above), the computed measure may not reflect the reputation of (or group opinion on) the entity in question, but the personalised trust that one community member holds on that entity.

In theory, any entity of the PRASIE system can receive opinions. For instance, one can give opinions on others’ performances, discussions, questions, answers, or comments. And the trust on opinion holders module helps assess how much trustworthy are opinion holders in giving their opinions, and then rank the entities receiving opinions accordingly.

Ranking of entities is usually used to aid in the selection process. For example, ranking of performances can be used to help select what piece will be played at the school’s concert, or to pinpoint which performances can be considered as good examples for the students to try to learn from them. It could also be used by the teacher to find interesting performances in other communities to share, discuss, and/or practice with its class. Ranking of questions can be useful for teachers (or other community members) who are usually interested in answering questions.
Ranking of answers can be used by those interested in the question. It can also be used by teachers to learn alternative ways of answering the question.

The ranking of entities receiving opinions is not a new concept. However, assessing the level of trust on an opinion holder with respect to his opinion on a given issue is not common. As such, a more novel impact of the trust on opinion holders module is using the knowledge on whose opinions are trusted as the basis for recommendations. For example, it may recommend to one member the performance (or discussion, question, lesson plan, etc.) that has been rated highly by his trusted members.

6.2 Trust on Advisers

The trust on advisers aims at assessing the trustworthiness of lesson plans, and ranking them, to help community members (whether teachers or students) choose a lesson plan.

For example, say Alex recommends to Mike that ‘playing twice a week for three months’ will help him ‘play on time’, which is specified in TrustIt as:

\[ \text{Commit}(\text{Alex}, [\text{Practice twice a week for three months}]_{\text{Mike}}, \text{Play on time}) \]

Mike may receive a number of recommended lesson plans from his colleagues, and the trust on advisers module aims at aiding Mike in deciding which recommended lesson plan to trust, and hence adopt.

The trust on advisers computes how much trusted is a given recommended plan. It does so by assessing the history of all past experiences to try to compute the probability of the outcome of such a recommended lesson plan. For example, it computes the probability of whether Mike will adopt the lesson plan, whether he will fulfil the lesson plan correctly (i.e. as intended), and whether he will be able to play on time by the end of the three months.

Again, we say the similarity of context when comparing past recommended lesson plans to a current lesson plan is crucial. For example, Alex might only be a good recommender for lesson plans on Jazz improvisation. Alex’s plans, even if they are very good plans, might not be good plans for Mike since they are intended for members less proficient than Mike. And so on. All such information is learned by the trust model from the history of past experiences, assuming the history is large enough to reflect this information.

An example of the history of experiences \( H \) is presented below:

\[
H = \{ \langle \text{Commit}(\text{Matt}, [\text{Use a metronome}]_{\text{Emily}}, \text{Play on time}) \rangle_{t_7}, \\
\text{Commit}(\text{Emily}, [\text{Use a metronome}]_{t_8}), \\
\text{Observe}(\text{Matt}, [\text{Use a metronome}]_{t_{60}}), \\
\text{Observe}(\text{OnSetCalc}, \text{Play on time})_{t_{61}} \\
\langle \text{Commit}(\text{Alex}, [\text{Practice once a week for a month}]_{\text{Emily}}, \text{Play on time}) \rangle_{t_3}, \\
\text{Commit}(\text{Kate}, [\text{Practice once a week for a month}]_{t_8}), \\
\text{Observe}(\text{System}, [\text{Practice once a week for three weeks}]_{\text{Kate}}_{t_{40}}), \\
\text{Observe}(\text{OnSetCalc}, \neg \text{Play on time})_{t_{41}} \\
\ldots \} \n\]
For example, the first experience states that Matt recommended that Emily should use a metronome to get better at playing on time. Emily accepted the recommended plan, Matt observed her practicing with a metronome, and the OnSetCalc web service confirmed that the uploaded performances by Emily indeed show that she was playing on time.

Given a recommended lesson plan and a history of past experiences like the examples above, the trust on advisers module then calculates how much trusted is the current lesson plan based on the similarity of the past experiences and following one of the preferred trust Equations 34, 35, or 36 (see Section 4.3.1 for a detailed discussion on the various trust equations).

Requirements on the PRAISE System. We list below the requirements of the trust on advisers module on the PRAISE system:

- The system should allow and facilitate the recording of experiences. An experience is composed of 4 elements, each of which needs to be identified and recorded:
  - **Recommended plan.** A recommended lesson plan is composed of four elements, which are the user recommending the plan, the user the plan is recommended to, the lesson plan, and the goal intended to be achieved. Either a linguistic analysis tool needs to be used to help identify recommended plans and their four elements, or the system needs to make plan recommendation more explicit to help identify these four elements.
  - **Adopted plan.** The system should also either keep track of who decides to adopt a given lesson plan.
  - **Executed plan.** Depending on the specific lesson plan, the system should either keep track on how are the adopted lesson plans being executed, or allow others to specify such information. For instance, the first experience in the sample history of experiences presented above, it is the teacher Matt who confirms whether Emily is using a metronome or not. In the second experience of $H$, it is the system itself that confirms whether Kate has been practicing once a week for a month or not by checking her uploaded performances.
  - **Fulfilled goal.** Depending on the specific goal, the system should either identify what goal is being fulfilled, or allow others to specify such information. For instance, in the first two examples of the sample history of experiences presented above, the OnSetCalc web service is responsible for checking whether uploaded performances are being played on time. If the goal was a more subjective goal that cannot be verified in an automatic way, then maybe the feedback of community members may be used to identify whether the goal has been fulfilled or not.

- Taxonomies of actions, goals, and roles need to be provided. This is crucial as it helps compute the semantic similarity between experiences, which we use as an indication of relevance between a past experience and a current lesson plan. We note that this requirement is dependent on the community and its lesson plans.

- A meronomy of actions needs to be provided. This is crucial as it helps compute the empowerment relation between plans (which helps specify, in addition to semantic similarity, how much relevant an experience is to a given lesson plan). We note that this requirement is also dependent on the community and its lesson plans.
Impact on the PRAISE Community. The main goal of the trust on advisers module is to sort lesson plans based on the most trusted plans. This could help the people looking for good plans, such as students who want to improve their performances.

However, this module could also be used to help the teacher learn new plans by pinpointing new good (trusted) lesson plans. For instance, one way is to bring successful lesson plans suggested by fellow colleagues, or even lesson plans from other classrooms, to the attention of the teacher. This helps in evolving the teacher’s stack of lesson plans and making it more varied over time.

In addition to sorting lesson plans, the trust on advisers module can also be used to rank those who recommend lesson plans. Ranking those recommenders may be understood as providing information on who is good at giving advice on improving performance. This could be used then by the teacher to pinpoint some students who may be considered good advisers, and promoting them to become the “teacher’s little helpers”. Such students can then aid the teacher in helping their colleagues in the class.

Alternatively, ranking lesson plan recommenders could also be used by any community member who is interested in looking for new lesson plans. As such, the community member can look for lesson plans based on who is suggesting these plans. For instance, one might say “I trust Mark’s lesson plans, let us see whether he has any new plans today!” Or the system can inform the user when a trusted member suggests a new lesson plan.

Furthermore, community members who do not wish to hear every single person’s advice may also ask the system to filter advice provided by members whose trustworthiness at giving advice is below a certain threshold.

6.3 Trust on Information Providers

Tracks and performances may be discussed by asking and answering questions. Answers may provide general information, such as ‘Chet Baker used similar chord progressions’. However, as discussed earlier in Section 5 there are several issues with providing answers or comments online in the traditional way. For instance, there is the issue of noise, which hinders finding relevant content. For example, how can one assess the reliability of content? In other words, how can we decide what to believe and what not to believe? Allowing anyone to contribute information in large online communities sometimes results in a massive volume of online information, which is usually very unorganised and suffers from redundancy, spam, bias, and lies.

The trust on information and information providers can help rank information provided (or answers) for the user. The ranking will take into consideration the reliability of information, its relevance, its redundancy, and its significance. When a new comment is added, it does not necessarily appear at the top of the list; its rank will depend on how much is the information provider trusted in providing information in the given context.

Again, we say the similarity of context when comparing past context to current context is crucial, and we say only past information on similar issues are relevant. Not everyone is good (or bad) at providing information about everything. For example, one may be informative when it comes to Jazz music, but not Operas. As such, the context of information is crucial, which we try to capture through the semantic similarity measure.
D3.3 Trust and Reputation in Online Social Learning Communities

For example, say the teacher opens the floor for fellow students to provide remarks on their colleague’s music performance. Alex and Emily initially leave the only two remarks ‘It would be interesting to see this played in C minor’ and ‘This will be played at the concert tonight’ respectively. After several months, while revisiting this performance (and gaining new knowledge), Mike leaves the comment ‘Chet Baker used similar chord progressions’.

Now assume Kate logs in to the discussion. The question then is, in what order should the three remarks be presented for Kate? We say information is presented based on their rank. The rank of past information is mostly influenced by the measure describing how much valuable the information itself is:

\[
\text{trust}^{\text{past}}(\alpha, i_\beta(\phi)) = \frac{\lambda_b \cdot \text{reliability}^\alpha(i_\beta(\phi)) + \lambda_r \cdot \text{relevance}^\alpha(i_\beta(\phi)) + \lambda_d \cdot \text{redundancy}^\alpha(i_\beta(\phi)) + \lambda_s \cdot \text{significance}^\alpha(i_\beta(\phi))}{\lambda_b + \lambda_r + \lambda_d + \lambda_s}
\] (47)

where reliability, relevance, redundancy, and significance are used in assessing how much valuable a piece of information is, and are calculated according to Equations 39, 41, 43, and 45 respectively.

The rank of new information is mostly influenced by the trustworthiness of the information provider in providing such information (\(\text{trust}^{\text{now}}(\alpha, \beta, i_\beta(\phi))\)), which is calculated following Equation 37.

For example, let us say both pieces of old information provided by Alex and Emily are relevant to the discussion since they have not been reported by any community member (i.e. \(\text{relevancy}^{\text{now}}_{\text{Kate}}(i_{\text{Alex}}(\phi)) = \text{relevancy}^{\text{now}}_{\text{Kate}}(i_{\text{Emily}}(\phi)) = 1\)) and both are not redundant as they have not been posted by the same author at another point in time (i.e. \(\text{redundancy}^{\text{now}}_{\text{Kate}}(i_{\text{Alex}}(\phi)) = \text{redundancy}^{\text{now}}_{\text{Kate}}(i_{\text{Emily}}(\phi)) = 1\)). Alex’s comment however is much more reliable than Emily’s, as it has received numerous ‘thumbs up’, whereas Emily’s comment received none (i.e. say \(\text{reliability}^{\text{now}}_{\text{Kate}}(i_{\text{Alex}}(\phi)) = 0.8\) and \(\text{reliability}^{\text{now}}_{\text{Kate}}(i_{\text{Emily}}(\phi)) = 0\)). On the other hand, Emily’s comment was retweeted by others, whereas Alex’s comment was not (i.e. say \(\text{significance}^{\text{now}}_{\text{Kate}}(i_{\text{Alex}}(\phi)) = 0\) and \(\text{significance}^{\text{now}}_{\text{Kate}}(i_{\text{Emily}}(\phi)) = 0.6\)). As such, and assuming relevance, redundancy, reliability, and significance are given equal weights (i.e. \(\lambda_b = \lambda_r = \lambda_d = \lambda_s = 1\)), we have:

\[
\text{trust}^{\text{now}}_{\text{Kate}, i_{\text{Alex}}(\phi)} = \frac{1 + 1 + 0.8 + 0}{4} = 0.7
\]

\[
\text{trust}^{\text{now}}_{\text{Kate}, i_{\text{Emily}}(\phi)} = \frac{1 + 1 + 0 + 0.6}{4} = 0.65
\]

Now assume given Mike’s history of past comments, he currently accumulates a trust measure (from Kate’s point of view) of \(\text{trust}^{\text{now}}_{\text{Kate}, \text{Mike}, i_{\text{Mike}}(\phi)} = 0.68\).

But when is information considered old? In other words, how do we decide whether we should rely on the trust gained by the information itself (\(\text{trust}^{\text{now}}_{\text{Kate}, i_{\text{Mike}}(\phi)}\)) or the trust of the information holder (\(\text{trust}^{\text{now}}_{\text{Kate}, \text{Mike}, i_{\text{Mike}}(\phi)}\))? We say the final trust measure is a combination of both, and the weight given to each depends on two criteria: (1) how recent is the information, which is specified as \(\Theta_{\beta}(\phi) \in [0, 1]\), where a value of 0 specifies old information and a value of 1 specifies recent information, and (2) whether the discussion is still lively or not, which is specified as \(\Psi_\phi \in [0, 1]\), where a value of 0 specifies
minimum liveliness and a value of 1 specified maximum liveliness. The basic idea behind this is that the more recent a piece of information is, then the more weight is given to the trust on the information provider; and the more lively the discussion is, then the more weight is given to the trust on the information itself. We also note that assigning values to both criteria is domain dependent. For example, in a highly dynamic community, a one day old comment might be considered very old, whereas in less dynamic communities, a one day old comment might be considered very recent. In summary, the dynamics of interaction of a community should affect how these values are assigned. However, this is outside the scope of this paper.

The final rank of the information is then defined accordingly:

$$\text{rank}^\text{now}(\alpha, i_\beta(\phi)) = |\Theta_{i_\beta(\phi)} - 1| \cdot \Psi_\phi \cdot \text{trust}^\text{now}(\alpha, i_\beta(\phi)) + (1 - |\Theta_{i_\beta(\phi)} - 1| \cdot \Psi_\phi) \cdot \text{trust}^\text{now}(\alpha, i_\beta(\phi))$$ (48)

To understand the equation above, we look at the limits of the variables $\Theta_{i_\beta(\phi)}$ and $\Psi_\phi$. For example, the equation states that if the information is recent (i.e. $\Theta_{i_\beta(\phi)} = 1$) and the discussion is dead (i.e. $\Psi_\phi = 0$), then we only consider the trust on the information provider ($\text{trust}^\text{now}(\alpha, i_\beta(\phi))$).

As such, in the example above, if we assume the discussion is dead ($\Psi_\phi = 0$), Alex’s and Emily’s comments are very old ($\Theta_{i_{Alex}(\phi)} = \Theta_{i_{Emily}(\phi)} = 0$), and Mark’s comment is recent ($\Theta_{i_{Mark}(\phi)} = 1$), then we will have:

$$\text{rank}^\text{now}(\text{Kate}, i_{Alex}(\phi)) = \text{trust}^\text{now}(\text{Kate}, i_{Alex}(\phi)) = 0.7$$
$$\text{rank}^\text{now}(\text{Kate}, i_{Emily}(\phi)) = \text{trust}^\text{now}(\text{Kate}, i_{Emily}(\phi)) = 0.65$$
$$\text{rank}^\text{now}(\text{Kate}, i_{Mark}(\phi)) = \text{trust}^\text{now}(\text{Kate}, i_{Mark}(\phi)) = 0.68$$

As a result, Alex’s comment will be ranked top, followed by Mike’s comment, and then Emily’s.

**Requirements on the PRAISE System.** We list below the requirements of the trust on information providers module on the PRAISE system:

- Allow users to rate (or give their opinion on) information.
- Allow users to report spam and other irrelevant information.
- Use a linguistic analyser to search for redundant information in a discussion. As discussed in Section 5, a simplified approach that may be used initially is to assume that two pieces of information are redundant if they are equal. However, in many cases that might not be true, since there may be minor modifications in the text but the content in general is still redundant. As such, it would be interesting to make use of linguistic analysis to help pinpoint redundant information in an automatic way.
- Allow the user to provide data that may be used to describe the attention captured by information. One aspect of the trust on information providers module is to assess whether a given
piece of information is significant or not based on how much attention it receives. Attention may be captured by a number of data, such as how many times was the information shared (emailed, tweeted, posted on Facebook, etc.), how many likes did the information receive, how many other comments did it stir, and so on. As such, any kind of such interaction would be useful.

- Provide a taxonomy of terms that describe the context of information. Similar to the remaining trust modules of this paper, we rely on past experiences to predict future experiences. As such, there is a need to assess whether a past experience is relevant or not. In the case of information providers, we say one piece of information is considered relevant to another if the contexts of both are semantically similar, and the semantic similarity computation is based on the taxonomy of the terms. We note that this requirement is dependent on the community and the context of its information.

- For a more sophisticated approach, the system may allow members to specify their trusted groups. Note that the trust on information providers makes use of the trust on opinion holders. As such, and if a more sophisticated approach is required, then the system may allow members to either manually specify their trusted groups, or specify some constraints that define who can be trusted (and then the system can automatically compute the group of trusted members based on who satisfies the constraints).

Finally, we note that information is currently specified as answers in the PRAISE Music Circle system. An interesting aspect to add, though not a strict requirement, would be to allow comments outside the scope of discussions’ questions and answers. For example, some community member might still like to add a comment, or a piece of information, without asking any questions. Similarly, questions may be commented upon, answers may be commented upon, and even comments may be commented upon. This could make the system richer in providing information, and it may also help identify which information has received more attention.

**Impact on the PRAISE Community.** First and for most, the trust on information providers module can be used by community members to assess how much can they trust a given piece of information.

These trust measures can also be used in sorting online discussions in a way that helps users read more valuable information first, and suppresses noise resulting from unreliable, irrelevant, redundant, and insignificant information.

Additionally, the trust model may also be used by the teacher to help figure out which students are contributing ‘valuable’ information to online discussion, or the trust model can even rank students based on their participation in online discussions.

## 7 Conclusion

This paper has introduced the TrustIt model, a model that assesses the trustworthiness of feedback and feedback providers. We divide feedback into three categories: (1) opinions, (2) advice, specified as plans that aim at fulfilling given goals, and (3) general information. As such, the TrustIt
model is divided into three modules: (1) trust on opinion holders, (2) trust on advisers, and (3) trust on information providers.

In all of these modules, we rely on similar past experiences in order to predict the outcome of future experiences, and we use this predicted outcome in describing the trustworthiness of members and entities in general. As such, the proposed approach is based on similarity measures, probability theory, and information theory.

Finally, we note that feedback is usually the main component in assessing the trustworthiness of various entities (at least in computational trust models). Without any form of feedback, it would be difficult to assess these entities. In our work, we have also illustrated how, by assessing the trust on feedback and feedback providers, we can generalise to assess the trust of any entity in the system, such as performances in PRAISE, teachers, good advisers, informative people, and so on.

A Note on Novelty. As for the novelty of our proposed work, concerning opinions, we say that instead of simply providing the average of opinions when assessing the trustworthiness of some entity, we propose aggregating opinions through a weighted arithmetic average, where the weights describe how much trusted is the opinion holder. The trust on a given opinion holder represents how much reliable are the opinions of the opinion holder with respect to a given context. We address this issue by calculating this trust measure in terms of how far were the past opinions of the same opinion holder from that of the group opinion.

Another novelty of the trust on opinion holders module lies in analysing the character of opinion holders (such as labelling them as decisive if they usually never change their opinions, or indecisive if they are always changing their opinions, or persuaders if they usually convince the group to move towards their opinion, and so on). We illustrate how such characters can be accounted for when assessing the trust in an opinion holder.

Concerning the trust on advisers, we believe this is a new concept in the field of trust and reputation. Usually either the trust or reputation of items is assessed, or the trust on (or reputation of) people in the context of performing a simple task is assessed. For example, one may assess the trust on sellers, buyers, teachers, doctors, etc. However, the trust on advisers module computes the trust on advisers (where an adviser can either be a person, or peer, or even an algorithm, say a recommender system) in the context of performing more complex tasks. The task of advisers is to recommend a given plan for a given person so that a given goal can be achieved. Hence, the trust on advisers should consider all these four orthogonal aspects: the adviser, the plan, the goal, and the person this plan is recommended to. To our knowledge, we believe we are one of the few to address the issue of trust on advisers.

Concerning the trust on information providers, we present our view of what is the current problem with the presentation of online information, which we believe does not address the most crucial issue of figuring out what information is truly valuable. Accordingly, we propose an approach for addressing this problem by assessing the reliability, relevance, redundancy, and significance of information.

A Note on Evaluation. For evaluating the different trust modules, we will rely on different techniques and different datasets. For instance, we will rely on simulation for evaluating the trust...
on advisers module. This is because it has been difficult to obtain online datasets that could be used to describe advisers and their plans, along with data on the different users who adopt and execute these plans and the outcome of these executions. In most cases, we were capable of obtaining data that describes plans, without information on the results of the execution of these plans by different users.

Concerning the trust on opinion holders and information providers, it might be possible to find online datasets that map our model’s required data. For instance, we plan to evaluate the trust on opinion holders by using the music recommendation dataset from the music website Last.fm or the Yahoo Research dataset of various ratings. We also plan to evaluate the trust on information providers by using the dataset of StackExchange (www.stackexchange.com), a question-and-answer website on topics in many different fields.

Future Research Work. We list below a set of issues that we plan to address in the following year, and we note that since this research is still in progress, the list of future work might continue to evolve with the research.

- In any trust model, there is always the issue of how to introduce new and varied entities (whether tracks, questions, answers, etc.) when older and more established ones are always ranked higher. Since entities with higher trust catch more attention, then ranking entities solely on their trustworthiness will simply ensure that whoever gets some attention first will continue to receive the most attention later on. As a result, new entities might not receive a fair chance of being presented to the community. Similarly, the user might get stuck into viewing only what he used to trust most, and not be given the chance to view varied entities (such as new music genres, different point of views, and so on). As such, there is a need to ensure that both new entities and varied entities are always given a chance to be presented to the community in an impartial manner.

  Different approaches may exist to achieve this, one of which is to assume entities have by default a high trust value which drops with negative feedback. Another approach might simply try to randomly introduce varied or new entities first for a brief period of time (where the span of this period should be long enough to allow the entity to start collecting sufficient feedback from the community). We plan to study the possible different approaches to achieve this and choose the most appropriate one for the PRAISE system.

- At first glance, the parameters used in this model may seem numerous, complicating the trust model even further. Future work will assess these parameters and whether they add any complexity to the trust model, provide guidelines on how one may assign the values of these parameters, and study whether there is any dependency amongst these parameters or not.

- Concerning the trust on opinion holders, if the system will permit the change of opinions over time, then we might investigate which characters might be of most interest to the community. We might also investigate in more detail how to best make use of such information. For instance, shall we investigate alternative and more interesting approaches on how to

---

Footnote:

7 http://webscope.sandbox.yahoo.com/catalog.php?datatype=r
incorporate such information into the final trust measure? Or should we introduce new measures to describe such information?

• The trust on opinion holders model allows the user to manually specify whose opinions to trust. The other trust modules do not provide such an option at the moment. The trust modules on advisers and information providers rely solely on past similar experiences. We plan to allow the users to manually modify their trust on others (or other entities). In such a case, the model needs to be revisited to incorporate such personal hard coded trust measures in the final calculated measure. This could be useful in many scenarios. For instance, consider the case of assessing lesson plans suggested by students. Not all suggested lesson plans will be accepted by the teacher. We believe that whether or not a teacher accepts to adopt a suggested lesson plan or not should influence, from the perspective of the teacher, the reliability of those who suggested this lesson plan. For example, students who never get any of their suggested lesson plans adopted by the teacher should be considered less reliable by the teacher with respect to their future recommended lesson plans.

• In the trust on information providers, when assessing the relevance of information, we plan to study incorporating the number of reports the information receives into the final measure. In fact, this approach may also be used by other parts of the model. For example, the number of opinions received may influence the final reliability measure, the number of similar past experiences considered might also influence the final trust measures, and so on.

• In the trust on information providers, when assessing the significance of information, we plan to define a clear set of interactions that we will rely on for extracting data describing the attention a piece of information receives, and study the possibility of giving different types of interactions different weights.

References


