Math. Struct. in Comp. Science (2017), *vol.* 27, *pp.* 215–233. © Cambridge University Press 2015
doi:10.1017/S0960129515000092 First published online 1 June 2015

On merging two trust-networks in one with bipolar preferences

STEFANO BISTARELLI[†§] and FRANCESCO SANTINI[‡¶]

[†]*Dipartimento di Matematica a Informatica, Università di Perugia, Italy*
Email: bista@dmi.unipg.it
[‡]*Istituto di Informatica e Telematica, IIT-CNR, Pisa, Italy*
Email: francesco.santini@iit.cnr.it

Received 20 January 2014; revised 24 February 2015

In this paper, we study weighted trust-networks (but also unweighted), where each edge is associated with either a positive or a negative score. Hence, we consider a distrust relationship as well, allowing a user to rate poor experiences with other individuals in his web of acquaintances. We propose an algorithm to compose two of such networks in a single one, in order to merge the knowledge obtained in two different communities of individuals (possibly partially-overlapping), through two different trust management-systems. Our algorithm is based on semiring algebraic-structures, in order to have a parametric computational-framework. Such composition can be adopted whenever two trust-based communities (with the same scope) need to be amalgamated: for instance, two competitor-companies that need to unify the trust-based knowledge on their (sub-) suppliers.

1. Introduction and motivations

In the last few years, interaction has considerably leaned towards indirect relationships, through the use of social networks, e-commerce sites and e-mail communications, where trust plays a crucial role. In most of the cases, this kind of electronic communication occurs between entities/people who do not actually know each other, and for which it is not possible to establish a relationship of trust that has a solid foundation. In order to better capture this broad concept of trust, in this paper we refer to the definition in MckNight and Chervany (1996): *'Trust is the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible*.' This definition is indeed general, but it implicitly manages to capture the notions of reliability, dependence, utility and risk attitude, which are all related to trust.

However, our goal is also to capture distrust at the same time: distrust is an important concept to consider in propagation, and it is totally separate from trust (Golbeck 2008, Ch. 6.4). It can be defined as the mirror-image opposite of trust (McKnight and Chervany 2001). Distrusting intentions means that one is against being willing to depend, or intends not to depend, on the other party. No willingness to depend means that one is against

[§] Partially supported by Istituto di Informatica e Telematica, CNR-Pisa, Italy.
[¶] Partially supported by MIUR-PRIN 'Security Horizons'.

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

making oneself vulnerable to the other party by relying on the other party: if *a* distrusts *b*, it does not mean that *a* barely believes that it can rely on *b* (that is, *a* mildly trusts *b*), buy it just believes it cannot rely on *b*. Moreover, Dimoka shows that trust and distrust activate different brain areas, implying that they are distinct constructs that are associated with different neurological processes (Dimoka 2010). Hence, in sum, distrust is not simply a lack of trust.

A *Trust Network* (TN), or *Web of Trust*, represents a social network that links some individuals according to the intended scope of the considered trust-relationship (i.e., named *trust scope*), e.g., 'to be a good car-mechanic' if a user is looking for a mechanic to repair his broken-down car. To be more precise, TNs actually are directed graphs: nodes represent individual users, and, if we observe a relationship as *'individual a trusts individual b,'* this results in an edge directed from node *a* to node *b*. Moreover, edge transitivity is an explicit component in those trust systems where trust is a transitive relationship, and it allows each user to expand his limited and personal view by also taking into account the referrals from acquaintances. As well, distrust can be represented with a different set of directed edges concerning the same collection of individuals; the edges need to be partitioned in two since, as explained in the previous paragraph, trust and distrust are related but separate concepts. Following the approach in Ch. 6.4.1 of Golbeck (2008), we consider distrust as a non-transitive relationship: if *a* distrusts *b*, then *a* cannot judge any knowledge acquired through *b*. Note that *Balance Theory* (Heider 2013) promotes two negative attitudes into a positive one instead, like 'the enemy of my enemy is my friend.'

Trust-based interactions between two individuals usually produce a score that reflects the relying party's subjective view of an entity's trustworthiness. This is the reason why in this work we adopt weighted TNs, where each edge is associated with a trust (or distrust) score that corresponds to the degree up to which *a* is willing to trust (or distrust) *b*. Note that the framework presented in this paper is parametric to the chosen bipolar structure (see Section 2), and, by selecting the Boolean semiring, it is possible to manage trust and distrust also in unweighted networks.

Each TN maps the trust-related knowledge of a community with a given specific scope. However, sometimes the need for a composition between two TNs can arise. Consider, for instance, when two companies merge together into a single one: their two different TNs, representing their respective knowledge about the trustworthiness of their suppliers (for example) has to be merged as well. In general, whenever two different (but identical in terms of trust scope) communities need to meet and join, this fusion of knowledge has to be accomplished in a proper way, also considering the conflicting opinions that are sparked from different experience. In this paper, we provide a composition algorithm that exploits general algebraic-structures to represent both trust and distrust metrics. The algorithm is parametric with bipolar semiring-based structures, which can consequently be instantiated according to the specific problem. Having an algebraic structure leads to a formal representation of values and operators, and opens to some expected properties during and after the composition of TNs (see Section 4). Note that this kind of structures has been already used in Bistarelli and Santini (2008) and Theodorakopoulos and Baras (2004) to compose trust along a (transitive) trust chain, but never to model distrust or merge TNs so far.

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

The following of the paper represents a revised and extended version of Bistarelli and Santini (2014), and it is organized as it follows: in Section 2 we summarize the background on bipolar preference-structures. Afterwards, Section 3 defines bipolar TNs and how to use formal operators to deal with trust transitivity, it presents and describes the algorithm pseudocode, and explains its complexity. Then, in Section 4 we show some formal properties, as a normal form for TNs. Section 5 shows a composition example for weighted and unweighted TNs. In Section 6, we report the related literature and we discuss the differences with respect to our choices. Section 7 wraps up the paper and suggests ideas about future work.

2. 'À-la-bipolar' preference structures

To handle both positive and negative preferences at the same time, we assemble two distinct algebraic structures into a single *bipolar preference structure* (Bistarelli *et al.* 2011). These two structures are $\langle P, +_P, \times_P, \bot_P, \top_P \rangle$ and $\langle N, +_N, \times_N, \bot_N, \top_N \rangle$.

The structure we use to model negative preferences is a c-semiring (or simply semiring in the following) as originally described in Bistarelli *et al.* (1997) and Bistarelli (2004). A semiring is a tuple $\langle N, +_N, \times_N, \bot_N, \top_N \rangle$ where N is a set of negative preferences with two special elements $\bot, \top$ (respectively, the bottom and top elements of N) and with two binary operations $+_N$ and $\times_N$ that satisfy certain properties: $+_N$ is defined over (possibly infinite) sets of elements of N and is commutative, associative, idempotent, it is closed and $\bot_N$ is its unit element and $\top_N$ is its absorbing element; $\times_N$ is closed, associative, commutative, distributes over $+_N$, $\top_N$ is its unit element, and $\bot_N$ is its absorbing element (for the exhaustive definition, please refer to Bistarelli *et al.* (1997)). The $+_N$ operation defines a partial order $\leqslant_{+_N}$ over N such that $a \leqslant_{+_N} b$ iff $a +_N b = b$; we say that $a \leqslant_{+_N} b$ if b represents a value *better* than a. Other properties related to the two operations are that $+_N$ and $\times_N$ are monotone on $\leqslant_{+_N}$, $\bot_N$ is its min and $\top_N$ its max, $\langle N, \leqslant_{+_N} \rangle$ is a complete lattice and $+_N$ is its lub. An example of a semiring instantiation is $\langle \mathbb{R}^+ \cup \{+\infty\}, min, sum, +\infty, 0 \rangle$, where *sum* is the arithmetic addition. Both $+$ and $\times$ can be easily extended to n-ary operations on n elements in the set of preferences, due to associativity and commutativity.

When dealing with positive preferences instead, classical bipolar structures (Bistarelli *et al.* 2011) require that combination ($\times_P$) leads to better preferences. However, in the following of the paper, we will use an 'à-la-bipolar' version of this dual structure. It will be nevertheless called bipolar as well for the sake of easiness, but also because it respects all its design principles, except for the property that $\times_p$ leads to better preferences. In fact, in our approach we use a semiring even to define positive preference-system, i.e., $\langle P, +_P, \times_P, \bot_P, \top_P \rangle$ is a semiring. The reason behind is that, when propagating trust along a path, it is better to have a non-increasing function, since a referral cannot be better than a direct experience between two individuals: hence, trust must not increase along a path, but it can only decrease.

Thus, a bipolar preference-structure is a tuple $\mathcal{B} = \langle N, P, +, \times, \bot, \square, \top \rangle$ that is assembled with a semiring $\langle N, +_N, \times_N, \bot_N, \top_N \rangle$ to model negative preferences, and a positive preference-structure $\langle P, +_P, \times_P, \bot_P, \top_P \rangle$. $+ : (N \cup P)^2 \longrightarrow (N \cup P)$ is an operator

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

s.t. $a_N + a_P = a_P$, $\forall a_N \in N$ and $a_P \in P$; $+$ induces a partial ordering on $N \cup P$: $\forall a, b \in (N \cup P), a \leqslant b$ iff $a + b = b$. The operator $\times : (N \cup P)^2 \longrightarrow (N \cup P)$ is commutative (i.e., $a \times b = b \times a$, with $a, b \in (P \cup N)$) and monotone (i.e., if $a \leqslant_+ b$, then $(a \times c) \leqslant_+ (b \times c)$, with $a, b, c \in (P \cup N)$). The preference value $\square$ is used to model an *indifference* score, i.e., neither positive nor negative, and to technically join the preference and negative structures through a single preference score (Bistarelli *et al.* 2011).

Given the way the ordering is induced by $+$ on $N \cup P$, we have $\bot \leqslant_+ \square \leqslant_+ \top$. Notice that, although all positive preferences are strictly above negative preferences, this framework does not prevent from using the same scale, or partially overlapping scales, to represent positive and negative preferences (Bistarelli *et al.* 2011). A bipolar preference structure allows us to have different ways to model and reason about positive and negative preferences. In fact, we can have different lattices $(P, \leqslant_{+_P})$ and $(N, \leqslant_{+_N})$. This is common in real-life problems, where negative and positive statements are not necessarily expressed using the same granularity. For example, we could be satisfied with just two levels of negative preferences, while requiring several levels of positive preferences.

In the following of the paper, our reference bipolar-structure will be $\mathcal{B} = \langle [-1, 0], [0, 1], max, sum, -1, 0, 1\rangle$, where the distrust structure corresponds to $\langle [-1, 0], max, min, -1, 0\rangle$, and the trust structure to $\langle [0, 1], max, min, 0, 1\rangle$. We remind that a bipolar structure cannot in general be flattened to a single semiring because, (i) we need a compensation operator between positive and negative preferences, (ii) we need to keep these two preference systems as separate, because they may be defined over different value-intervals, or they may need different definitions of the same operator.

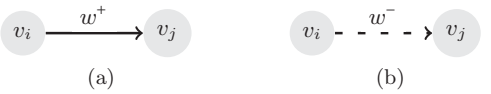
3. Composing two trust networks with bipolar preferences

We consider a bipolar TN as represented by a directed graph $G = (V, E = E_t \cup E_d)$. To represent when $v_i \in V$ trusts $v_j \in V$, we graphically use a direct-edge labelled with a trust weight $w^+ \in P$ (see Figure 1a). On the other side, a distrust relationship is represented by a dashed edge between v_i and v_j, labelled with a distrust weight $w^- \in N$ (see Figure 1b). Formally, a function W is used to retrieve the trust score of an edge, i.e., either $W(v_i, v_j) = w^+$ or $W(v_i, v_j) = w^-$. Moreover, we have that, if $(v_i, v_j) \in E_t$, then $W(v_i, v_j) = w^+$, otherwise $(v_i, v_j) \in E_d$ leads to $W(v_i, v_j) = w^-$: E_t and E_d collect trust and distrust edges, respectively. To model the two dual metrics we adopt the bipolar structure $\mathcal{B}$ presented in Section 2, i.e., $\langle N, P, +, \times, \bot, \square, \top\rangle$, which is assembled with a semiring $\langle N, +_N, \times_N, \bot_N, \top_N\rangle$ to model distrust, and a semiring $\langle P, +_P, \times_P, \bot_P, \top_P\rangle$ to model trust. Formally, we have the following:

Definition 3.1 (Bipolar trust network). A *Bipolar Trust Network* is a directed graph $G = \langle V, E = E_t \cup E_d, W, \mathcal{B}\rangle$, where V is a set of vertices, E a set of edges, $W : E \to (N \cup P)$ is a total function that assigns a weight to each edge in E (E_t/E_d collects trust/distrust edges respectively), and $\mathcal{B}$ is a bipolar preference-structure $\langle N, P, +, \times, \bot, \square, \top\rangle$.

Therefore, $w^+ \in P$ can be composed by using $\times_P$, while $+_P$ defines a partial order over the trust levels in P. Finally, $\top_P$ and $\bot_P$, respectively represent the best and worst trust-level in P. Specular considerations hold for the negative metric, e.g., the levels $w^- \in N$.

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press



(a) (b)

Fig. 1. Node v_i trusts (a), distrusts (b) node v_j with weight $w^+ \in P$ (a), $w^- \in N$ (b).

The $\times$ operator of $\mathcal{B}$ will be used to solve possible conflicts that arise after merging two TNs. Three different possible instantiations of $\times$ in a crisp (i.e., unweighted) case can be: $Trust \times Distrust = Trust$, $Trust \times Distrust = Distrust$, or $Trust \times Distrust = \Box$. Finally, the $+$ operator of $\mathcal{B}$ is used to sequentially compose the two orderings given by $+_N$ and $+_P$. The $\Box$ value, through which the positive and negative preference structures are joint together, can represent either an *indifference* score (i.e., neither trust nor distrust), or an *ambivalence* score (i.e., trust and distrust at the same time). In this paper, we opt for indifference: we safely remove both the trust and distrust edges between the same two nodes when the $\times$-composition of their weights is equal to $\Box$. Otherwise, in ambivalence we can choose to leave a trust edge with weight $\Box$, thus keeping the edge to still allow transitivity through it.

In the following, with respect to a bipolar TN, we write $v_i \rightarrow v_j$ when there is a trust edge between v_i and v_j, and $v_i \dashrightarrow v_j$ when there is a distrust edge between them.

A first result we need to obtain on weighted TNs is to apply a *transitive closure* step on trust and distrust edges: after it, we are able to state if any node is reachable from another, i.e., if a given node trusts/distrusts any other node in the same network. In Proposition 3.2, we define trust transitivity, which is needed to understand how to perform such closure step.

Definition 3.2 (trust/distrust transitivity). Given a bipolar TN, $G = \langle V, E = E_t \cup E_d, W, \mathcal{B} \rangle$, a node $v \in V$ always fully trusts itself (these $\top$-valued edges are not considered). Transitivity closure is applicable only if the first edge along a path is a trust edge: given $v_i, v_j, v_k \in V$, and $(v_i, v_j), (v_j, v_k) \in E$, if $v_i \rightarrow v_j$ and $v_j \rightarrow v_k$ (or $v_j \dashrightarrow v_k$), then we may close transitivity by adding $v_i \rightarrow v_k$ (or $v_i \dashrightarrow v_k$, in case $v_j \dashrightarrow v_k$). Otherwise, if $v_i \dashrightarrow v_j$, then transitivity cannot be closed between v_i and v_k, either if $v_j \rightarrow v_k$ or if $v_j \dashrightarrow v_k$.

The reason is that, while the trust relationship is transitive (as in Jøsang *et al.* (2006b) and Bistarelli and Santini (2008)), clearly distrust is not transitive (Golbeck 2008, Ch. 6.4.1). If Alice directly trusts Bob, then Alice trusts Carol as well, if Carol is trusted by Bob. However, if Alice does not trust Bob, positive or negative referrals through Bob are not considered. In some sense, trust propagation is interrupted when a distrust edge is met along a path.

In Propositions 3.1 and 3.2 we show in detail how to extend the trust relationship along a two-edge path.

Proposition 3.1. If the first edge along a path has an associated trust-value w_1^+, and the successive edge has a trust value w_2^+, then the derived edge has a trust value $w_1^+ \times_P w_2^+$, as represented in Figure 2.

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

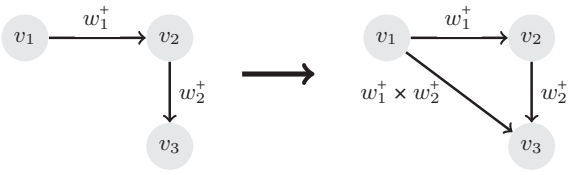


Fig. 2. Transitivity (trust).

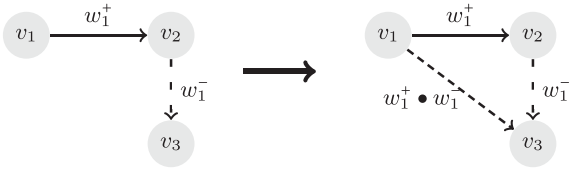


Fig. 3. Transitivity (trust/distrust). The result of $\bullet$ determines how much v_1 distrusts v_2.

For instance, if we instantiate $\times_P$ by choosing the minimum weight of all the edges, we have:

$$w_1^+ \times_P w_2^+ = min(w_1^+, w_2^+) = \begin{Bmatrix} if\ w_1^+ \leqslant w_2^+\ then\ w_1^+ \\ if\ w_1^+ > w_2^+\ then\ w_2^+ \end{Bmatrix}$$

Proposition 3.2. If the first edge along a path has an associated trust-value w_1^+, and the successive edge has a distrust value w_1^-, then the derived edge has a distrust value $w_1^+ \bullet w_1^-$, as represented in Figure 3.

The $\bullet$ operator is used to compose trust and distrust along the same path. It can be instantiated to different operators: in a unweighted-network configuration, it classically extends the transitivity notion that can be found in the literature (e.g., Guha *et al.* (2004) and Thirunarayan and Verma (2008)): (*a trusts b*) $\bullet$ (*b distrusts c*) $\Longrightarrow$ *a distrusts c*. In the weighted case, in $w_r = w^+ \bullet w^-$ the result depends on how $\bullet$ is instantiated. $\bullet$ can be defined as follows:

Definition 3.3. Given a bipolar preference structure $\langle N, P, +, \times, \bot, \Box, \top \rangle$, $\bullet$ is a function $\bullet : (P \times N) \to N$. Moreover,

i. if $p \in P$, $n \in N$, $p \bullet n = n'$ (with $n' \in N$), we have $n' \geqslant n$.
ii. if $p', p'' \in P$, and $p' \geqslant p''$, then $p' \bullet n \leqslant p'' \bullet n$.

Property i in Definition 3.3 means that distrust cannot get worse through referrals, with respect to direct experience. Property ii in Definition 3.3 means that, the more one trusts a distrust referral, the more it directly distrusts that final individual. Note that $\bullet$ is different from the compensation operator of a bipolar preference (i.e., $\times$): $\bullet$ is used to compensate trust and distrust along the same path, while $\times$ is used to compensate between a trust and a distrust edge in parallel between two nodes. The difference is also marked by the different domain and codomain of the two functions: $\times : (N \cup P)^2 \longrightarrow (N \cup P)$.

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

3.1. *The algorithm*

In Algorithm 1, we summarize the pseudocode to compose two TNs. Intuitively, the final network at the end of the algorithm has all trust and distrust relationships between any two nodes as elicited (considering trust as transitive). Therefore, all hidden information is extracted and trust/distrust conflicts between the same two nodes are solved into one (trust) or another (distrust) case. Such new information needs to be extracted and made consistent because the ultimate goal is to merge two distinct trust-focused pieces of knowledge, which can be partially-overlapping.

Algorithm 1 Composition of two Trust Networks (Part 1)

Require: $(G_1 = \langle V_1, E_1, W_1, \mathcal{B}\rangle, G_2 = \langle V_2, E_2, W_2, \mathcal{B}\rangle)$ *with* $\mathcal{B} = \langle N, P, +, \times, \bot, \square, \top\rangle$

1: RemoveConflicts$_{\mathcal{B}}(G_1)$
2: RemoveConflicts$_{\mathcal{B}}(G_2)$
3: G_M =DisjointUnion(G_1, G_2)
4: Floyd–Warshall$_{\mathcal{B}}(\langle V_M, E_t \subseteq E_M, W_t\rangle)$
5: Distrust-Closure$_{\mathcal{B}}(V_M, E_M, W_M)$
6: RemoveConflicts$_{\mathcal{B}}(V_M, E_M, W_M)$

Ensure: $G_M = \langle V_M, E_M, W_M = W_t \cup W_d, , \mathcal{B}\rangle$ is a conflict-free composition of G_1 and G_2

The input of the algorithm consists in two $\mathcal{B}$-bipolar TNs; $\mathcal{B}$ is also the bipolar structure under which the composition has to be accomplished. $G_1 = \langle V_1, E_1, W_1, \mathcal{B}\rangle$ and $G_2 = \langle V_2, E_2, W_2, \mathcal{B}\rangle$ are the two input TNs: V is the set of nodes ($V_1 \cap V_2$ may be different from $\varnothing$), E is the set of edges, and W is the function that associates a trust score with each edge. $W(v_i, v_j) = \bot$ if there is no direct edge connecting node v_i to node v_j, or $W(v_i, v_j) \in (P \cup N \backslash \{\bot\})$ otherwise.

As a first step, we need to remove all the possible conflicts that are present in both G_1 and G_2 (lines 1 and 2 in Algorithm 1) in order to remove inconsistencies. A conflict in a bipolar TN is defined as follows:

Definition 3.4 (trust/distrust conflict). Given a TN, $G = \langle V, E, W, \mathcal{B}\rangle$, we have a conflict if $\exists v_i, v_j \in V . v_i \rightarrow v_j \in E \wedge v_i \dashrightarrow v_j \in E$.

In Algorithm 2, we describe the *Remove-Conflicts*$_{\mathcal{B}}$ function used to remove all the conflicts from a bipolar TN. To accomplish conflict removal, we take advantage of the compensation-operator $\times$. Given the weight w^+ of a trust edge, and the weight w^- of a distrust edge in conflict with it, if $w^+ \times w^- = w \in P$ we remove the distrust edge from our conflict, and we leave a trust edge with weight w. Similarly, if $w^+ \times w^- = w \in N$ we remove the trust edge from the conflict, and we leave the distrust edge with weight w. The third and last possible case corresponds to $w^+ \times w^- = \square$: in this case we remove both the edges between two nodes, since, in this paper, we interpret it as an indifference score. In Algorithm 2, we scan all the conflicts in the loop at line 8, then we compute the compensation weight by using the $\times$ operator at line 9, and we update W_t or W_d depending if $t \in P$ or $t \in N$ (lines 10–14). Finally, from lines 15 to 23 we update E and W, by removing one of the two edges in conflict (or both, in case of $\square$), and thus removing the conflict.

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

Algorithm 2 Composition of two Trust Networks (Part 2)

```
 7: function REMOVE-CONFLICTS_B(G = ⟨V, E = E_t ⊎ E_d, W = ⟨W_t, W_d⟩⟩)
 8:     for all ((v_i → v_j ∈ E_t) ∧ (v_i ⇢ v_j ∈ E_d)) do
 9:         t = W_t[v_i, v_j] × W_d[v_i, v_j]
10:         if (t ∈ P) then
11:             W_t[v_i, v_j] = t
12:         else                                                    ▷ t ∈ N
13:             W_d[v_i, v_j] = t
14:         end if
15:         if (t = □) then
16:             E = E \ {(v_i → v_j), (v_i ⇢ v_j)}
17:             W_t[v_i, v_j] = ⊥
18:             W_d[v_i, v_j] = ⊥
19:         else if (t ∈ P) then
20:             E = E \ {(v_i ⇢ v_j)}, W_d[v_i, v_j] = ⊥
21:         else                                                    ▷ t ∈ N
22:             E = E \ {(v_i → v_j)}, W_t[v_i, v_j] = ⊥
23:         end if
24:     end for
25: end function
```

The second step of the composition algorithm consists in a first raw merge of G_1 and G_2; this is invoked at line 3 in Algorithm 1, and fully implemented in Algorithm 3. At lines 27–30, a first graph is obtained by adding the two sets of nodes together, and by obtaining a multi-set of their two sets of edges; moreover, we assemble two matrices respectively representing a weight function of trust edges (i.e., W_t), and a weight function of distrust edges (i.e., W_d). For all the trust or distrust edges repeated twice in the multi-set E_M (lines 31 and 32), we make a single edge out of them, with a trust/distrust weight equal to the application of $\times_P/\times_N$ to two edge weights (lines 33–37); W_t and W_d are updated in accordance. Afterwards, one of the two edges is removed from the graph since it is redundant (line 38). All the other edges appearing only once in E_M are left with their original trust (lines 39–44) or distrust (lines 45–51) weight instead. The first version of the final graph G_M is then assembled at line 53, and returned as the result of this function.

The first core step of the algorithm is the transitive closure of trust paths: the score of a trust path between two nodes is defined by applying $\times_P$ to all the positive scores of its trust edges, as summarized in Definition 3.5.

Definition 3.5 (trust score of a path). Given a bipolar TN, $G = \langle V, E = E_t \cup E_d, W, \mathcal{B}\rangle$, the trust score associated with a path between two nodes $v_i, v_j \in V$, defined by the set of edges $H^+_{v_i,v_j} = v_i \rightarrow^* v_j$, is computed as $W(H^+_{v_i,v_j}) = (w^+_1 \times_P w^+_2 \times_P \ldots \times_P w^+_k)$, where $\forall e \in H^+_{v_i,v_j}, e \in E_t \wedge W(e) \in \{w^+_1, w^+_2, \ldots, w^+_k\}$, where $k = |H^+_{v_i,v_j}|$.

Now we can formally define the transitive-closure step of trust paths. For any relationship R, its transitive closure R^* always exists; moreover, we have that $R \subseteq R^*$.

Definition 3.6 (transitive closure of trust paths). Let $G = \langle V, E = E_t \cup E_d, W, \mathcal{B}\rangle$ be a bipolar TN. The reflexive transitive closure of trust is the relationship $E^* = \{v_i \rightarrow v_j : v_i, v_j \in V \wedge v_i \rightarrow^* v_j \subseteq E_t\}$, where $v_i \rightarrow^* v_j$ requires the reachability of v_j from v_i by using only the edges in E_t. Each new edge $v_i \rightarrow v_j$ added to E_t is associated with a trust score

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

Algorithm 3 Composition of two Trust Networks (Part 3)

26: **function** DISJOINTUNION($(G_1 = \langle V_1, E_1, W_1, \mathcal{B}\rangle,\ G_2 = \langle V_2, E_2, W_2, \mathcal{B}\rangle)$ *with* $\mathcal{B} = \langle N, P, +, \times, \bot, \Box, \top\rangle$ *and* G_1, G_2 *are conflict−free*)
27: $V_M = V_1 \cup V_2$
28: $E_M = E_1 \uplus E_2$
29: $W_t = [1..|V_M|]$
30: $W_d = [1..|V_M|]$
31: **for all** $((v_i, v_j) \in E_M)$ **do**
32: **if** $((v_i, v_j) \in (E_1 \cap E_2))$ **then**
33: **if** $(v_i \to v_j)$ **then**
34: $W_t[v_i, v_j] = W_1(v_i, v_j) \times_P W_2(v_i, v_j)$
35: **else** ▷ $v_i \dashrightarrow v_j$
36: $W_d[v_i, v_j] = W_1(v_i, v_j) \times_N W_2(v_i, v_j)$
37: **end if**
38: $E_M = E_M \backslash \{(v_i, v_j)\}$ ▷ Removing one repeated edge
39: **else if** $(v_i \to v_j)$ **then**
40: **if** $(v_i \to v_j \in E_1)$ **then**
41: $W_t[v_i, v_j] = W_1(v_i, v_j)$
42: **else** ▷ $v_i \to v_j \in E_2$
43: $W_t[v_i, v_j] = W_2(v_i, v_j)$
44: **end if**
45: **else** ▷ $v_i \dashrightarrow v_j \in E$
46: **if** $(v_i \dashrightarrow v_j \in E_1)$ **then**
47: $W_d[v_i, v_j] = W_1(v_i, v_j)$
48: **else** ▷ $v_i \dashrightarrow v_j \in E_2$
49: $W_d[v_i, v_j] = W_2(v_i, v_j)$
50: **end if**
51: **end if**
52: **end for**
53: **return** $G_M = \langle V_M, E_M, W_M = \langle W_t, W_d\rangle, \mathcal{B}\rangle$
54: **end function**

equal to $+_P(W(H_1^+), W(H_2^+), \ldots, W(H_k^+)) \in P$, where $+_P$ is the prefixed $+_P$ of $\mathcal{B}$, and $H_1^+, H_2^+, \ldots, H_k^+$ are all the possible trust-paths between v_i and v_j. After a transitive-closure step, the new obtained graph is $G^* = (V, E' = E_t^* \cup E_d, W')$.

The *Floyd–Warshall*$_{\mathcal{B}}$ procedure is invoked at line 4 in Algorithm 1, and its pseudocode is shown in Algorithm 4 (lines 55–66). It has been modified, with respect to the original Floyd–Warshall algorithm in Floyd (1962), in order to consider the two parametric operators of a bipolar structure $\mathcal{B}$ (e.g., $+$ and $\times$): instead of computing the shortest path (i.e., *min*/*sum*) between any two nodes in the graph as in Floyd (1962), we compute the least upper bound of all path-costs (the partial order is given by $+_P$) by composing the cost of its edges using $\times_P$. If the preference order is total instead of strictly partial, we compute the best (dominating) path. At each iteration, the algorithm computes the best path from v_i to v_j by considering all the paths traversing v_k. The score of the best trust-path is computed at line 59. A new edge (if missing, see line 60) is added to E at line 61.

As stated in Section 3, only trust is transitive, and then it can be exposed to a 'real' transitivity-closure step. Nevertheless, even distrust paths need a closure step, as stated by Proposition 3.2, and graphically represented in Figure 3. However, this closure is simpler

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

Algorithm 4 Composition of two Trust Networks (Part 4)

```
55: function FLOYD–WARSHALL_B(V, E, W)
56:     for (k from 1 to |V|) do
57:         for (i from 1 to |V|) do
58:             for (j from 1 to |V|) do
59:                 W[v_i, v_j] = W[v_i, v_j] +_P (W[v_i, v_k] ×_P W[v_k, v_j])
60:                 if ((v_i → v_j ∉ E) ∧ (W[i, j] ≠ ⊥)) then
61:                     E = E ∪ (v_i → v_j)
62:                 end if
63:             end for
64:         end for
65:     end for
66: end function
67:
68: function DISTRUST-CLOSURE_B(G = ⟨V, E = E_t ⊎ E_d, W = ⟨W_t, W_d⟩⟩)
69:     for (v_k ⇢ v_j ∈ E) do
70:         for (v_i → v_k ∈ E) do
71:             t = W_t[v_i, v_k] • W_d[v_k, v_j]
72:             if (v_i ⇢ v_j ∈ E) then
73:                 W_d[v_i, v_j] = W_d[v_i, v_j] +_N t
74:             else                                          ▷ v_i ⇢ v_j ∉ E
75:                 W_d[v_i, v_j] = t
76:                 E = E ∪ {v_i ⇢ v_j}
77:             end if
78:         end for
79:     end for
80: end function
```

than trust transitivity, since we only need to check paths as $v_i \rightarrow v_k \dashrightarrow v_j$, i.e., only two-hop paths, where we have a trust and a distrust edge in sequence.

Definition 3.7 (distrust score of a path). Given a bipolar TN, $G = \langle V, E = E_t \cup E_d, W, \mathcal{B} \rangle$, the trust/distrust score associated with a path $v_i \rightarrow v_k \dashrightarrow v_j \in V$ between two nodes v_i, v_j, is computed as $w_1^+ \bullet w_1^- = w^-$, where $W(v_i, v_k) = w_1^+$, and $W(v_k, v_j) = w_1^-$. $H^-_{v_i,v_j}$ is a distrust path with cost w^-.

Distrust closure is implemented by the *Distrust-Closure* function (lines 68–80 in Algorithm 4). A formal definition of distrust closure is given in Definition 3.8 instead.

Definition 3.8 (distrust closure). Given a bipolar TN, $G = \langle V, E = E_t \cup E_d, W, \mathcal{B} \rangle$, where trust is already transitively closed (see Definition 3.6), to close the distrust relationship we add $v_i \dashrightarrow v_j$ to E_d, $\forall v_i, v_j, v_k \in V$ such that $v_i \rightarrow v_k$ and $v_k \dashrightarrow v_j$. The cost of such distrust edge is $t = W(v_i, v_k) \bullet W(v_k, v_j)$.

The *Distrust-Closure*$_{\mathcal{B}}$ function is called at line 5 in Algorithm 1, while the new compensated score is computed at line 71 in Algorithm 4. Each new added-edge has to be correctly merged with the direct distrust-edge $v_i \dashrightarrow v_j$ if already in E_d, in order to have at most one distrust edge between any two nodes. This is procedurally shown at lines 72–77 in Algorithm 4.

After trust/distrust transitivity closure, conflicts in G_M need to be removed (line 6 in Algorithm 1), since this step may have added new edges in conflict. In the example in

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

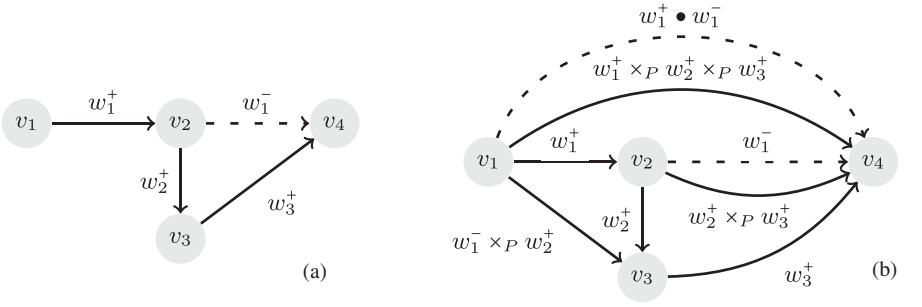


Fig. 4. TN (b) shows TN (a) after trust/distrust transitivity closure: this leads to two new conflicts $v_1 \rightarrow v_4 / v_1 \dashrightarrow v_4$ and $v_2 \rightarrow v_4 / v_2 \dashrightarrow v_4$.

Figure 4, we show a TN (subfigure a) and its trust/distrust transitive closure (subfigure b). By closing trust transitivity we obtain edges $v_1 \rightarrow v_4$, $v_1 \rightarrow v_3$, and $v_2 \rightarrow v_4$. By closing distrust through node v_2, we add $v_1 \dashrightarrow v_4$ with cost $w_1^+ \bullet w_1^-$. As we can see from Figure 4b, transitivity closure leads to two conflicts, i.e., $v_1 \rightarrow v_4 / v_1 \dashrightarrow v_4$ and $v_2 \rightarrow v_4 / v_2 \dashrightarrow v_4$.

The complexity of Algorithm 1 is $O(V^3)$. Our generalization of the Floyd–Warshall algorithm is still $O(V^3)$ as the original one, while the distrust-closure procedure is $O(VE)$: distrust edges can be at most $O(E)$ (line 69), and for each of them there can be at most $O(V)$ antecedent trust-edges, that is one for each of the other nodes in G_M. Since $E = O(V^2)$, this procedure is $O(V^3)$ as well. Finally, Remove-Conflicts is $O(E)$, since it checks at most all the edges for a conflict.

4. Some properties of composition

In this section, we report some formal properties of the algorithm shown in Section 3.1. The trust/distrust transitive-closure procedures on a graph G (shortly $TC(G)$), implemented by Floyd–Warshall$_{\mathcal{B}}$ and Distrust-closure$_{\mathcal{B}}$ functions in Algorithm 4, find a fix point:

Theorem 4.1 (trust/distrust closure as a fix point). Given a bipolar TN, $G = \langle V, E, W, \mathcal{B} \rangle$, then applying Floyd–Warshall$_{\mathcal{B}}$ and *Distrust-closure*$_{\mathcal{B}}$, hence applying a complete transitive closure of trust and distrust over G (i.e., $TC(G)$), computes a fix point of the network: $TC(G) = TC(TC(G))$.

Proof. The proof directly comes from the idempotency of $+_P$ and $+_N$: all the trust paths between each pair of nodes are composed by using $+_P$, i.e., $+_P(W(H_1^+), W(H_2^+), \ldots, W(H_k^+)) = +_P(+_P(W(H_1^+), W(H_2^+), \ldots, W(H_k^+)), W(H_1^+), W(H_2^+), \ldots, W(H_k^+))$. The same holds for distrust paths, composed with $+_N$ instead. □

Note that, using $+$ among all the trust and distrust paths endorses an optimistic view of transitive closure, since this operator always selects the most trustworthy and the least untrustworthy path among all.

Given Algorithm 1 in Section 3.1, the following definition summarizes in a *normal form* the properties expressed by the network obtained through $TC(G)$.

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

Definition 4.1 (normal form). A bipolar TN $G = \langle V, E, W, \mathcal{B} \rangle$ is in normal form if for each trust edge $v_i \rightarrow v_j$ with an associated score w^+, or distrust edge $v_i \dashrightarrow v_j$ with an associated score w^-, all the h trust paths $H^+_{v_i,v_j}$ have a score w^+_i $(i = 1..h)$ such that $w^+_i \leqslant_{+_P} w^+$, and all the k distrust paths $H^-_{v_i,v_j}$ have a score w^-_j $(j = 1..k)$ with $w^-_j \leqslant_{+_N} w^-$.

Therefore, a TN in normal form is 'minimal,' in the sense that the score associated with each edge between any two nodes v_i, v_j condenses all the $H^+_{v_i,v_j}$ and $H^-_{v_i,v_j}$: this score represents how much v_i trusts/distrusts v_j, based on all the paths connecting them. The following property is proved by algorithm construction.

Proposition 4.1. By executing the code at lines 1–5 in Algorithm 1, the obtained TN, $G_M = \langle V_M, E_M, W_M, \mathcal{B} \rangle$ is in normal form.

Note that, if we need to bring a single conflict-free bipolar-TN to normal form, we only need to perform the steps at lines 4 and 5 in Algorithm 1. Therefore, from Algorithm 1 we can directly extract all the necessary code to allow a graph to be in normal form.

The transitive-closure step can of course modify a TN by adding new edges, or changing their labels: in general we have $G \neq TC(G)$. However, the TC procedure does not alter the trust/distrust global evaluation between any two nodes, but it only elicits this score and makes it explicit as the label of the related edge. In words, a TC step elicits the trust scores between any pair of individuals, 'emerging' from 'hidden' paths and becoming visible. It is important to separate between trust in the ability to recommend a good car mechanic, which represents *referral trust* (indirect), and trust in actually being a good car mechanic, which represents *functional trust* (direct) (Jøsang *et al.* 2006b). Therefore, trust eliciting corresponds to a derivation of functional trust through referral trust. Hence, we can define an *explicit form* for graphs:

Definition 4.2 (explicit form). A bipolar TN, $G = \langle V, E, W, \mathcal{B} \rangle$, is in the explicit form if $G = TC(G)$.

In the following theorem we relate the explicit form and the normal form:

Theorem 4.2 (explicit implies normal). If a bipolar TN, $G = \langle V, E, W, \mathcal{B} \rangle$, is in the explicit form, then it is also in the normal formal, i.e., $explicit \Rightarrow normal$.

Proof. The considered G is already a fix point of TC, and from Theorem 4.1 we have that G is also in normal form. □

Note that the property of being in normal form, and consequently being in explicit form, can disappear after the last step of conflict removal (line 6 in Algorithm 1), since it may remove some of the trust and/or distrust paths. In the future, we will investigate alternative composition-procedures that enforce this normal form even after the conflict-removal step (see Section 7).

5. Composing two trust networks: an example

In order to show the behaviour of the proposed algorithm, as a running example we consider the two TNs (a) and (b) in Figure 5. The vertices of such two networks partially

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

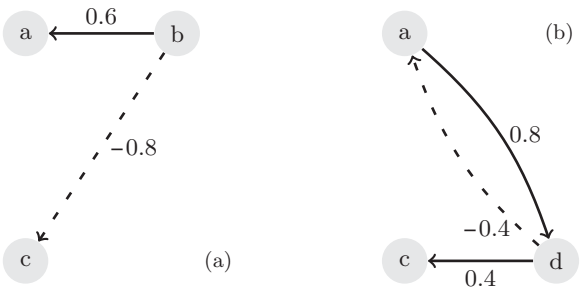


Fig. 5. The two TNs to be composed.

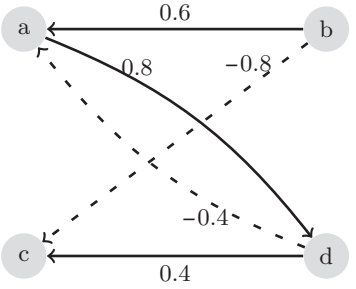


Fig. 6. The preliminary composition of Figure 5.

overlap, since a and c are present in both of them. We can consider, for instance, that b and d are two fresh friends: thus, they want to share their past experience with two friends they have in common (a and c). The scope of these two TNs can be simply 'being a reliable friend.'

To study this case, we consider the 'à-la-bipolar' preference-structure $\langle N, P, \geqslant, sum, -1, 0, 1\rangle$, where $P = [0..1]$ and $N = [-1..0]$, $\times = sum$ is the arithmetic sum, $+ = \geqslant$, -1 is the worst trust level, 1 is the best one, and 0 stands for indifference. We consider the positive preference structure as a semiring $\langle P, +_P, \times_P, 0, 1\rangle$, where $\times_P = min$ and $+_P = max$. As well, in $\langle N, +_N, \times_N, -1, 0\rangle$ we have $\times_N = min$, and $+_N = max$. Moreover, $\bullet = min$.

Figure 6 shows the first network G resulting from merging the two bipolar TNs in Figure 5. Now, by following Algorithm 1, we can apply the transitive closure of trust/distrust edges. Four sets of paths need to be checked during transitivity closure:

i. $a \to d \to c$
ii. $b \to a \to d$
iii. $b \to a \to c$ (with $a \to c$ introduced through step i and $b \to a \to d \to c$
iv. $b \to d \dashrightarrow a$ ($b \to d$ derived from $b \to a \to d$). Here we close distrust.

Considering path i, we need to add a new trust-edge between a and c with a weight of $0.4 \times_P 0.8 = 0.4$ ($\times_P \equiv min$). For path ii, we add a new trust-edge between b and d with a weight of $0.6 \times_P 0.8 = 0.6$. From the two paths reported in iii we obtain a single edge between b and c, whose weight is obtained by composing all the positive preference-scores of the two paths in the following way: $(W(b,a) \times_P W(a,c)) + (W(b,a) \times_P W(a,d) \times_P W(d,c)) = (0.6 \times_P 0.4) + (0.6 \times_P 0.8 \times_P 0.4) = 0.4$. At last, for path iv we add a new

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

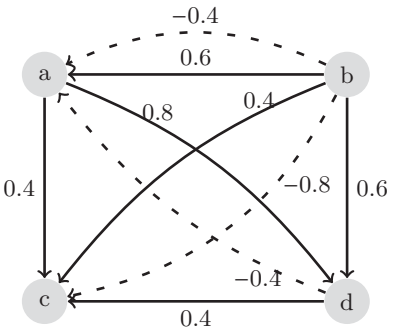


Fig. 7. Figure 6 after trust/distrust transitive closure.

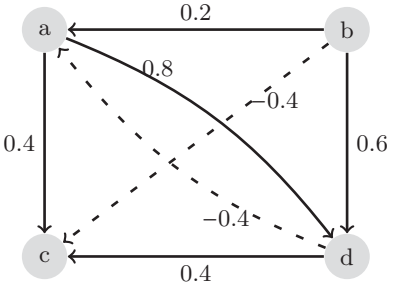


Fig. 8. Our final TN, obtained by composing the two TNs in Figure 5.

distrust edge with a weight of $-0.4 \bullet 0.6 = min(-0.4, 0.6) = -0.4$ between b and a. In total, we obtain four new edges (see Figure 7). Note that we just list the previous four sets of paths without accurately following the behaviour of the algorithm, for the sake of clarity. In practice, their order is not relevant (iii uses an edge found for i), since the *Floyd–Warshall*$_{\mathcal{B}}$ procedure in Algorithm 1 finds all the trust edges at the same time.

As a final step, we need to remove all the possible inconsistencies introduced after the transitive-closure step. From Figure 7 we notice we have two different conflicts, (i) between nodes a and b (since $b \rightarrow a$, and $b \dashrightarrow a$ at the same time), and (ii) between b and c (since $b \rightarrow c$, and $b \dashrightarrow c$). In case of i, we need to use our compensation-operator $\times$ between the two edges in parallel, that is $-0.4 \times 0.6 = 0.2$. For (ii) instead, we obtain $-0.8 \times 0.4 = -0.4$. Our final TN is presented in Figure 8.

5.1. *An application to unweighted trust-networks*

The procedure we have explained in Section 3.1, can be straightforwardly applied to classical unweighted TN, where the relationship between two individuals only consists in a plain trust/distrust feedback (without an associated score).

In this section, we show an example of composition between the two TNs in Figure 9. Both of them are in normal form (see Definition 4.1). To handle this crisp composition, we need to use a different bipolar structure, i.e., $\langle N = \{distrust, \Box\}, P = \{\Box, trust\}, +, \times, \bot = distrust, \Box = indifference, \top = trust\rangle$. The $+$ operator of this bipolar structure is defined

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

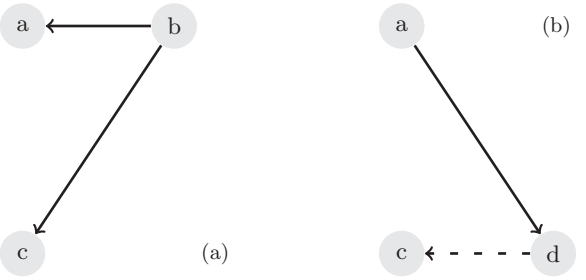


Fig. 9. Two unweighted TNs to be composed.

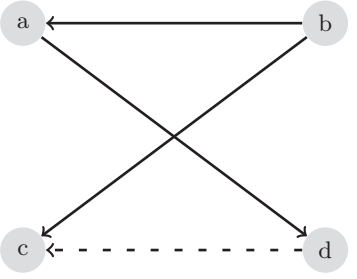


Fig. 10. The composition of a and b in Figure 9.

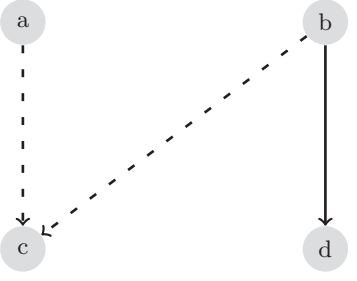


Fig. 11. The edges added during trust/distrust transitive-closure.

by $trust \geqslant indifference \geqslant distrust$, thus $distrust + trust = trust$, while e.g., $distrust \times trust = distrust$, and $trust \times indifference = indifference$. Moreover, we have $\bullet = \times$, i.e., $trust \bullet distrust = distrust$.

After the disjoint union, we obtain the network in Figure 10. Still following the same algorithm in Section 3.1, then we apply a transitive closure step on the following distinct three paths: (i) $a \rightarrow d \dashrightarrow c$ (where (a trusts d) $\bullet$ (d distrusts c) = a distrusts c), (ii) $b \rightarrow a \rightarrow d$ (where (b trusts a) $\times_p$ (a trusts d) = b trusts d), and (iii) $b \rightarrow d \dashrightarrow c$ (where (b trusts d) $\bullet$ (d distrusts c) = b distrusts c). The same distrust edge between b and c can be added also by examining the paths $b \rightarrow a \dashrightarrow c$ and $b \rightarrow a \rightarrow d \dashrightarrow c$. All new edges are represented in Figure 11, while Figure 12 shows the composition of these edges with the TN in Figure 9.

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

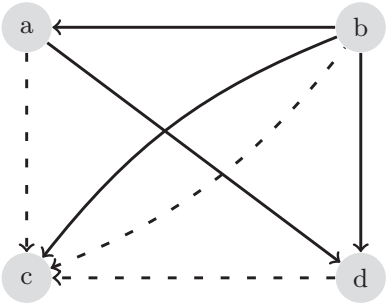


Fig. 12. Our TN after the trust/distrust-closure step.

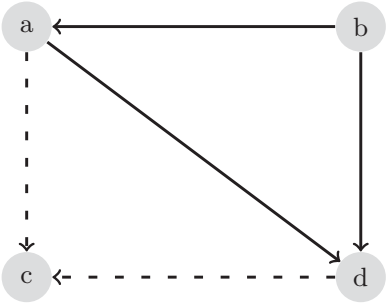


Fig. 13. The final bipolar TN we obtain by composing the two networks in Figure 9.

Finally, we note that the TN in Figure 12 contains one conflict between b and c. Therefore, we resolve it by compensating $trust \times distrust = distrust$, and we remove both edges between b and c. The final result is shown in Figure 13.

6. Related work and discussion

In Thirunarayan and Verma (2008) the authors model trust as a value in the interval $[0 \dots 1]$, while they represent distrust as a score in the interval $[-1 \dots 0]$. Trust and distrust evaluations are represented in two separate matrices of values T and D: $T_{i,j}$ represents how much i trusts j, while $D_{i,j}$ how much i distrusts j. The mathematical operations consist in subtracting one matrix from another, i.e., $R = T - D$; thus, R finally stores the result scores. By using semirings, we can consider different operations instead.

In Guha *et al.* (2004), the authors associate a weight of 1 to all trust edges, and -1 to all distrust edges. In order to compute a final result between i and j, all paths leading from i to j are considered, and their weights are summed together. In the end, a positive result means trust, while a negative means distrust between the two entities.

In Victor *et al.* (2009), the authors make the use of a trust model in which trust scores are (trust/distrust)-couples, drawn from a bilattice that preserves valuable trust information as gradual trust, distrust, ignorance, and inconsistency. The composition is used to refine a recommendation-system in order to improve its precision.

In Verbiest *et al.* (2012) the authors explicitly consider TNs as social networks in which users can assign trust scores to each other. To estimate these scores for agents

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

that are indirectly connected through the network, they introduce and evaluate several path-length incorporating aggregation-strategies. The aim is to strike the right balance between generating more predictions on the one hand, and maintaining a high prediction accuracy on the other hand. Both trust and distrust are propagated through a network by using a *bilattice* structure.

In Prade (2007), the authors propose a qualitative bipolar argumentative-modelling of trust. The approach encompasses only a finite number of levels the trust scale. The proposed methodology is bipolar because both trust and distrust can be independently assessed, and it is argumentative since trust assessment is the result of an argument-based process. The last three works adopt lattices of preferences, similarly to semirings, but their application is not focused on merging TNs.

Some work has been developed using trust as a metric to integrate different information together. For instance, several papers have been proposed to compose (Web) services in a trustworthy way (Singh 2003), or, in general, to provide information fusion (Rogova and Nimier 2004). Trust fusion, intended as combining personal experience with recommendation coming from others, is part of trust propagation and it can be used to derive a more reliable measure of trust in the same entity. For instance, the work in Jøsang *et al.* (2006a) investigates possible formal models that can be implemented using belief reasoning based on *Subjective Logic*.

While providing a computational framework to model propagation of trust in TNs has been widely investigated in the literature (e.g., Jøsang *et al.* (2006b) and Bistarelli and Santini (2008)), few works deal with the propagation of trust *and* distrust at the same time. In addition, as far as we know, our work represents the first attempt to composing two TNs. This is our primary goal, that is to produce a computational and formal framework a user can adopt to merge two different subjective views, with the same scope, of trust/distrust relationships in partially-overlapping communities. Some of the other works in the literature, as, for instance, Jøsang *et al.* (2006b), Bistarelli and Santini (2008), Verbiest *et al.* (2012) and Guha *et al.* (2004) propose a computational framework to propagate trust/distrust on a single TN.

Both positive and negative ratings are quite common in today's trust/reputation systems. Such duality can be found in form of *likes* and *dislikes* in Internet Fora or Social Networks, in the rating of product reviews in e-commerce portals (as *Amazon*), or consumer review-sites, as well as in the positive/negative ratings of sellers (as *eBay*) in e-commerce systems. From this we can deduce that, in modern systems, distrust is widely recognized as a distinct concept from trust; modelling distrust by using a low trust-score is clearly less informative and shows less ties to human reasoning Dimoka (2010). In the presented approach, the ultimate goal is TN composition, not trust propagation or recommendation-systems, differently from all the works presented in this section.

7. Conclusion

In this paper we have presented our ideas on composing two TNs with the help of à-la-bipolar preference-structures (see Section 3). These formal structures are general and parametric, so that they can be replaced with specific metric instances according to

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

the given problem: different semiring-instances can represent different trust and distrust metrics, and they can be replaced in the algorithm as required. Clearly, the semiring operators help to compose trust and distrust along a path, and to remove conflicts when both trust and distrust are experienced between the same two individuals.

Trust transitivity, as trust itself, is a human mental phenomenon, so there is no such thing as objective transitivity, and transitivity therefore lends itself to different interpretations. We are aware that entirely capturing subjective-judgment is not something that can be done in general, but this is also the reason why we propose a parametric preference structure to represent it.

In the future, we plan to continue working on novel features of TNs and further studying their properties, still using an algebraic-based framework. For instance, as explained in Section 3.1, when we elicit all the transitive paths on the network, we discover new trust/distrust direct relationships. Then we use an operator (i.e., $\times$) to compensate between trust and distrust paths. Instead, we can take a different decision by removing one of the two paths from the network, rather than compensating. In this case we obtain a less ambiguous network, not allowing ambivalences in trust and distrust: we name it a 'safe TN.' A network is defined as safe if the transitive closure of trust/distrust does not lead to ambivalences. We can obtain this by removing some edges, or by declaring them as not transitive, and we can perform this step by minimizing the number of removed edges, for example. The goal of this different approach is to keep a TN in normal form: the final conflict-removal step of the algorithm can make such property do not hold anymore (as explained in Section 4).

Moreover, we aim at investigating the use of different operators from Subjective Logic Jøsang *et al.* (2006a), instead of semiring operators.

Acknoledgements

We would like to thank Simon N. Foley (Department of Computer Science, UCC Ireland) for sharing with us some of the ideas on merging trust-network, presented in this paper.

References

Bistarelli, S. (2004). *Semirings for Soft Constraint Solving and Programming*. Lecture Notes in Computer Science, vol. 2962, Springer.

Bistarelli, S., Montanari, U. and Rossi, F. (1997). Semiring-based constraint satisfaction and optimization. *Journal of the ACM* **44** (2) 201–236.

Bistarelli, S., Pini, M. S., Rossi, F. and Venable, K. B. (2011). Uncertainty in bipolar preference problems. *Journal of Experimental and Theoretical Artificial Intelligence* **23** (4) 545–575.

Bistarelli, S. and Santini, F. (2008). Propagating multitrust within trust networks. In: *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC)*, ACM 1990–1994.

Bistarelli, S. and Santini, F. (2014). Two trust networks in one: Using bipolar structures to fuse trust and distrust. In: *12th Annual International Conference on Privacy, Security and Trust*, IEEE 383–390.

Dimoka, A. (2010). What does the brain tell us about trust and distrust? Evidence from a functional neuroimaging study. *Mis Quarterly* **34** (2) 373–396.

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press

Floyd, R. W. (1962). Algorithm 97: Shortest path. *Communications of the ACM* **5** (6) 345.

Golbeck, J. (2008). *Computing with Social Trust*. 1st edn. Springer Publishing Company, Inc.

Guha, R., Kumar, R., Raghavan, P. and Tomkins, A. (2004). Propagation of trust and distrust. In: *Proceedings of the 13th International Conference on World Wide Web*. WWW '04, ACM 403–412.

Heider, F. (2013). *The Psychology of Interpersonal Relations*, Psychology Press.

Jøsang, A., Hayward, R. and Pope, S. (2006b). Trust network analysis with subjective logic. In: *Proceedings of the 29th Australasian Computer Science Conference* – volume 48, ACSC '06. Australian Computer Society, Inc. 85–94.

Jøsang, A., Marsh, S. and Pope, S. (2006a). Exploring different types of trust propagation. In: *Proceedings of the 4th International Conference on Trust Management*, iTrust'06. Springer-Verlag 179–192.

MckNight, D. H. and Chervany, N. L. (1996). The meanings of trust. Technical Report MISRC Working Paper Series 96-04, University of Minnesota, Management Information Systems Research Center.

McKnight, D. H. and Chervany, N. L. (2001). While trust is cool and collected, distrust is fiery and frenzied: A model of distrust concepts. In: *Amcis 2001 Proceedings* 171.

Prade, H. (2007). A qualitative bipolar argumentative view of trust. In: *Proceedings of the 1st International Conference on Scalable Uncertainty Management*, SUM '07, Springer-Verlag 268–276.

Rogova, G. L. and Nimier, V. (2004). Reliability in information fusion: Literature survey. In: *Proceedings of the 7th International Conference on Information Fusion*, vol. 2, 1158–1165.

Singh, M. P. (2003). Trustworthy service composition: Challenges and research questions. In: *Proceedings of the 2002 International Conference on Trust, Reputation, and Security: Theories and Practice*, AAMAS'02, Springer-Verlag 39–52.

Theodorakopoulos, G. and Baras, J. S. (2004). Trust evaluation in ad-hoc networks. In: *Proceedings of the 3rd ACM Workshop on Wireless Security*, WiSe '04, ACM 1–10.

Thirunarayan, K. and Verma, R. (2008). A framework for trust and distrust networks. In: *Proceedings of Web*, vol. 2.

Verbiest, N., Cornelis, C., Victor, P. and Herrera-Viedma, E. (2012). Trust and distrust aggregation enhanced with path length incorporation. *Fuzzy Sets and Systems* **202** (Sept.) 61–74.

Victor, P., Cornelis, C., De Cock, M. and Pinheiro da Silva, P. (2009). Gradual trust and distrust in recommender systems. *Fuzzy Sets and Systems* **160** (10) 1367–1382.

https://doi.org/10.1017/S0960129515000092 Published online by Cambridge University Press