Trust Trust Me (The Additivity)

Ken Mano, Hideki Sakurada, Yasuyuki Tsukada

Trust Trust Me (The Additivity)

Ken Mano, Hideki Sakurada, and Yasuyuki Tsukada

NTT Communication Science Laboratories, NTT Corporation,
3-1 Morinosato Wakamiya Atsugi Kanagawa 243-0198 Japan,
`[mano.ken|sakurada.hideki|tsukada.yasuyuki]@lab.ntt.co.jp`

Abstract. We present a mathematical formulation of a trust metric using a quality and quantity pair. Under a certain assumption, we regard trust as an additive value and define the soundness of a trust computation as not to exceed the total sum. Moreover, we point out the importance of not only soundness of each computed trust but also the stability of the trust computation procedure against changes in trust value assignment. In this setting, we define trust composition operators. We also propose a trust computation protocol and prove its soundness and stability using the operators.

Keywords: Trust, Metric, Protocol, Soundness, Stability, Subjective logic

1 Introduction

We discuss mathematical formulation of a trust metric. There are two classical approaches to such formulation, namely logical [1, 2, 7–9, 11, 13], and computational [3, 4, 10, 12, 14] approaches. The logical approach involves modal logics such as Epistemic logic or Doxastic logic and is aimed at revealing the logical structure of a trust problem. The computational approach introduces operations on a trust metric to compute the required trust values, and involves probability theory, the subjective logic or fuzzy logic to justify the validity of the computed values. Our approach belongs to the latter and has the characteristics that a trust metric is formulated as a quality and quantity pair.

Why quality and quantity? To recall how such problems have been treated in existing research, let us consider the case of the subjective logic.

The subjective logic is a logical system with the set of opinions (b, d, u) as its domain. The elements b, d and u of the tuple represent the proportions of belief, disbelief and uncertainty, respectively. Therefore, it is assumed that $b, d, u \in [0, 1]$ and $b+d+u = 1$. This is not a tailor-made theory for trust, but rather a general system for uncertainty. There are studies that have applied the subjective logic to the computation of trust metrics [5, 6].

A sequential composition of opinions called discounting, denoted by $\otimes$, is defined as follows. Suppose A's opinion on the trust concerning B is (b_{AB}, d_{AB}, u_{AB}), and B says that his opinion on the trust concerning C is (b_{BC}, d_{BC}, u_{BC}). Then, the trust of A concerning C is

$$\begin{aligned} &(b_{AB}, d_{AB}, u_{AB}) \otimes (b_{BC}, d_{BC}, u_{BC}) \\ &= (b_{AB} \cdot b_{BC}, b_{AB} \cdot d_{BC}, 1 - b_{AB} \cdot b_{BC} - b_{AB} \cdot d_{BC}). \end{aligned}$$

Since the certainties (belief and disbelief) are defined as a multiplication of values in $[0, 1]$, they decrease unless the case of perfect trust or distrust, and uncertainty increases accordingly. If we interpret b and d as probability, this seems natural. But is it always valid for trust composition?

Let us consider the following story regarding measurement as an analogy. To measure a target C, we must use two measuring instruments A and B sequentially. That is, B directly measures C and makes some output. Then A measures the output of B, and finally makes some output that the observer actually sees.

Then, if the accuracy of A is 12 bits and that of B is 16 bits, the total accuracy is 12 bits. If the accuracy of A is 20 bits and that of B is 16 bits, the total accuracy is 16 bits. The accuracy is regarded as a quantitative metric of the trustworthiness of the results, and their composition is not determined by multiplication but by *min*.

We are seemingly making a similar judgment in the everyday life. For instance, let us consider a situation where A is informed concerning C from B who has been a friend of A for over 10 years. If the information is "I came to know C last year, and he is a fairly good guy.", then it is a rational option for A to believe the information as it is. On the other hand, if the information is "C is a friend from childhood, and I entrust him with the management of all my property.", then A typically will not believe the information as it is. Although the degree to which the value of the information is discounted depends on the person, it is natural to regard the value as quantitatively limited by the length of the friendship between A and B.

Thus we propose using the quantity as an element of the trust metric (without converting it to a proportion) to represent the uncertainty caused by the quantity. We then define the quantity of the sequential composition of two trust values as the *min* of their quantities. The idea of using a quality and quantity pair is not new. For instance, it is used implicitly in [3].

For the trust computation we also need parallel composition. In the subjective logic, parallel composition is called consensus, which is defined based on the quantitative summation of evidence of belief and disbelief. At that time, a supplemental parameter called atomicity is introduced in order to map the opinions to evidences. For trust as a quality and quantity pair, we can define the corresponding composition simply using a quantity-weighted average, without any supplemental parameters. But here we face the problem of evidence independence in the subjective logic.

For instance, suppose that C performed a good action for each of A and B, and they regard the actions as evidences of trust, respectively. Then, in order to combine these evidences using consensus, the actions must be probabilistically independent of each other.

We regard this requirement not always appropriate at least in the context of human trust. This is because there is no general way to decide whether or not two actions performed by a person are independent. Moreover, is independence truly necessary? If we think that a good guy tends to perform a good action, then any two good actions that he performs are somewhat dependent on each other.

Should we abandon combining them? Let's return to common sense. People would naturally think as follows: he is really a good guy 'cause he did good twice!

This casual sense provides us with a completely new idea for trust computation, namely to regard trust as an additive value. We say a value is additive when the value of the whole system is the total sum of all subsystem values. We do not claim that this is the only solution to this problem. However, we believe this is at least one valid mathematical modeling of trust.

Treating trust adequately as an additive value is nontrivial. Even if the definition of each composition is valid, its application generally may not be valid since the computed value can be invalidly amplified by duplicate counting. To avoid such invalidity, we must clarify the way of determining the basic trust values that each person initially holds, and the way of combining them. We formulate this problem using a kind of ordered algebra where the partial order $\preceq$ represents the amount of information.

This paper is organized as follows. Section 2 explains how our model is applied in reality. In Section 3, we describe the basic problem setting and the trust composition operators. In Section 4 we define the validity of trust computation (called soundness and stability in this paper) and introduce the syntax of linear terms for representing valid computation. In Section 5, we present a protocol for distributed trust computation and prove its validity using the algebraic properties of operators. In Section 6, we present a comparison with existing studies, and in Section 7 we discuss inherent issues when applying our model. Due to the lack of space, all proofs are omitted.

2 Application

We consider a situation in which trust values are distributed in a network. That is, we assume that people hold their trust values concerning others, and do not assume the existence of a trusted third party. We also assume they may answer correctly, ignore the question, or tell a lie when they are asked about their trust values.

The trust computation presented in this paper is applicable to any network service, e.g., SNS and market place, in which trust or reputation information is needed. For instance, the stars used by Amazon can be regarded as trust information represented by quality and quantity pair, where quality is represented by the proportion of five cases. Moreover, PKI and ad hoc networks are expected to be good applications.

One of the main contributions of our paper as regards such applications is to enable de-centralized management of trust information. Distributed trust management has some advantages compared to server-centric management. One is that local trust information is easy for the holder to add and/or update, and thus users can obtain more correct and up-to-date information. It is also advantageous that the user can choose a preferred source of trust information. By asking to a person who is reliable and who has the same taste, we can obtain desirable trust information.

3 Trust

3.1 Quality and Quantity of Trust

In this section, we define trust as a pair of quality and quantity. For any person A and person B distinct, a trust t_{AB} of A concerning B is a pair (p_{AB}, q_{AB}). Here q_{AB} is a non-negative real called the quantity of t_{AB}. Its intended interpretation is the amount of interaction between A and B, for instance, the number of communication messages, and the transaction value. p_{AB} is called the quality of t_{AB}, and we assume $p_{AB} \in [0, 1]$. We also assume $p_{AB} = 0$ when $q_{AB} = 0$, so we often write 0 instead of $(0, 0)$. For instance, let quantity be the number of queries and quality the rate of correct answers. If A sent B 100 queries, and received 90 correct answers in the past, then the trust of A to B is $(0.9, 100)$.

3.2 Composition Operators of Trust

We introduce two types of trust composition operators: parallel and sequential.

Parallel composition $\uplus$ of trust

Assume someone asked A and B about their trust concerning C, and received $t_{AC} = (p_{AC}, q_{AC})$ and $t_{BC} = (p_{BC}, q_{BC})$, respectively. Then, assuming that A and B are totally reliable, how should the person consolidate these two values? We define the parallel composition $\uplus$ of trust as follows:

$$t_{AC} \uplus t_{BC} = \left(\frac{q_{AC} \cdot p_{AC} + q_{BC} \cdot p_{BC}}{q_{AC} + q_{BC}}, \, q_{AC} + q_{BC} \right).$$

That is, the composition of quantities is simple addition and that of qualities is a quantity-weighted average. We define $(0, 0) \uplus (0, 0) = (0, 0)$. This definition can be justified by the following analogy: quantity is the number of independent trials, and quality is the success probability. For instance, if $t_{AC} = (0.9, 100)$ and $t_{BC} = (0.8, 1000)$, then $t_{AC} \uplus t_{BC} = (0.81, 1100)$.

The operator $\uplus$ is associative and commutative, and satisfies $0 \uplus t = t$.

Sequential composition $*$ of trust

Suppose that B told A that the trust of B concerning C is $t_{BC} = (p_{BC}, q_{BC})$, and that the trust of A concerning B is $t_{AB} = (p_{AB}, q_{AB})$. Then, to compute the trust concerning C, A should discount t_{BC} by t_{AB}.

We define the sequential composition $*$ of trust as follows:

$$t_{AB} * t_{BC} = (p_{AB} \cdot p_{BC}, \, min(q_{AB}, q_{BC})).$$

According to the analogy of probability, the quality of the composition result is the expected value in the case of p_{BC} with probability p_{AB}, and 0 with probability $1 - p_{AB}$. The definition of quantity is based on the idea that A can quantitatively rely on the trust value t_{BC} provided by B at most q_{AB}.

For instance, if $t_{BC} = (0.9, 1000)$ and $t_{AB} = (0.8, 100)$, then $t_{AB} * t_{BC} = (0.72, 100)$. The sequential composition represents an inferiorization of trust by communication. We think that information is degraded by communication since people can tell a lie. A liar can provide either a higher or lower trust value than the truth, but because of the nature of the trust problem, higher is worse. Moreover, since we cannot generally gather all the trust information in a network, the computed value is necessarily quantitatively smaller than the network-wide total value. The above definition reflects these observations.

Roughly speaking, the above definition of $*$ implicitly assumes the following properties of a lie: if the trust of A concerning B is (p_{AB}, q_{AB}), when B informs A of the trust (p, q),

1. p is at most $1/p_{AB}$-times higher than the truth, and
2. q is not too large when the truth is less than or equal to q_{AB}.

Under such assumptions, the above definition is justified. In Section 4.1 we present a generalized form of these assumptions. The operator $*$ is associative and commutative, and satisfies $0 * t = 0$.

Example 1 We do not insist that the above is the only possible definition of compositions. It is simply a running example, and variations are possible depending on the purpose and user preference. The following are examples of such variations.

1. Replacing the parallel composition with
$$t_{AC} \uplus_{max} t_{BC} = (max(p_{AC}, p_{BC}), max(q_{AC}, q_{BC})).$$
2. Replacing the sequential composition with
$$t_{AB} *_2 t_{BC} = (p_{AB} \cdot p_{BC}, min(p_{AB} \cdot q_{AB}, q_{BC})).$$

The former example has little practical significance, but is useful for making it clear that the validity argument in this paper does not depend on probability theory.

The latter example is more significant. In the definition of the quantity of the composition, the first argument of min is replaced with $p_{AB} \cdot q_{AB}$, which implies that a lower p_{AB} yields a smaller quantity. Intuitively, this definition says that information from low quality source is unreliable both qualitatively and quantitatively. It is, however, noticeable that this sequential composition is neither associative nor commutative.

3.3 Problem of Duplicate Counting

In the previous section, we defined composition operators of trust. However, their applications are not always valid. This is closely related to duplicate counting and the additivity of trust. In this section, we present three types of duplications in which the applications of operators are invalid.

Example 2 Assume that trust values among $A, B_1, \cdots, B_{100}, C, D$ are defined as $t_{AB_i} = (1, 1)$, $t_{B_iC} = (1, 1)$ and $t_{CD} = (1, 10)$ as shown in Figure 1. Then, let us consider the following examples of trust calculations:

1. "The trust of A concerning D is $t_{AB_1} * t_{B_1C} * t_{CD} \uplus \cdots \uplus t_{AB_{100}} * t_{B_{100}C} * t_{CD} = (1, 100)$".

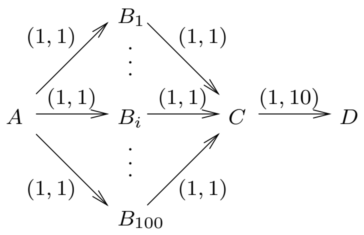

Fig. 1. Duplicate Counting

2. "The trust of A concerning C is $t_{AB_1} * t_{B_1C} \uplus \cdots \uplus t_{AB_{100}} * t_{B_{100}C} = (1, 100)$".

Are these calculations valid? ∎

The problem with the former example is clear, that is, although the quantity is originally 1, it is (or at least seems to be) invalidly amplified to 100 because of the duplicate counting of the trust value t_{CD}. On the other hand, there seems to be no apparent duplication in the latter example, but the problem here is how $t_{B_1C}, \cdots, t_{B_{100}C}$ are determined. If such trust values are determined since $B_1, \cdots, B_{100}$ observed just one action of C simultaneously, then the total quantity should be 1.

For instance, suppose that there is an NGO with 100 members $B_1, \cdots, B_{100}$. Assume that C made a donation of 100 dollars because he approved of its aim, and that, based on this single fact, each B_i decided to give 100 dollars' worth of trust concerning C. Then is it valid to *add* the trust values quantitatively and to conclude that C obtained 10000 dollars' worth of trust?

The two problems are similar but different. The problem of 2 is concerned with how to determine the basic trust values, while the problem of 1 is concerned with how to calculate using the basic values.

Let us first consider the problem of 2. We introduce the distinction between basic trust values and others. A basic trust value (or simply, a basic trust) is a trust value determined by each person based on his direct and exclusive experiences. The other trust values are those computed using communicated information.

We say that someone's experience is direct if he sees it with his eyes or hears it with his ears. Hearsay information and conjecture are not direct. We say someone's experience is exclusive if he is the only one who experienced it. If we must think of several people's experiences concerning a single event, a share of the quantity is distributed to each person so that the sum is 1, e.g., $1/n$ to n individuals. If the share cannot be determined, such an experience is regarded as not direct. Under this assumption, the addition of quantity in $\uplus$ is justified.

We denote the set of all persons by $\mathcal{P}$, and assume $\mathcal{P}$ is finite. For any $A, B \in \mathcal{P}$ distinct we write t_{AB} to denote the basic trust of A concerning B. We also call t_{AB} a basic trust concerning B, or simply a basic trust. Based on the assumption that a basic trust is determined by direct and exclusive experiences,

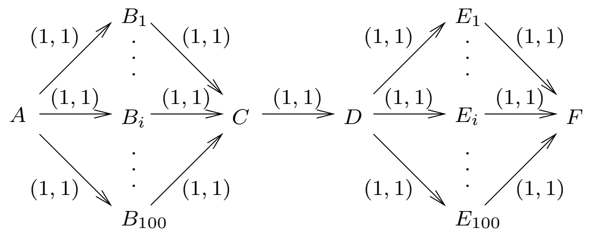

Fig. 2. Duplicate Counting of Communication Pathways

we regard basic trust as an additive value, and define the total basic trust t_B of B as the total sum of the basic trusts concerning B:

$$t_B = \sum_{P \in \mathcal{P} - \{B\}} t_{PB}.$$

Next, let us consider the problem of 1. In Figure 1, if t_{CD} is the only non-zero trust concerning D, the result of this example exceeds the total basic trust concerning D because there is duplicate counting of t_{CD}. Such duplication must be avoided for a valid computation of additive values.

Then, what about the duplicate counting of basic trusts not directly concerning D, that is, the trusts on the communication pathways to D, when calculating a computed trust concerning D?

Example 3 In the situation shown in Figure 2, consider a calculation that involves, for example, first calculating the following values for 100 paths,

$$A \to B_i \to C \to D \to E_i \to F \ (i = 1, \cdots, 100)$$

and then summing them. Is this valid? ∎

Note that the paths are chosen so that they share just one C-D edge. In the above example, the calculation result does not exceed the total basic trust concerning F. However, in the summation

$$\sum_{i=1}^{100} t_{AB_i} * t_{B_iC} * t_{CD} * t_{DE_i} * t_{E_iF} = (1, 100),$$

a large amount of trust is divided into 100 parts, which run through the C-D edge with relatively small quantity. Therefore, this violates the basic idea of sequential composition whereby C can quantitatively rely on the trust value provided by D at most the quantity of t_{CD}.

In fact, if t_{CD} is updated to $(0.5, 2)$ by a new experience, the calculation result changes to $(0.5, 100)$. This means that a result with quantity 100 is heavily influenced by a change in quantity 1. Such a situation is contrary to the nature of quantity, and thus should be avoided.

In the next section, we will formulate two properties implying that the above problem does not occur using a binary relation on trusts.

4 Network of Trust

Using the composition operators presented in the previous section, we investigate trust computation by gathering trust values from people on a network.

4.1 Soundness and Stability

In this section, we define the validity of trust computation independent of the specific way of calculation. In the rest of this paper, the quality and quantity of trust t is denoted by $p(t)$ and $q(t)$, respectively.

First, we define a binary relation $\preceq$ on trusts as follows:

$$t \preceq t' \text{ iff } \exists\, t_1, t_2 \; t \uplus t_1 = t_2 * t'.$$

This definition states that the left-hand side $t \uplus t_1$, of which t is a part, is equal to the right-hand side $t_2 * t'$, which is inferior to t' by t_2. That is, $\preceq$ means that the left-hand side is partial and inferior to the right-hand side, and thus is regarded as representing the relative amount of information.

For instance, $(0.8, 10) \preceq (0.9, 100)$ clearly holds. Moreover, $(0.8, 100) \preceq (0.9, 100)$ (by letting $t_1 = (0, 0)$ and $t_2 = (8/9, 100)$) and $(0.9, 50) \preceq (0.8, 100)$ (by letting $t_1 = (0.7, 50)$ and $t_2 = (1, 100)$) also hold. On the other hand, $(0.8, 100) \not\preceq (0.9, 10)$ and $(0.9, 90) \not\preceq (0.8, 100)$.

We present basic properties of $\preceq$. For any trust $\mathit{\Delta} t$, $t \preceq_{\mathit{\Delta} t} t'$ iff $q(t) \preceq q(t') \;\wedge\; t' \preceq t \;\uplus \mathit{\Delta} t$.

Lemma 4 $\preceq$ satisfies the following properties.

1. Reflexivity: $t \preceq t$.
2. Transitivity: If $t \preceq t'$ and $t' \preceq t''$, then $t \preceq t''$.
3. Anti-symmetry: If $t \preceq t'$ and $t' \preceq t$, then $t = t'$.
4. Decreasing: $t * t' \preceq t'$.
5. Monotonicity: If $t \preceq t'$, then $t'' \uplus t \preceq t'' \uplus t'$.
6. Semi-distribution: $t * (t' \uplus t'') \preceq t * t' \uplus t * t''$.
7. Overtaking: For any trust $\mathit{\Delta} t$, if $t \preceq_{\mathit{\Delta} t} t'$, then $t'' \uplus t \preceq_{\mathit{\Delta} t} t'' \uplus t'$ and $t'' * t \preceq_{\mathit{\Delta} t} t'' * t'$. ∎

Intuitively, $t \preceq_{\mathit{\Delta} t} t'$ represents a relation where t is not superior to t' ($q(t) \leq q(t')$), but can overtake it by making $\mathit{\Delta} t$ progress further than t' ($t' \preceq t \uplus \mathit{\Delta} t$). The overtaking property implies that this relation is preserved by $\uplus$ and $*$.

Remark 5 The monotonicity of $*$ concerning $\preceq$ does not hold in general. When $t \preceq t'$, each $t'' * t \preceq t'' * t'$, $t'' * t \succeq t'' * t'$ and another case (namely where they are incomparable with respect to $\preceq$) are possible. Of course, whether or not the monotonicity of the sequential composition holds depends on the definition of the composition operators. For instance, let $\preceq_{max}$ be the relation defined using the parallel composition $\uplus_{max}$ of Example 1 and $*$. Then, $*$ is monotonic with respect to $\preceq_{max}$, and $\preceq_{max}$ is characterized as follows:

$$t \preceq_{max} t' \text{ iff } p(t) \leq p(t') \wedge q(t) \leq q(t').$$

The lack of monotonicity of $*$ is one of the main difficulties as regards proving the validity of trust computation. The relation $\preceq_{\mathit{\Delta} t}$ introduced in Lemma 4 is needed to overcome it. ∎

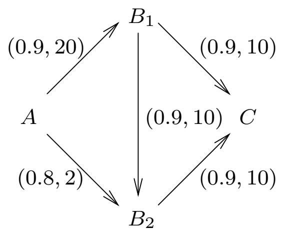

Fig. 3. Sound and Unsound Trust Computation

In the following, we formulate the soundness of a computed trust using the partial order $\preceq$ defined above.

Definition 6 We say a computed trust t concerning B is sound if $t \preceq t_B$. ∎

In the context of this paper, it is generally impossible to totally and completely compute $t_B = \sum_{P \in \mathcal{P} - \{B\}} t_{PB}$. The intuition behind the definition is that t may be partial and inferior, but correct in the sense that it can be the result of a valid computation that does not contain duplicate counts of the basic trusts concerning B and thus treats them adequately as additive values.

Example 7 Suppose basic trusts are defined as in Figure 3. A computed trust $s = t_{AB_1} * t_{B_1C} \uplus t_{AB_1} * t_{B_1B_2} * t_{B_2C} = (0.7695, 20)$ of A concerning C is sound since $(0.7695, 20) \preceq (0.9, 20) = t_C$. So as $s' = t_{AB_2} * t_{B_2C} = t_C$. However, $s \uplus s' = (0.765, 22)$ is not sound as a computed trust concerning C since $(0.765, 22) \not\preceq t_C$. In fact, t_{B_2C} is counted twice here. ∎

However, it is insufficient to consider each computed trust for determining the validity of trust computation. As shown in Example 3, it is possible that the computed trust itself is sound but is overly influenced by a change in a basic trust. We next define the stability of computation as not to occur such a problem. But here is a technical difficulty that procedures for trust computation discussed in this paper are partial (that is, may not output any value) and non-deterministic in general. In the next section, we present a procedure that distributedly computes trusts using a protocol that is non-deterministic with respect to the selection of the request's receivers and the construction of a response. Moreover, we assume that receivers of requests may ignore them or tell a lie. Below we formulate such a procedure as a function from the inputs to the set of possible outputs, and define stability using Hoare's preorder.

Definition 8 A basic trust assignment T (or simply, assignment) is a function from a pair of distinct persons to a trust. $T(A, B)$ denotes the basic trust of A concerning B with respect to T. Instead of $T(A, B)$, we also write t^T_{AB}, or simply t_{AB}, when T is apparent from the context. Moreover, the basic trust assignment obtained by increasing the value t_{EF} of T by $\mathit{\Delta} t$ is denoted by $T \uplus_{EF} \mathit{\Delta} t$. ∎

Definition 9 A trust computation procedure f is a procedure that, given assignment T and $A, B \in \mathcal{P}$ distinct as inputs, outputs a trust on termination. The

set of all possible outputs of f with inputs T, A and B is denoted by $f(T, A, B)$.
∎

Definition 10 A preorder $\sqsubseteq$ on trust sets is defined as follows:

$$\mathcal{T} \sqsubseteq \mathcal{T}' \ \text{ iff } \ \forall t \in \mathcal{T} \ \exists t' \in \mathcal{T}' \ t \preceq t'.$$

Moreover, for a trust set $\mathcal{T}$ and a trust t, we define $\mathcal{T} \uplus t$ as

$$\mathcal{T} \uplus t = \{t' \uplus t \mid t' \in \mathcal{T} \cup \{0\}\}.$$ ∎

Definition 11 We say a trust computation procedure f is stable if it satisfies the following properties for any distinct $A, B \in \mathcal{P}$:

1. $f(T, A, B) \sqsubseteq \{0\}$ if the total basic trust concerning B with respect to T is 0.
2. $f(T \uplus_{EF} \varDelta t, A, B) \sqsubseteq f(T, A, B) \uplus \varDelta t$ for any distinct $E, F \in \mathcal{P}$ and a trust $\varDelta t$.

Intuitively, the second condition means that the computed trust $f(T \uplus_{EF} \varDelta t, A, B)$ is bigger than $f(T, A, B)$ since the assignment for t_{EF} is increased by $\varDelta t$, but the difference is bounded by $\varDelta t$ itself. Roughly speaking, the stability of the trust computation procedure means that the procedure adequately treats all basic trusts as additive values.

Example 12 Suppose that T is an assignment obtained from that in Example 3 by replacing t_{CD} with $(1, 0.5)$, and that $T' = T \uplus_{CD} (1, 0.5)$. Then, let us consider the (deterministic) procedure using the same formula as in the example.

$$\sum_{i=1}^{100} t^T_{AB_i} * t^T_{B_iC} * t^T_{CD} * t^T_{DE_i} * t^T_{E_iF} = (1, 50),$$

$$\begin{aligned} \textstyle\sum_{i=1}^{100} t^{T'}_{AB_i} * t^{T'}_{B_iC} * t^{T'}_{CD} * t^{T'}_{DE_i} * t^{T'}_{E_iF} &= (1, 100) \\ &\not\preceq (1, 50) \uplus (1, 0.5). \end{aligned}$$

Thus, this procedure is not stable. ∎

Lemma 13 If a trust computation procedure is stable, then its output is sound as the computed trust concerning the third input. ∎

Next, we present the assumption concerning a lie mentioned in Section 3.2 in a more general form using $\preceq$. In this paper, we assume that each lie from one person to another in a trust communication is determined by the communicated trust. The function representing the communicated trust containing a lie from B to A is called a lie function of B to A, denoted by L_{AB}. If B holds a (true) trust s and sends it to A, then A actually receives $L_{AB}(s)$. For lie functions, we assume that the following inequation holds:

$$t_{AB} * L_{AB}(s) \preceq s.$$

That is, any lie of B to A can be canceled by the application of "$t_{AB} * \_$". We call this assumption the upper limit assumption on a lie.

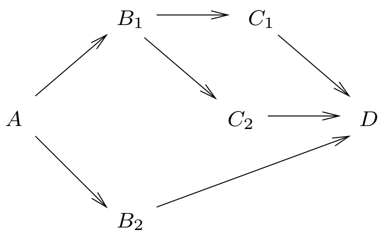

Fig. 4. Graph represented by Linear Term

We do not claim that this assumption is realistic. It is very strong, or rather too idealized. But what we are concerned with here is whether or not soundness and stability are conserved under such a strong and idealized assumption. In the next sections, we present a trust computation with linear terms, which is sound and stable without a lie. Under the limit assumption on a lie, soundness is conserved but its proof is nontrivial, and more surprisingly, there is a counter example for stability.

4.2 Computation with Linear Term

Let us consider a directed graph with people as vertices where each edge goes from a truster to a trustee. We define linear terms to represent computation without duplicate counting. For any distinct $A, B \in \mathcal{P}$ we introduce a constant symbol $\tilde{t}_{AB}$ called a basic trust symbol, and consider terms constructed with the symbols, $\uplus$ and $*$.

Definition 14 Let A, B and C be any distinct vertices. We define an A-B linear term and the graph (a set of directed edges) represented by the term as follows.

1. $\tilde{t}_{AB}$ is an A-B linear term representing the singleton set with the A-B edge as its only member.
2. If A-B linear terms $S_1, \cdots, S_n$ $(n \geq 1)$ represent graphs that share no edges, then $S_1 \uplus \cdots \uplus S_n$ is an A-B linear term representing $S_1 \cup \cdots \cup S_n$.
3. If S is a B-C linear term and $A \in \mathcal{P}$ does not appear in the graph represented by S, then $\tilde{t}_{AB} * S$ is an A-C linear term that represents the graph S increased by the A-B edge. ▮

Example 15 Figure 4 shows the graph represented by a linear term $\tilde{t}_{AB_1} * (\tilde{t}_{B_1C_1} * \tilde{t}_{C_1D} \uplus \tilde{t}_{B_1C_2} * \tilde{t}_{C_2D}) \uplus \tilde{t}_{AB_2} * \tilde{t}_{B_2D}$. There is no linear term representing the graph in Figure 1. In this graph, a linear term can represent, for instance, its path $\tilde{t}_{AB_i} * \tilde{t}_{B_iC} * \tilde{t}_{CD}$. ▮

Let C and D be any distinct persons. The following properties of the A-B linear term S derive directly from the definition.

- The graph represented by S contains the C-D edge iff $\tilde{t}_{CD}$ appears in S.
- (Linearity) $\tilde{t}_{CD}$ appears in S at most once.

Thus, given an A-B linear term S and an assignment T, the trust obtained from S by interpreting each occurrence of $\tilde{t}_{CD}$ as t_{CD}^{T} is called the trust linearly computed by S with respect to T, denoted by $[S]^{T}$.

Lemma 16 For any A-B linear term S and assignment T, $[S]^{T} \preceq t_{B}$. ∎

Note that we cannot employ simple induction on the construction of the term because of the lack of $*$'s monotonicity.

5 Trust Computation Protocol

In this section, we present our protocol for computing trust in a distributed manner. The results in this section depend only on the properties in Lemma 4, associativity, commutativity and zero of operators, and thus are independent of the specific definition of operators.

The basic protocol is a non-deterministic protocol exchanging the following messages:

Request: A pair $\langle C, P\rangle$ of the target C to whom a trust is computed in the session, and the sequence P along with which the request is relayed.

Response: A pair $\langle s, S\rangle$ of a computed trust s, and the linear term S by which s is linearly computed.

For any $A, A' \in \mathcal{P}$ distinct, we assume $t_{AA'} = 0$ if A has never communicated with A'. When A receives a request $\langle C, P\rangle$ from D, he processes it as follows:

1. If $t_{AC} \neq 0$, then A sends himself a response $\langle t_{AC}, \tilde{t}_{AC}\rangle$.
2. Then A non-deterministically chooses $B_1, \cdots, B_n$ satisfying the following three conditions and sends them a request $\langle C, P \cdot A\rangle$:
   - B_i is neither A nor C.
   - B_i does not occur in P.
   - $t_{AB_i} \neq 0$.
3. A waits as long as possible for responses from $B_1, \cdots, B_n$.
4. From among the received responses, A chooses $\langle s_{B'_1 C}, S_{B'_1 C}\rangle, \cdots, \langle s_{B'_k C}, S_{B'_k C}\rangle$ so that $S_{B_1 C}, \cdots, S_{B_n C}$ share no basic trust symbol with each other (if A chooses nothing, the process terminates immediately), and sends the pair $\langle t_{AB'_1} * s_{B'_1 C} \uplus \cdots \uplus t_{AB'_k} * s_{B'_k C}, \tilde{t}_{AB'_1} * S_{B'_1 C} \uplus \cdots \uplus \tilde{t}_{AB'_k} * S_{B'_k C}\rangle$ to D. If $B'_i = A$, then $t_{AB'_i} * s_{B'_i C}$ denotes $s_{B'_i C}$, and $\tilde{t}_{AB'_i} * S_{B'_i C}$ denotes $S_{B'_i C}$.

Here we are assuming that for every response a participant in the protocol can determine the corresponding request. In step 3 we do not have to wait for all responses from $B_1, \cdots, B_n$; the basic idea of this paper is that we cannot totally and completely compute the trusts. If someone wants to initiate a session to compute a trust concerning C, he sends himself a request $\langle C, \lambda\rangle$, where λ denotes the empty sequence.

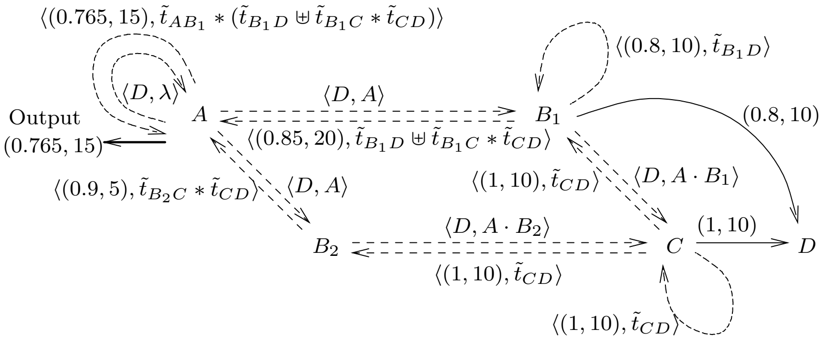
$\langle (0.765, 15), \tilde{t}_{AB_1} * (\tilde{t}_{B_1D} \uplus \tilde{t}_{B_1C} * \tilde{t}_{CD})\rangle$
$\langle (0.8, 10), \tilde{t}_{B_1D}\rangle$
$\langle D, \lambda\rangle$
$\langle D, A\rangle$
Output
$(0.765, 15)$
A
B_1
$(0.8, 10)$
$\langle (0.85, 20), \tilde{t}_{B_1D} \uplus \tilde{t}_{B_1C} * \tilde{t}_{CD}\rangle$
$\langle (0.9, 5), \tilde{t}_{B_2C} * \tilde{t}_{CD}\rangle$
$\langle D, A\rangle$
$\langle (1, 10), \tilde{t}_{CD}\rangle$
$\langle D, A \cdot B_1\rangle$
$\langle D, A \cdot B_2\rangle$
$(1, 10)$
B_2
C
D
$\langle (1, 10), \tilde{t}_{CD}\rangle$
$\langle (1, 10), \tilde{t}_{CD}\rangle$

Fig. 5. Execution of Trust Computation

Lemma 17 Let T be an assignment determined by the basic trusts all persons actually hold. Suppose, in a session with the basic protocol, no participant tells a lie, and A sends a response $\langle s, S\rangle$ answering a request $\langle C, P\rangle$. Then S is an A-C linear term and $s = [S]^T$. ∎

Using the basic protocol, we can define the following trust computation procedure in a straightforward manner. Given inputs T, A and B,

1. the basic trust of each person is determined according to T.[1]
2. A initiates a trust computation session concerning B.
3. If A receives a response, he outputs its first element.

Figure 5 shows an example execution of the procedure $f(T, A, D)$ using the basic protocol. Suppose that the only non-zero basic trusts to D are $t_{B_1D} = (0.9, 10)$ and $t_{CD} = (1, 10)$ represented with solid arrows, and that $t_{B_1C} = (0.9, 30)$, $t_{B_2C} = (0.9, 5)$, $t_{AB_1} = (0.9, 15)$, and $t_{AB_2} = (0.9, 20)$. Assume that the participants do not tell a lie. Requests and responses are represented with dash arrows. The execution of $f(T, A, D)$ proceeds as follows:

- First A sends a request $\langle D, \lambda\rangle$ to himself, and receives it. Then he chooses receivers B_1 and B_2, and sends them $\langle D, A\rangle$. Then he waits for the responses.
- Upon receiving the request from A, B_1 sends himself a response $\langle (0.8, 10), \tilde{t}_{B_1D}\rangle$ since he holds non-zero basic trust concerning D. Then B chooses C to send a request $\langle D, A \cdot B_1\rangle$ to C, then waits for the response.
- Upon receiving the request from B_1, C sends himself a response $\langle (1, 10), \tilde{t}_{CD}\rangle$ since he holds non-zero basic trust concerning D. C chooses no receiver for the request, and thus it is the only response. So he sends response $\langle (1, 10), \tilde{t}_{CD}\rangle$ to B_1.
- B_1 consolidates the two obtained responses, and sends a response $\langle s_{B_1}, S_{B_1}\rangle = \langle (0.85, 20), \tilde{t}_{B_1D} \uplus \tilde{t}_{B_1C} * \tilde{t}_{CD}\rangle$ to A.

[1] We agree that it is unnatural that T determines each person's basic trust. In fact, each person's basic trust is given and the formal input T is determined accordingly.

- On the other hand, upon receiving the request from A, B_2 chooses C, sends a request $\langle D, A \cdot B_2 \rangle$ to C and waits. (He sends nothing to himself since his basic trust concerning D is 0.) C processes the request in the same way as with B_1, and sends the response $\langle (1, 10), \tilde{t}_{CD} \rangle$ to B_2. Upon receiving it, B_2 sends a response $\langle s_{B_2}, S_{B_2} \rangle = \langle (0.9, 5), \tilde{t}_{B_2C} * \tilde{t}_{CD} \rangle$ to A.
- A receives the responses from B_1 and B_2. Their linear terms share the same basic trust symbol $\tilde{t}_{CD}$, so A chooses the response from B_1 and sends a response $\langle s_A, S_A \rangle = \langle (0.765, 15), \tilde{t}_{AB_1} * (\tilde{t}_{B_1D} \uplus \tilde{t}_{B_1C} * \tilde{t}_{CD}) \rangle$ to himself.
- Upon receiving of the response, A outputs $(0.765, 15)$.

Theorem 18 Let f be a trust computation procedure defined using the basic protocol.

1. Assume that, while executing f, the participants in the session tell lies only when they determine the first element of the request within the upper limit assumption on a lie. Then, every trust computed by f is sound.
2. If no protocol participant tells a lie, f is stable. ∎

If a participant lies, the trust computation procedure defined by the basic protocol can be unstable. For instance, let us consider a situation in which B tells a lie when he provides A a trust s_{BC}. Let L_{AB} and L^{+}_{AB} be lie functions when the basic trust of A concerning B is t_{AB} and $t_{AB} \uplus \mathit{\Delta} t$, respectively. Also suppose

$$\begin{aligned}
&t_{AB} = (1, 1),\\
&\mathit{\Delta} t = (0, 1),\\
&s_{BC} = (0.5, 2),\\
&L_{AB}((0.5, 2)) = (0.5, 2),\\
&L^{+}_{AB}((0.5, 2)) = (1, 2).
\end{aligned}$$

Note that in the above setting both L_{AB} and L^{+}_{AB} satisfy the upper limit assumption on a lie, and $L_{AB}((0.5, 2))$ cannot have a larger value with respect to $\preceq$ since $t_{AB} = (1, 1)$. In this case, however,

$$\begin{aligned}
&(t_{AB} \uplus \mathit{\Delta} t) * L^{+}_{AB}(s_{BC}) = (0.5, 2),\\
&t_{AB} * L_{AB}(s_{BC}) \uplus \mathit{\Delta} t = (0.25, 2).
\end{aligned}$$

Thus, the second condition of stability does not hold here.

6 Related Work

The problem of trust computation in a network has been studied in [5, 6]. The authors use the two operators called discounting $\otimes$ and consensus $\oplus$ introduced in [4], which roughly correspond to the sequential and parallel compositions, respectively, in this paper. Two criticisms were presented [14] of their formulation of the discounting:

1. It does not have a natural interpretation in terms of evidence handling.
2. It is not distributive with respect to the consensus, that is, $t \otimes (t' \oplus t'') \neq (t \otimes t') \oplus (t \otimes t'')$.

As regards distribution, we do not think the equality always holds. However, they must be related, and the subjective logic does not provide any generic way to discuss it.

Thus, [14] proposed a reformulation of discounting based on scalar multiplication. The new discounting $\boxtimes$ is defined as $t_{AB} \boxtimes t_{BC} = g(t_{AB}) \cdot t_{BC}$, where $g(x)$ is a non-negative real, and g can be chosen at will, depending on the context. This has a very simple interpretation in evidence space, and satisfies distribution with respect to consensus. But there is a problem regarding the choice of g. For instance, as for the friendship example in Introduction, it seems impossible to choose one discount rate $g(t_{AB})$.

To solve these problems, we separately and directly represent the uncertainty caused by the quantity of evidence concerning t as $q(t)$.

Semi-distribution is weaker than distribution, but has a very natural interpretation in the trust calculation with linear terms, that is, trust information t' and t'' obtained from two distinct sources is more trustworthy than $t' \uplus t''$ from one source. The information order $\preceq$ on trusts enables us to reflect such a casually correct fact in the theory.

The notion of the canonical expression [5, 6] corresponds to that of the linear term in this paper in the sense that these are expressions in which every person-to-person edge appears only once. The authors explain that canonical expressions are necessary since the values of $t \otimes (t' \oplus t'')$ and $(t \otimes t') \oplus (t \otimes t'')$ differ. This is best understood as an independence issue. That is, the consensus operator works properly only for a pair of independent trust information, while $(t \otimes t')$ and $(t \otimes t'')$ are not independent of each other. However, the independence notion is explained very informally in [4].

On the other hand, canonical expressions are unnecessary for the trust calculation in [14] since the distribution of discounting holds there. But as mentioned above, there seem to be some cases where their definition of discounting is invalid.

We do not insist that linear terms are necessary for valid trust computation. The validity we need are soundness and stability, and the utilization of linear terms is a sufficient condition for them.

7 Discussion

For the actual implementation there are some problems to be solved. One is how to define the criterion of quantity; it should be uniform, independent of user preference. A promising candidate for a practically useful criterion is the monetary value. Another problem is how to determine a basic trust by direct and exclusive experiences. Usual pecuniary transactions naturally achieve this by determining the quantity of trust from the monetary value. It would be difficult to determine the trust by the experience that cannot be evaluated in terms of money, or that is shared with a large unspecified number of people. However, this seems to be an intrinsic problem whose solution needs psychological and sociological findings, beyond the scope of this paper.

8 Conclusion

We formulated a trust metric using a pair of quality and quantity, and presented the algebraic properties of its composition operations. Moreover, we defined the validity of the trust computation in terms of the operations, and thus we do not need probabilistic assumptions.

We can consider variations of trust formulations and composition definitions including a many-value extension of quality. We are also interested in a relaxation of the stability condition so that the basic protocol can be satisfied. Moreover, stable trust computation is closely related to the maximum flow problem. Extensions and clarifications in these directions constitute future work.

An evaluation is needed to justify the validity and efficacy of our approach. Building a prototype would be helpful for showing the advantages of the approach, e.g., the robustness of our metric against attacks in trust networks. These topic will also be considered future work.

References

1. R. Demolombe. Reasoning about trust: A formal logical framework. In *iTrust 2004*, pages 291–303, 2004.
2. R. Demolombe. Transitivity and propagation of trust in information sources: An analysis in modal logic. In *CLIMA 2011*, pages 13–28, 2011.
3. J. Huang and D. M. Nicol. A calculus of trust and its application to PKI and identity management. In *IDtrust 2009*, pages 23–37, 2009.
4. A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.
5. A. Jøsang, E. Gray, and M. Kinateder. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems*, 4(2):139–161, 2006.
6. A. Jøsang, R. Hayward, and S. Pope. Trust network analysis with subjective logic. In *29th Australasian Computer Science Conference*, pages 85–94, 2006.
7. C.-J. Liau. Belief, information acquisition, and trust in multi-agent systems – a modal logic formulation. *Artif. Intell.*, 149(1):31–60, 2003.
8. E. Lorini and R. Demolombe. From binary trust to graded trust in information sources: A logical perspective. In R. Falcone, K. S. Barber, J. Sabater-Mir, and M. P. Singh, editors, *TRUST 2008*, volume 5396 of *LNAI*, pages 205–225, 2008.
9. E. Lorini and R. Demolombe. From trust in information sources to trust in communication systems: An analysis in modal logic. In *KRAMAS 2008*, pages 81–98, 2008.
10. T. Muller, Y. Liu, and J. Zhang. The fallacy of endogenous discounting of trust recommendations. In *AAMAS '15*, pages 563–572, 2015.
11. M. P. Singh. Trust as dependence: a logical approach. In L. Sonenberg, P. Stone, K. Tumer, and P. Yolum, editors, *AAMAS 2011*, pages 863–870, 2011.
12. G. Theodorakopoulos and J. S. Baras. Trust evaluation in ad-hoc networks. In *WiSe 2004*, pages 1–10, 2004.
13. P. Venkat Rangan. An axiomatic basis of trust in distributed systems. In *IEEE S&P*, pages 204–211, 1988.
14. B. Škorić, S. de Hoogh, and N. Zannone. Flow-based reputation with uncertainty: Evidence-based subjective logic. *International Journal of Information Security*, 15(4):381—402, 2015.