Individualised Trust in Social Networks

Personalisiertes Vertrauen in sozialen Netzwerken
Bachelor-Thesis von Oliver Richters aus Wiesbaden
März 2010

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Physik
Institut für Festkörperphysik
Theorie komplexer Systeme

Abstract

Non-centralised recommendation based decision making is a central feature of several social and technological processes, such as market dynamics, peer-to-peer file-sharing and webs of trust of digital certification. We propose a metric for calculating transitive trust in social networks, based on the direct trust among agents. Our metric fully captures the individualised nature of trust and does not rely on any specific topological characteristic of the network, contrary to similar methods proposed in the literature. Further, we investigate the general properties of trust in random networks according to different strategies of choice of direct trust between agents.

Contents

1 Introduction

Systems in the form of networks are universally present. Common examples include power grids [1], the world wide web [2] or relationships between people [3]. Other important examples are food webs [4] and neural networks [5] in biology or the famous Seven Bridges of Königsberg [6], known as the foundation of mathematical analysis of networks. The nodes of the network, called vertices, may thus be computers, human beings, places or species depending on the specific application. The connections or edges can represent friendship, geographical proximity or predator-prey relationship. This linkage can be reciprocal like connected routers [2] or directed as signals in gene regulatory networks [7].

In particular cases, the nodes act independently but share or exchange information, resources or other goods. If this interaction is voluntary, the concept of trust is fundamental, since it measures how much a given agent can confide in the quality, reliability or authenticity of another agent.

In cryptography, the authenticity of the binding between a public key and a user is an important condition for the security of the encryption. “Pretty Good Privacy” (PGP) [8][9] is an encryption software using certificates, thus users can sign keys to affirm this binding, expressing a form of trust. That leads to a network of trust connections, the “web of trust”.

In everyday life, human beings have established strategies for allocating and dealing with trust. However, this notion of trust is extremely localised, since it is mostly formed by experiences with close acquaintances. In a globally connected system, such as some of those mentioned previously, agents have the possibility to interact with a number of other agents which far surpasses their local neighbourhood of close acquaintances. Trust propagation in such large systems far exceeds the limits of traditional trust comprehension. One straightforward concept which can be used to generalise trust in this situation is transitivity, where non-local trust relationships are inferred from direct relationships: Assume that Alice trusts Bob and Bob trusts Carl. Applying transitivity, Alice should trust Carl to some extent. Progressing this way, Alice may have an opinion which reaches far outside her sphere of direct interactions. However, what if Alice’s friend Eve does not trust Carl and therefore the transitivity-induced trust is ambiguous?

In reality, trust may be composed of countless characteristics involving complicated rules and may be heavily dependent on the context. However, in its simplest and most tractable manifestation, trust can be reduced to a single, non-negative real number which can be interpreted as a probability. With this simplification, transitivity can be achieved by the multiplication of those values and a trust metric may be defined in a straightforward manner. Without a doubt, the concept of transitivity does not necessarily lead to unique results as it is already obvious in Alice’s environment. Hence a trust metric has to mediate between different opinions.

In former work, this has been done in various ways [17] [20] [21], trying to establish mathematical concepts of trust propagation on networks. We will describe them precisely and show why these known approaches are inappropriate to describe trust in combination with transitivity.

To ease the understanding of the measures and methods used in our work, a short introduction to graph-theoretical basics is given in the next section. Section 3 treats previous metrics, highlighting limitations and conceptual problems. Subsequently, a stepwise development starting with simple transitivity approaches leads to a sophisticated metric that combines various alternatives to a single trust value. We illustrate usability and behaviour of this metric with the help of simple examples. In the last section, we will demonstrate its application on random

networks and examine an emerging non-linear behaviour with probability calculus. We resume our results and demonstrate possibilities for future research in the conclusion.

2 Mathematics

Our approach to deal with networks is to treat them as graphs. The following concepts are thus mainly taken from graph-theory and are fundamental for the analysis and derivations to come.

2.1 Basic definitions

A graph or network is an abstract representation of a set of vertices V where some pairs of them are connected by a set of edges E. On a directed graph, every edge $\vec{e} \in E$ points from the source $s(\vec{e}) \in V$ to the target $t(\vec{e}) \in V$.

In the example graph in figure 1, $s(\vec{e}) = c$ and $t(\vec{e}) = d$. On random graphs, the topological structure is mainly described by the size, thus the number of vertices $N = |V|$, and the probability distribution of the number of neighbours.

2.2 Neighbours of a vertex and degree

For each vertex v, the neighbours $N(v) \subset V$ is the set of vertices with a connecting edge to v. On a directed graph, we must differentiate between in-neighbours $I(v)$ and out-neighbours $O(v)$. Thus for every vertex $j \in I(v)$, there exists an edge $\vec{e}_{jv}$ from j to v and therefore $v \in O(j)$.

The degree $d(v)$ is the cardinality of the set of neighbours $N(v)$. On a directed graph, we must distinguish between in-degree $d_i(v) \equiv |I(v)|$ and out-degree $d_o(v) \equiv |O(v)|$. The notion of mean degree

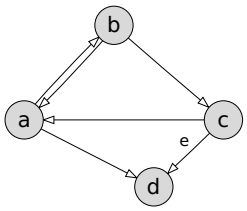

Figure 1: A directed graph

$$\overline{d} = \frac{\sum_{v \in V} d(v)}{|V|} \tag{1}$$

can also be trivially expanded to mean in-degree $\overline{d_i}$ and mean out-degree $\overline{d_o}$. In the example on the side, $I(a) = \{b, c\}$, $O(a) = \{b, d\}$ and $d_i(a) = d_o(a) = 2$.

As mentioned above, the degree distribution of a random graph is a measure of the connection pattern in the underlying network [10]. What is mainly used in this work is the Poisson distribution, thus the fraction of vertices of degree d is expected to be

$$p(d) = \frac{\lambda^d}{d!} e^{-\lambda}, \tag{2}$$

with a mean degree $\overline{d}$ of λ. Such graphs are known as Erdős - Rényi graphs [11].

To describe connections of non-adjacent vertices in a network, we have to introduce the formalism of a path.

2.3 Path

A path P is a list of non-repeating edges $[\vec{e}_1, ..., \vec{e}_n]$ in which $t(\vec{e}_k) \equiv s(\vec{e}_{k+1}) \ \forall \ k$. All traversed vertices have to be distinct. A useful notation for source and target of a path is $s(P) \equiv s(\vec{e}_1)$ and $t(P) \equiv t(\vec{e}_n)$. The length of the path $|P|$ is n.

2.4 Geodesic distance g

The length of the shortest path from s to t is called shortest or geodesic distance $g_s(t) \in \mathbf{N_0}$. If no path exists between the two vertices, the distance is defined as infinity. The shortest path and in this manner the geodesic distance can be figured out by breadth-first search BFS [15]. In the graph in figure 1, $g_b(d) = 2$ while $g_d(b) = \infty$.

The mean geodesic distance[1] $\overline{g}$ is defined as

$$\overline{g} = \frac{\sum_{i,j \in V} g_i(j)}{|V|^2}. \tag{3}$$

In a very basic approach, we can assume that the number of vertices n reachable within x steps from a given vertex is $n = \left(\overline{d_o}\right)^x$. For $n = N$, we get an approximation for the mean geodesic distance as

$$\overline{g} = \frac{\log(N)}{\log\left(\overline{d_o}\right)}. \tag{4}$$

3 Previous trust metrics

In this section, we look at former approaches to describe trust or related concepts on networks and highlight their applications and limitations. The treated metrics are the link analysis algorithm PageRank [17], the peer-to-peer reputation management EigenTrust [20] and the personalised metric TrustWebRank [21].

3.1 PageRank

PageRank [17] is a link analysis algorithm calculating the relative importance c_i of each node i. This process, invented by Larry Page et al., is used by the popular search engine "Google" to rank websites [18]. If the existence of a link can be interpreted as providing confidence from the source to the target, this relative importance can be used as a measure of trust. The basic principle is that a node should have a higher value the more important its neighbours are:

$$c_i = \beta \sum_{j \in I(i)} \frac{c_j}{d_o(j)} + (1 - \beta), \tag{5}$$

where $\beta \in [0, 1)$ is a damping factor. Defining a stochastic transition matrix P with

$$P_{ij} = \begin{cases} d_o^{-1}(j), & \text{if there exists a link between } j \text{ and } i, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

we can write equation (5) in matrix notation, calling I the unity matrix and $\vec{I}'$ a vector with 1 in every component,

$$\vec{c} = \beta P \vec{c} + (1 - \beta)\vec{I}'. \tag{7}$$

[1] As firstly discovered by Milgram in his famous "small-world" experiments [12] [13], the mean geodesic distance in most real networks is rather small. He realised that the mean geodesic distance in the worldwide network of human acquaintances is close to 6.

This equation can be solved using elementary algebra leading to a vector

$$\vec{c} = (I - \beta P)^{-1}(1 - \beta)\vec{I}' \tag{8}$$

that contains the values c_i of the vertex importance.

Note the numerous limitations of this approach: The direct trust is a stochastic matrix, meaning that the total out-going trust of an agent is normalised to 1. That signifies that the measure of trust is only relative. As this model does not allow the agent to define different trust values for its neighbours, diversity in the amount of trust can not be expressed. The damping factor β introduces a characteristic path length because the influence of long paths decays exponentially with its length. There is no justification why long paths should *a priori* be less important. In the original application, they assumed that an Internet user may randomly restart from a new website with a probability given by $1-\beta$ instead of further following the links. If the network of links contains cycles, connections will be taken into account repeatedly, thus the same opinion can be used more than once. Finally, another strong point of criticism is that PageRank is a global metric, thus the trust value is independent of the asker and therefore not personalised.

We must conclude that the assumptions implied by PageRank are not sufficient for trust on social networks even though they are suitable for citations or the links of the world wide web.

In the next section, we examine the central role of trust in other domains of computer technologies. In particular, resource sharing on a network is a paradigm for the necessity of trust between peers. In this context, a metric called Eigentrust [20] has been proposed.

3.2 EigenTrust

A peer-to-peer network is a distributed network architecture composed of participants that make a portion of their resources such as processing power, disk storage or network bandwidth directly available to other network participants, without the need for central coordination [19]. EigenTrust [20] by Kamvar et al. introduces reputation in peer-to-peer networks in order to increase reliability on the network. It allocates to each peer a global trust value based on its traffic history. For each pair of peers $i \neq j$, the direct trust value s between them is calculated by subtracting the number of unsatisfactory transfers from the number of satisfactory transfers

$$s_{ij} = \text{sat}_{ij} - \text{unsat}_{ij}. \tag{9}$$

The following normalisation, which assigns the same total weight to the opinion of each peer, limits consequences of malicious peers misleading other agents by assigning wrongful direct trust values to other peers:

$$c_{ij} = \frac{\max[0, s_{ij}]}{\sum_k \max[0, s_{ik}]}. \tag{10}$$

Representing the out-going values for every node as a vector, we get the notation

$$\vec{c}_i = \sum_j c_{ij}\vec{e}_j \text{ where } \|c_i\|_1 = 1. \tag{11}$$

If the matrix $C = [c_{ij}]$ is aperiodic and irreducible[2], $\vec{t}(i) = (C^T)^n \cdot \vec{c}_i$ converges for $n \to \infty$ to the left eigenvector $\vec{t}$ of matrix C, which is a global value independent of i. Each component t_j

[2] For other cases, Kamvar et al. propose different strategies. [20]

of this global trust vector $\vec{t}$ quantifies how much the system as a whole trusts peer j. In a probabilistic interpretation, this vector $\vec{t}$ is the stationary distribution of the Markov chain induced by the transition matrix C.

In contrast to PageRank, the direct trust s_{ij} contains more information than a simple link, but instead a satisfaction value. The rather arbitrary introduction of the damping factor β is avoided here, but the problem with cycles remains. However, two important critiques persist: the lack of personalisation and the fact that $\vec{t}$ contains only relative values.

In the next section, we describe how TrustWebRank [21] implements a personalisation of the concepts above.

3.3 TrustWebRank

Introducing the metric TrustWebRank [21], Walter et al. expand the concept of PageRank by personalising it. The direct trust matrix D, where D_{ij} reflects the direct trust from agent i to j, will be normalised to get a stochastic matrix $\tilde{D}$, hence $\sum_j \tilde{D}_{ij} = 1 \ \forall \ i$. They define T_{ij} to be the indirect trustworthiness score from i to j, satisfying the matrix equation

$$T = \tilde{D} + \beta \tilde{D} T, \tag{12}$$

where $\beta \in [0, 1)$ has a similar role as the damping factor in PageRank, discounting the impact of agents far away in the network. Thus the indirect trust from agent i to j is computed as a combination of the direct trust and the trust that the neighbours of agent i have in j. Equation (12) can be derived using elementary algebra:

$$T = (I - \beta \tilde{D})^{-1} \cdot \tilde{D}. \tag{13}$$

Note that when dealing with huge graphs, instead of directly inverting a matrix (complexity $O(N^2)$) as required by equation (13) and equation (8) in PageRank, an iterative method can be used.

To avoid T_{ij} being out of the range $[0, 1]$, a normalisation is applied a posteriori

$$\tilde{T}_{ij} = \frac{T_{ij}}{\sum_{k \in O(i)} T_{ik}}. \tag{14}$$

In contrast to PageRank and EigenTrust, this metric is personalised because it leads to a matrix instead of a vector of trustworthiness. However, several limitations remain: Firstly, TrustWebRank does not differentiate between bad opinion (low trust) and no opinion. Secondly, because of the first normalisation $D \to \tilde{D}$, there is no difference if an agent trusts in all his neighbours equally with a high or a low trust. Analog to PageRank, the introduction of a characteristic path length by the factor β is not justified. Finally, the resulting trust value is not inherently in the range $[0, 1]$. The given normalisation simply reduces the sum of outgoing trust of each agent to 1, providing comparability between normalised direct trust $\tilde{D}_{ij}$ and indirect trust $\tilde{T}_{ij}$. A comparison with the initial direct trust D is impossible. Therefore TrustWebRank only provides a relative trust.

4 A new approach to trust

Faced with the limitations of the known trust metrics, our aim is to establish a metric describing trust on an arbitrary network. The constraint for the trust values has to be as small as possible. To achieve this goal, the metric should work independently of topology on every directed graph[3]. The resulting trust from node s to t ought to be personalised, hence be specific for s. In addition to that, easy manipulations through malicious agents are to be avoided and numerical stability is desired.

After presenting some general considerations, we will introduce simple attempts, which are themselves not sufficient, but if combined, result in our final proposed trust metric.

As a pre-condition for calculating trust in a network, we allocate a direct trust value $D_{\vec{e}}$ to each edge $\vec{e} \in E$, as explained in the next section.

4.1 Direct trust

The direct trust D is by definition a real-valued edge property in the range of $[0,1]$. The relevance of this constraint is that it guarantees boundedness and convergence on graphs. D can be seen as a probability, thus $D_{\vec{e}} = 1$ signifies absolute trust, whereas $D_{\vec{e}} = 0$ is total absence of trust. If i is the source-vertex $s(\vec{e})$ and j the target-vertex $t(\vec{e})$, the direct trust $D_{ij} \equiv D_{\vec{e}}$ is in this way well-defined. For every vertex v, we can define $\overline{D_i}(v)$ as the mean in-trust and $\overline{D_o}(v)$ as the mean out-trust,

$$\overline{D_i}(v) = \frac{\sum_{k \in I(v)} D_{kv}}{d_i(v)}, \tag{15}$$

$$\overline{D_o}(v) = \frac{\sum_{k \in O(v)} D_{vk}}{d_o(v)}. \tag{16}$$

The first step in generalising trust from direct connections is to consider the transitivity of consecutive edges.

4.2 Trust along a path (transitivity)

For every path P, we define the trust D_P along P as the product of the direct trust of the containing edges:

$$D_P = \prod_{\vec{e}_i \in P} D_{\vec{e}_i}. \tag{17}$$

Given two vertices s and t, we can easily find the maximum trust value H_{st} between s and t with Dijkstra's algorithm [22],

$$H_{st} = D_{P^*_{st}} = \max \left[D_P \text{ with } s(P) = s \text{ and } t(P) = t \right], \tag{18}$$

where P^*_{st} is called the path of highest trust. If no path exists[4], we set H_{st} to 0. The self-trust H_{ss} is set to 1. It can easily be seen that all the introduced measures are in the same range $[0,1]$ as the direct trust D. The seemingly arbitrary choice of this interval becomes inevitable, since it is the only non-trivial real interval that is mapped bijectively on itself by any function $f(x) = x^c$ for $c > 0$.

[3] We consider graphs without self-loops and parallel edges, which are meaningless on trust networks.

[4] That may be the case if the network is not fully connected.

As an example, let us calculate the trust in the graph in figure 2 for the solid black path $P_{b-} = [\vec{a}, \vec{b}, \vec{c}]$ and the solid grey path $P_{g-} = [\vec{d}, \vec{e}]$:

$$D_{P_{b-}} = a \cdot b \cdot c = 0.36,$$

$$D_{P_{g-}} = d \cdot e = 0.35.$$

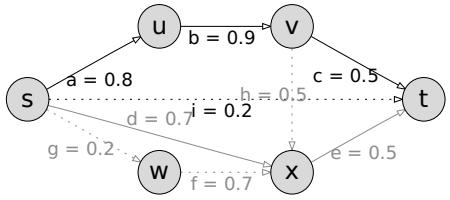

Figure 2: Trust along a path with $x = D_{\vec{x}}$

It can easily be seen that P_{b-} is the path of highest trust between s and t and therefore:

$$H_{st} = D_{P_{b-}} = 0.36.$$

With this concept in mind, we can attempt the definition of a first trust metric.

4.3 The Lotta metric

The Lotta metric[5] is based on the "trust along a path" (4.2) and defines the indirect trust L_{st} between nodes s and t as the largest possible trust from s to t.

$$L_{st} = H_{st} \tag{19}$$

The Lotta metric has the advantage of being defined and meaningful on every possible graph. Furthermore, the indirect trust value is automatically normalised to $[0, 1]$ and therefore comparable to the direct trust. However, taking simply a single path does not exploit the advantage of the network structure. Besides, connections with low trust to t are completely neglected resulting in a very optimistic view. This gives huge opportunities to mislead the source node s by adding high trusted links between non-trustworthy nodes, leading to a metric which is not robust against manipulation. Even a low direct trust D_{st} will not be taken into account if a path of larger trust exists. To provide a realistic description of trust on a network, this metric is therefore not appropriate.

We attempt now to compensate this by considering the alternative paths leading to the target.

4.4 The Rome metric

The Rome metric[6] involves all possible paths P_i from s to t to determine the trust R_{st}. We denote P_i' as the path P_i without the last edge $\vec{e}_i$, thus P_i' ends at an in-neighbour of t. We now define the Rome metric using the trust $D_{P'}$ of the in-neighbours of t as a weight on the final trust D_P:

$$R_{st} = \frac{\sum_i D_{P_i'} \cdot D_{P_i}}{\sum_i D_{P_i'}} = \frac{\sum_i (D_{P_i'})^2 D_{\vec{e}_i}}{\sum_i D_{P_i'}}. \tag{20}$$

Hence the Rome metric, by taking into account every possible path to t, uses all the information available in the network. However, this leads to some problems. For instance, considering a fully connected graph with N vertices and constant trust c on all edges. In total, $\sim N!$ paths

[5] Established in a personal conversation with Lotta Heckmann in December 2009.

[6] Freely adapted from Alanus ab Insulis: "All paths lead to Rome".

are involved in the calculation. Apart from the numerical complexity of considering all those paths, it can be shown that the trust decays with c^N, thus tending to zero in the limit of infinite vertices if $c < 1$, as follows: There are $n! \cdot \binom{N-2}{n}$ different paths P of length $n+1$, with $D_{P'} = c^n$ and $D_P = c^{n+1}$. According to equation (20), the trust can be calculated

$$R_{st} = \frac{\sum_{n=0}^{N-2} c^{1+2n} \cdot n! \cdot \binom{N-2}{n}}{\sum_{n=0}^{N-2} c^n \cdot n! \cdot \binom{N-2}{n}} \tag{21}$$

$$= c^{N-1} \frac{\exp\left(\frac{1}{c^2}\right) \cdot \Gamma\left(N-1, \frac{1}{c^2}\right)}{\exp\left(\frac{1}{c}\right) \cdot \Gamma\left(N-1, \frac{1}{c}\right)} \tag{22}$$

$$\leq c^{N-1} \cdot \exp\left(\frac{1}{c^2} - \frac{1}{c}\right), \tag{23}$$

where Γ is the lower incomplete gamma function. A visualisation of this behaviour can be seen in figure 5.

Even on sparse networks, there is a significantly larger number of long paths than of short ones and thus they dominate, leading to small trust values. The amount of information required to calculate the trust is far from being optimised and many nodes may appear in several paths, thus their opinion is involved repeatedly. Therefore the Rome metric is inappropriate for most networks, especially for huge or highly connected ones. The field of application is narrow and it misses the target of generality.

Since equation (20) satisfies the preliminary consideration of comparability of directed and indirect trust, it seems to be an appropriate strategy to combine varying alternatives to a single trust value. The limitations mentioned above were mainly caused by the tremendous number of paths we considered. In order to reduce that number, we combine the advantages of the Rome metric and the Lotta metric and introduce the Pervasive Trust Transitivity.

4.5 Pervasive Trust Transitivity (PTT)

This new trust metric allows calculating an indirect trust value T_{st} for given vertices $s \neq t$ on a network. In order to use every available opinion concerning the target t without counting it repeatedly, we take into account only the best path from s to each in-neighbour $k \in I(t)$. That limits the number of considered paths to the number of in-neighbours $d_i(t) < N$.

To avoid calculating trust to node t based on his own opinion, the following search is done on a subgraph with $V' = V \backslash \{t\}$: For every node $k \in I(t)$, the path of highest trust P'_k is determined such that the product of the weights of its constituent edges is maximised, as explained in 4.2, leading to the values H_{sk}. This way for each incoming edge $\vec{e}_{kt}$, the path of highest trust $P_k = P'_k \cup \vec{e}_{kt}$ passing through $\vec{e}_{kt}$ is found. The indirect trust is defined as the mean over all trust values D_{P_k}, weighted with $D_{P'_k}$. This is similar to the Rome metric, but the average is taken only over the best path to each neighbour, thus also incorporating an aspect of the Lotta metric.

We define the Pervasive Trust Transitivity T of agent s to t as

$$T_{st} = \frac{\sum_{k \in I(t)} D_{P'_k} \cdot D_{P_k}}{\sum_{k \in I(t)} D_{P'_k}} = \frac{\sum_{k \in I(t)} (H_{sk})^2 \cdot D_{kt}}{\sum_{k \in I(t)} H_{sk}}. \tag{24}$$

For the familiar example on the side, we calculate the Pervasive Transitive Trust from s to t: As the in-degree $d_i(t)$ is 3, we have to find the largest possible trust from s to all the three in-neighbours of t: s, v and x. It can easily be seen that we have to involve the black solid path (P_{b-}), the grey solid path (P_{g-}) and the black dotted path ($P_{b:}$). The grey dotted edges are not used in the calculation. The resulting trust T_{st} can be calculated as follows

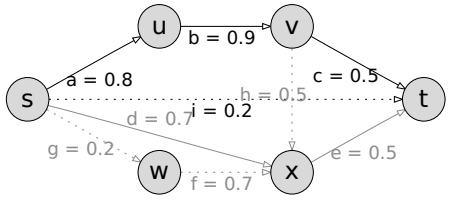

Figure 3: Trust along a path with $x = D_{\vec{x}}$

$$T_{st} = \frac{ab \cdot abc + d \cdot de + 1 \cdot i}{ab + d + 1} \approx 0.29. \tag{25}$$

The bounded values of D and therefore H_{sk_i} guarantee the perfect comparability between direct trust D and indirect trust T without any initial normalisation necessity. The metric is topology-independent and cycles are never taken into account. Disconnected graphs can be treated separately or not, without changing the results. Actually even parallel edges do not cause algorithmic problems[7], only the edge with the highest trust from a set of parallel edges is used.

Furthermore the metric is personalised, since the used paths will always depend on the source vertex. Since there is nothing similar to the damping factors in PageRank or TrustWebRank, no characteristic path length is introduced: If the direct trust $D_{\vec{e}_i}$ of the edges $\vec{e}_i$ composing a path P is equal to 1, the trust along P is 1 as well, independent of its length.

The PTT solution is exact and can be derived directly without iterative methods. The impact of small perturbations ϵ in one of the values of D is not bigger than ϵ, so the calculation is very robust against numerical imprecision. The complexity of the calculation of one single value T_{ij} scales with $O(V \log V)$ [22]. Surely one disadvantage is the necessity to calculate T_{ij} separately for every tuple of vertices. The fastest way to perform that is to determine the trust from all the vertices to one single target t simultaneously. For every in-neighbour $k \in I(t)$, the determination or the paths of highest trust from all $v \in V$ to k can be done using Dijkstra's algorithm [22] on the reversed graph with complexity $O(V \log V)$. Since this has to be done for every edge, the complexity of calculating the indirect trust on the whole network is $O(VE \log V)$.

In the following, we describe some simple examples which will illustrate the behaviour of the metric on some topologies.

4.6 Simple examples with PTT

In this section, we examine three different types of networks: On a directed tree and a fully connected "complete" graph with homogeneous direct trust, T will be determined exactly, whereas in the case of an arbitrary network with homogeneous direct trust, only a general limit can be specified.

[7] Only some of the notations used to describe the metric are not valid or unique anymore.

4.6.1 Directed tree

In the case of a directed tree as in the illustration on the side, it is easy to verify the implementation of transitivity. As there is only one possible path P from a to d, the three metrics we introduced return equivalent results, namely the product of the weights of its constituting edges $\vec{a}, \vec{b}, \vec{c}$, furnishing proof that transitivity in its simplest form is integrated in all of them. Note that EigenTrust and TrustWebRank are not applicable on this nearly trivial example hence node d has no out-going edge.

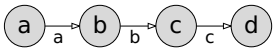

Figure 4: Directed tree

Surely this example does not show many of the qualities of the Pervasive Trust Transitivity. Hence we consider the upper extreme of degree distributions, a network in which all vertices are pairwise connected.

4.6.2 Fully connected graph with homogeneous trust

On a fully connected graph with N vertices and $D_{ij} = c \ \forall \ i \neq j$, there is one direct path from s to t and $N-2$ paths of length 2, one for each possible intermediate node. The resulting trust T_{ij} according to our metric is independent of i and j because of symmetry. It can be calculated for $i \neq j$ as

$$T_{ij} = \frac{(N-2) \cdot c^3 + c}{(N-2) \cdot c + 1}, \tag{26}$$

$$\lim_{N \to \infty} T_{ij} = c^2. \tag{27}$$

Figure 5 shows this dependence on N. While N increases, the importance of the direct connection decays and the paths of length 2 become dominant. The dashed line corresponds to the trust value R_{ij} according to the Rome metric as calculated in equation (21). It can be seen that the PTT metric does not show the unbounded decay but a meaningful behaviour in the limit of infinite graph size.

Even if no specific topology is given, we can nevertheless indicate limits for the PTT trust as it can be seen in the next section.

4.6.3 Arbitrary graph with homogeneous trust

On an arbitrary network with N vertices and $D_{\vec{e}} = c \ \forall \vec{e} \in E$, we can give a bound for T_{st} based on the idea that the length of P_k can not be smaller than the shortest distance $g_s(t)$ and is always smaller then the total number of vertices. Therefore, if $g_s(t) < \infty$, [8] the trust along the path P_k is bounded $D_{P_k} \leq c^{g_s(t)} \ \forall \ k$ and therefore

$$c^{N-1} \leq T_{st} \leq c^{g_s(t)}. \tag{28}$$

5 Pervasive Trust Transitivity on random networks

To understand the characteristics of the PTT metric, we study its behaviour on random networks with different sizes, whereas the degree distribution is a Poissonian defined by the parameter λ.

[8] If no connection exists between s and t, $T_{st} = 0$ as seen before.

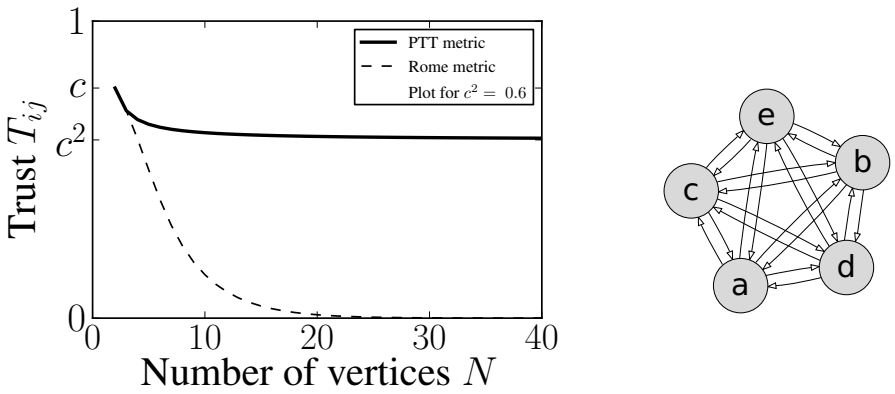

Figure 5: Trust on a fully connected network dependent on its size

The constraint for the distribution of $D_{\vec{e}}$ is a constant mean in-trust to avoid preferential treatment for single vertices:

$$\overline{D_i}(v) = c \in [0,1] \ \forall \ v \in V. \tag{29}$$

For the calculation of $D_{\vec{e}}$, different models will be described in the following subsections.

Once the graph and the direct trust values $D_{\vec{e}}$ are given, we can use our metric to calculate the indirect trust T_{st} for randomly chosen nodes s and t. In order to get a good statistic, we repeat this last step for numerous pairs of nodes and rerun the algorithm until we examined lots of graphs, always with different mean in-trusts c. [9]

In order to compare between different network sizes and degree distributions, we define a statistical measure, the trustworthiness W.

5.1 Trustworthiness W

Whereas one of the most important tasks in establishing a metric was the amount of personalisation, the incoming trust T_{ik} averaged over all vertices i is nevertheless an interesting specific quantity for each node k. We define the trustworthiness W_k of node k as

$$W_k = \frac{1}{N-1} \sum_{i \in V, i \neq k} T_{ik}. \tag{30}$$

As a global network property, the mean trustworthiness is defined as

$$W = \frac{1}{N} \sum_{k \in V} W_k = \frac{1}{N(N-1)} \sum_{i,k \in V, i \neq k} T_{ik}. \tag{31}$$

5.2 Homogeneous direct trust

The easiest way of distributing $D_{\vec{e}}$ is to assign a constant direct trust $D_{\vec{e}} = c$ to all edges. We can then determine the mean trustworthiness W for different degree distributions and network sizes. The dependence on the mean in-trust c can be seen in figure 6.

[9] All numerical calculations were performed using python with the scientific libraries SciPy, Numpy [23] and Matplotlib [24]. For the graph-specific tasks, the free python library graph-tool [25] was used.

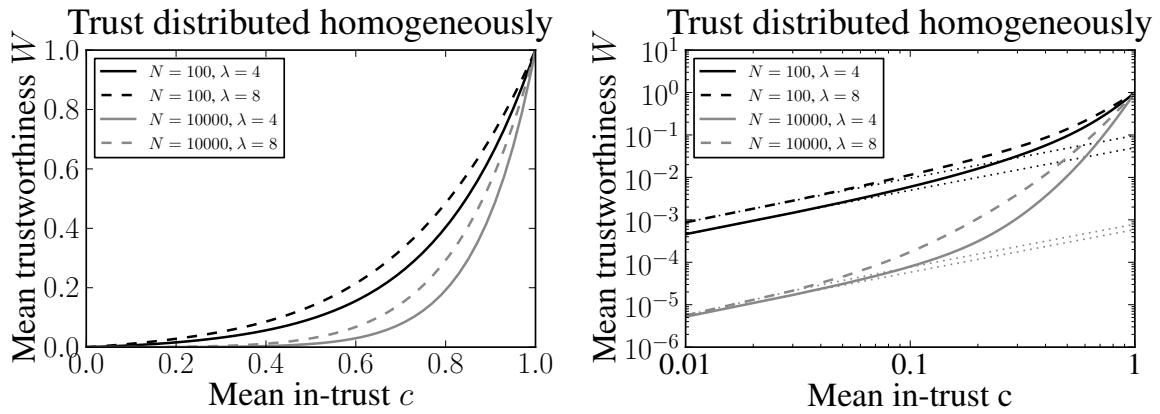

Figure 6: Mean trustworthiness W on the network dependent on the mean in-trust c. In the right plot, linear approximations for $c \ll 1$ are added using dotted line style.

Comparing the different sizes and degree distributions induced by the parameter λ, it is obvious that bigger network size leads to lower mean trustworthiness whereas a higher mean degree causes higher mean trust. This can be explained keeping in mind that the trust T is mainly dependent on the geodesic distance between the two nodes. This value increases with the network size and decreases with higher mean degree.

As we know from 4.2, the trust along each path P is given by $D_P = c^{|P|}$. The trust calculated with our metric is therefore

$$T_{st} = \frac{\sum_{k \in I(t)} c^{2g_s(k)} \cdot c}{\sum_{k \in I(t)} c^{g_s(k)}}. \tag{32}$$

For c small, the following approximation is valid:

$$\lim_{c \to 0} T_{st} = c^{\min\left[g_s(k)\right]}. \tag{33}$$

In order to calculate the mean trustworthiness W, we have to average over T_{st}. If we keep only linear terms and call $p = \frac{\overline{d_o}}{N}$ the probability that nodes are direct neighbours, we get:

$$W \approx p \cdot c \tag{34}$$

This rough approximation can be validated fitting a function $f(c) = a \cdot c$ to the plots. For $N = 100$ and $\lambda = 4$, we get $a = 0.050$, while $\overline{d_o} = 0.04 \cdot N$. For a network with $\lambda = 8$ and same size, $a = 0.974$ while $\overline{d_o} = 0.08 \cdot N$, confirming our approach. A plot of the linear approximation can be seen in the log-log plot in figure 6. For $c \to 1$, the direct trust on all edges and thus W converges to 1 as visible in the graph.

In the next section, we will abandon the homogeneous direct trust. Even in the situation where the average in-trust $\overline{D_i}$ is the same for each node, namely the intrinsic value c, the individual direct trust values can be distributed in a variety of ways. We are interested in testing the robustness of the proposed trust metric in a situation where all agents try to selfishly increase their own trustworthiness by favouring more centrally connected vertices. The betweenness is a measure of this centrality.

5.3 Betweenness centrality c_B

The betweenness centrality [16] is an often used value of relative importance of a vertex $v \in V$ within a graph. We define $\sigma_{st}(v)$ as the number of shortest paths passing through v and σ_{st} the total number of shortest paths, whereas all the path lead from s to t. The betweenness centrality is calculated as follows

$$c_B(v) = \sum_{\substack{(s,t)\in V^2 \\ s,t\neq v}} \frac{\sigma_{st}(v)}{\sigma_{st}} \in [0,1]. \tag{35}$$

Hence vertices that occur on many shortest paths between other vertices have a high betweenness. We now incorporate this notion of centrality in the direct trust distribution.

5.4 Centrality based trust repartition

We allow each node to distribute the incoming trust according to his preferences, but keeping the mean in-trust $\overline{D}_i$ fixed. A particular node k may be interested in increasing his trustworthiness W_k. Therefore, it is reasonable to get higher trust from central nodes, i.e. with high betweenness centrality c_B. Hence distributing the incoming trust D_{ik} proportional to the centrality $c_B(i)$ seems appropriate. Additionally, this approach is robust against small modifications of the centrality values. Since D_{ik} has to be bounded to $[0,1]$, we use

$$D_{ik} = \min\left[\kappa_k \cdot c_B(i), 1\right]. \tag{36}$$

Here $\kappa_k \geq 0$ is chosen for each node k to guarantee the constraint of fixed $\overline{D_i}(k)$ as specified in equation (29):

$$\sum\nolimits_{j\in I(k)} D_{jk} = c \cdot d_i(k) \;\forall\; k. \tag{37}$$

In the improbable case of $c_B(i) = 0 \;\forall\; i \in I(k)$, set all D_{ik} to c.

The whole procedure is called centrality based trust repartition.

If one single node k applies the technique above, we can compare its trustworthiness W_k in the case of homogeneous trust distribution with W_k' using centrality based trust repartition. We observe an increase of its trustworthiness using the technique, since most of the measured values in the plots of figure 7 are located above the dotted diagonal. However, as we can see in the left plot of figure 8, the difference between W_k' and W_k is rather small, that is the procedure does not completely change the trustworthiness W_k of the node k, proofing the stability of our metric against selfish behaviour of single vertices.

However, since the aim of maximising its trustworthiness W_k is identical for each node k, this procedure is recommendable for every vertex. If the whole community uses centrality based trust repartition, the mean trustworthiness shows an interesting behaviour visible in the right plot of figure 8.

For small c, the application of the technique does not cause big variation because the trust available for repartition is proportional to c and therefore small. In the limit of $c \to 1$, the distribution of the direct in-trust D according to equation (36) leads to a high percentage of edges $\vec{e}$ with full trust $D_{\vec{e}} = 1$. Examining the definition of trust along a path in equation (17),

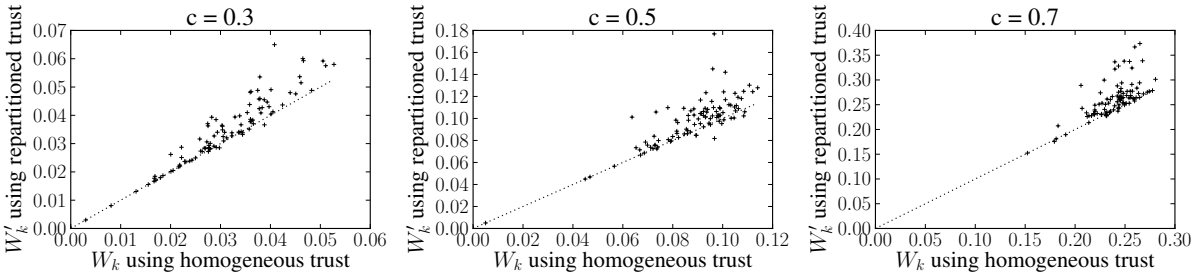

Figure 7: The black points are samples of nodes k, contrasting the trustworthiness W_k with homogeneous trust and W'_k if k uses centrality based trust repartition.
The results were obtained on graphs with $N = 100$ and $\lambda = 4$ for three different mean in-trust c. The dotted line signifies $W_k = W'_k$.

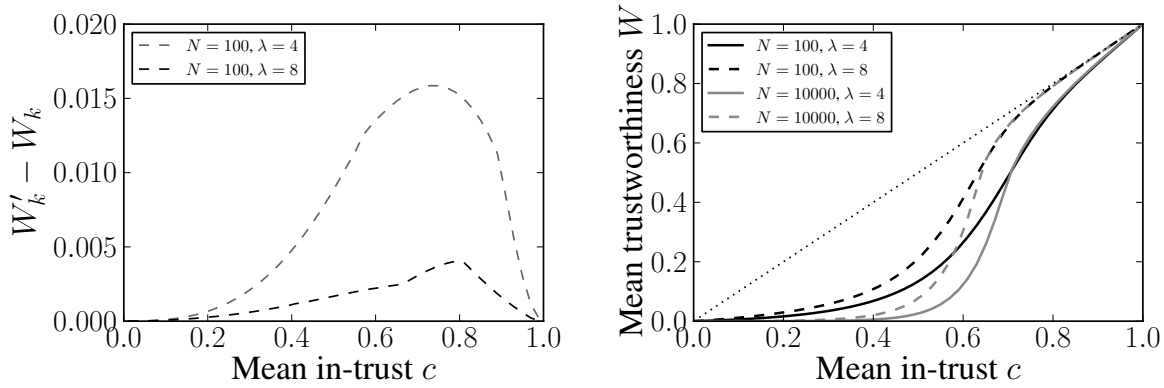

Figure 8: Left plot: Difference of trustworthiness W_k with homogeneous trust distribution and W'_k if k uses centrality based trust repartition. Right plot: Trustworthiness W if all nodes use centrality based trust repartition.

we recognise that full trust along the path P leads to $D_P = 1$. In the limit of $H_{ij} = 1 \; \forall \; i, j$, thus there exists a path with full trust between any pair of vertices, the trust according to the PTT metric yields[10]

$$T_{st} = \frac{\sum_{k \in I(t)} 1^2 \cdot D_{kt}}{\sum_{k \in I(t)} 1} = \frac{\sum_{k \in I(t)} D_{kt}}{d_i(t)} = \overline{D_i}(t) = c. \tag{38}$$

This limit is well visible in the right graph of figure 8 as the plot approaches the diagonal. At the limit defined in equation (38), the PTT trust T_{st} is independent of the geodesic distance between the nodes s and t. We loosely denominate the emergence of this cluster of nodes connected by path with full direct trust as 'trust percolation'[11].

[10] We neglect the fact that in the definition of the PTT metric, H_{ij} is determined on the subgraph $V' = V \backslash \{t\}$, thus H_{ij} depends on t. We will face this problem in section 5.4.2.

[11] As it will be seen below, even for infinite networks, the 'percolation' transition will only be abrupt if the average degree becomes very large. In general, it will not correspond to a second order transition and thus the comparison with actual percolation if very tenuous.

The achievement of 'trust percolation' requires $H_{ij} = 1 \; \forall \; i, j$, thus the existence of at least one incoming edge $\vec{e}$ with a direct trust value $D_{\vec{e}} = 1$ for each node is a necessary condition. We define $D_{max}(k)$ as the maximal incoming trust value for a given vertex k,

$$D_{max}(k) = \max \left[D_{ik} \text{ for } i \text{ in } I(k) \right]. \tag{39}$$

This value grows with the mean in-trust c and its expectation $\overline{D_{max}}$ can be used in a very simple model to approximate the mean trustworthiness W by

$$W \approx c \cdot \left(\overline{D_{max}} \right)^{\overline{g}}, \tag{40}$$

where $\overline{g}$ is the mean geodesic distance. The trust of each edge along the path from an arbitrary vertex to each in-neighbour of t can be roughly approximated as $\overline{D_{max}}$. The mean geodesic distance $\overline{g}$ is a characteristic length of this path and the factor c is the mean in-trust of the target t. In the limit of graphs with infinite size, thus $\overline{g} \to \infty$, the trustworthiness is

$$W = \begin{cases} 0, & \text{if } \overline{D_{max}} < 1, \\ c, & \text{if } \overline{D_{max}} = 1. \end{cases} \tag{41}$$

Since $\overline{D_{max}}$ is proportional to c as long as it is smaller than 1, we observe discontinuous behaviour of W while increasing c.

$D_{max}(v)$ depends on the centrality distribution of the neighbours $I(v)$ and obviously on c. In the following, we examine its average $\overline{D_{max}}$.

On random graphs with Poisson degree distribution, we can observe a very strong correlation between the betweenness centrality $c_B(v)$ and the product of in- and out-degree $d_i(v) \cdot d_o(v)$ as it can be seen in figure 9. As the distribution of c_B is not determinable analytically from the degree distribution but $d_i \cdot d_o$ is, we use this replacement in the following mathematical treatment.

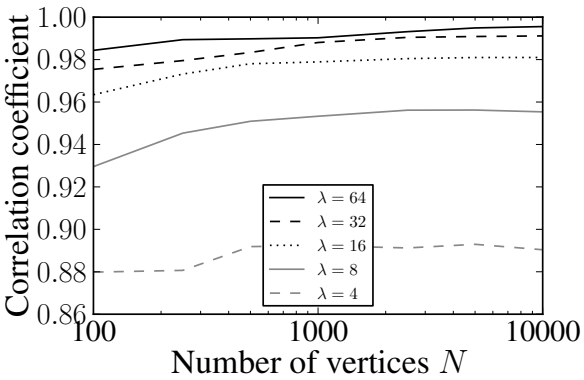

Figure 9: Correlation coefficient between $c_B(v)$ and $d_i(v) \cdot d_o(v)$ for different Poisson degree distributions.

5.4.1 Mathematical approach to 'trust percolation'

In order to understand the properties of the 'trust percolation', we work out the distribution of the maximal incoming trust D_{max} to get its expection $\overline{D_{max}}$.

$p_i(d)$ and $p_o(d)$ are the probability mass functions for the in-degree and the out-degree, although they were identical in all examples used until now. The centrality measure b is defined as $d_i \cdot d_o$. Thus the probability mass function for b can be calculated as follows

$$p_b(b) = \delta_{b,0} \cdot p_i(0) + \frac{1-\delta_{b,0}}{1-p_o(0)} \sum\nolimits_{d_o=1}^{b} p_o(d_o) \cdot p_i\left(\frac{b}{d_o}\right). \tag{42}$$

To understand the normalisation $1 - p_o(0)$, you should keep in mind that the source-vertex of an edge must not have out-degree $d_o = 0$. For a given vertex with in-degree d_i, the probability mass function of the maximum b_{max} of the centrality values b_i can be calculated:

$$\tilde{p}_b(b, d_i) = \left[\sum_{b' \le b} p_b(b')\right]^{d_i} - \left[\sum_{b' < b} p_b(b')\right]^{d_i}. \tag{43}$$

Looking back at our definition in equation (36) together with the fixed in-trust in equation (37), we can determine

$$D_{max} = \min\left[\frac{b_{max} d_i c}{\sum_i b_i}, 1\right] = \min\left[\frac{b_{max} d_i c}{b_{max} + S}, 1\right], \tag{44}$$

where S is defined as $\sum_i b_i - b_{max}$. The probability mass function for S is the $(d_i - 1)$-fold convolution of the modified distribution $p_{b_{max}}$, ensuring that $b_i \le b_{max}\ \forall\ i$,

$$p_{b_{max}}(b) = \frac{p_b(b) \cdot \Theta(b_{max} - b)}{\sum_{b' \le b_{max}} p_b(b')}, \tag{45}$$

$$p_S(S, d_i) = \left(p_{b_{max}}(b) \circ\right)^{d_i - 1}(S), \tag{46}$$

where Θ is the discrete unit step function, leading to $p_{b_{max}}(b) = 0\ \forall\ b > b_{max}$. Now the expectation of D_{max} can be written as

$$\overline{D_{max}} = \sum_{d_i=1}^{\infty} p_i(d_i) \cdot \overline{D_{max}(d_i)}, \tag{47}$$

with

$$\overline{D_{max}(d_i)} = c\tilde{p}_b(0, d_i) + \sum_{b_{max}=1}^{\infty} \sum_{S=0}^{d_i \cdot b_{max}} \min\left[\frac{b_{max} d_i c}{b_{max} + S}, 1\right] \cdot p_S(S, d_i) \cdot \tilde{p}_b(b_{max}, d_i). \tag{48}$$

The first summand represents the cases where $b(i) = 0$ for all in-neighbours. The sum is taken over D_{max} defined in equation (44), weighted with the probability mass functions of S and b_{max}. The meaning of this result will be discussed in the next section.

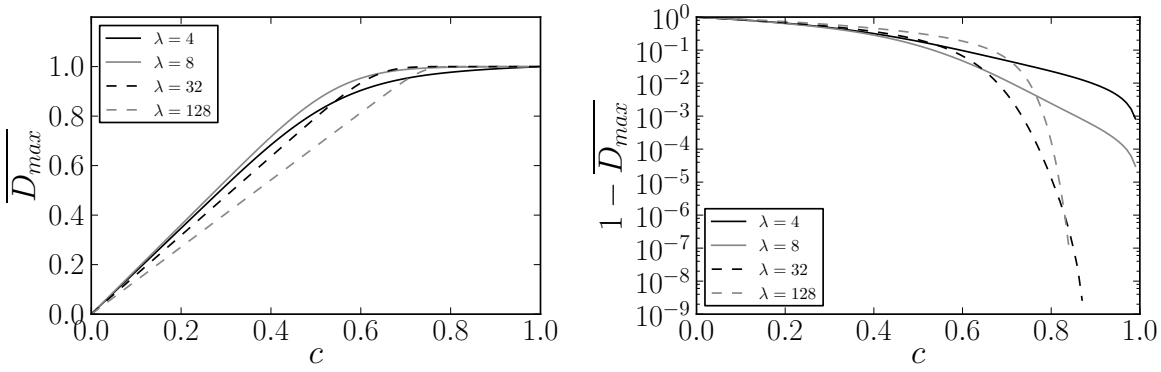

Figure 10: $\overline{D_{max}}$ dependent on c for different Poisson degree distributions.

5.4.2 Evaluation of 'trust percolation'

The numerical evaluating of equation (47) dependent on c for different parameters λ is shown in figure 10. In the log-log plot, the gap between $\overline{D_{max}}$ and 1 is shown. The value c where the graph disappears in the right plot (thus $\overline{D_{max}} = 1$) is thus equivalent to the saltus of the trustworthiness in equation (41).

Examining the derivation of the expectation of D_{max}, we discover that for a node v with one single in-neighbour, $D_{max}(v)$ is always equal to c. Therefore, the transition in equation (40) will be discontinuous only in the limit $p_i(0) \to 0$ which is achieved only for $\lambda \gg 1$ in the case of graphs with Poisson degree distribution. This explains why the trustworthiness does not reach c in the case of $\lambda = 4$ in figure 8.

We assumed in equation (40) that the direct trust $D_{\vec{e}}$ of the edges along the paths used to calculate the Pervasive Trust Transitivity in our simulations is mainly given by $\overline{D_{max}}$, since always the best path is used. To support this assumption, we plot $\overline{D_{max}}$ over c and add a density plot of the distribution of $D_{\vec{e}}$ of the edges that were passed by the paths used to calculate T. The correspondence is visualised in the left graph in figure 11, confirming this assumption.

We should keep in mind that we assumed the existence of one single incoming edge $\vec{e}$ per node with D_{vece} to reach 'trust percolation'. But during the calculation of T, we mainly work on a subgraph, thus some of these edges may be filtered out. This will decrease the resulting trust T and therefore the mean trustworthiness W, especially if the mean in-degree $\overline{D_i}$ is small.

To test the application of our approach on networks with finite size, we define

$$W_{theo} = c \cdot \left(\overline{D_{max}}\right)^{\overline{g}} \tag{49}$$

with the mean shortest distance taken from equation (4) as

$$\overline{g} = \frac{\log(N)}{\log\left(\overline{d_o}\right)}. \tag{50}$$

This is a simplification assuming that the length of all the paths considered in calculating the Pervasive Trust Transitivity is equal to the mean geodesic distance and all edges along the paths

have same weights $\overline{D_{max}}$. We can use the right plot in figure 11 to compare W_{theo} with the mean trustworthiness W determined in the simulations. We can see that W_{theo} overestimates W for small c. This can be explained looking at the left graph: In our simulations, the trust along the paths used for the calculation of the PTT trust T was a little smaller than $\overline{D_{max}}$. To calculate the trust, multiple trust values are multiplicated leading to a significantly lower result T. Contrasting the both plots, the match between simulation and theory happens at nearly the same value of c. The lack of accordance for lower c is caused by the assumption that the trust along the paths is mainly determined by $\overline{D_{max}}$. This is true only in the limit $\overline{D_{max}} \to 1$, where the shorter distance of the direct paths loses its relevance.

Hence our mathematical description is an adequate approach to the 'trust percolation', as desired.

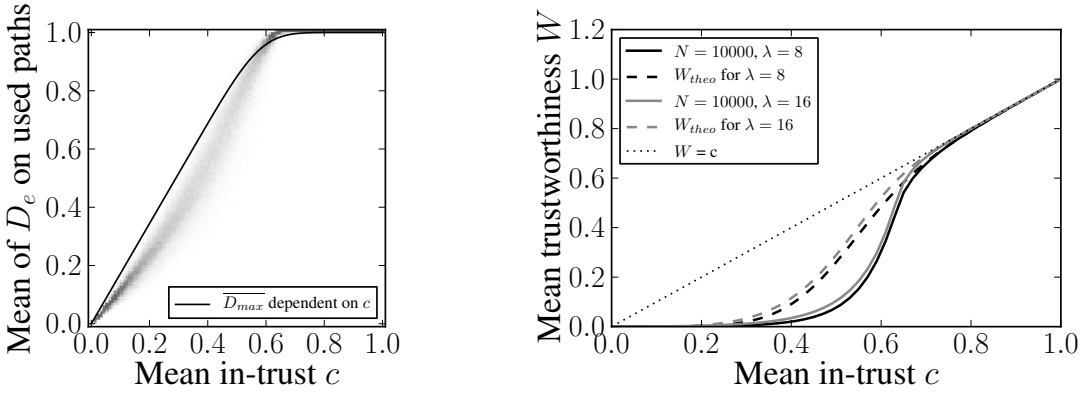

Figure 11: Left plot: Density plot of used edges in the calculation of T with a plot of $\overline{D_{max}}$ dependent on c. Right plot: Comparison between simulation and W_{theo}, defined in equation (40).

6 Conclusion

In this work, we motivated the necessity of developing a novel metric for computing indirect trust on networks and introduced the so-called Pervasive Trust Transitivity (PTT) metric based on the concept on transitivity. The result is a personalised metric, which makes no assumptions about the network topology and takes into account all necessary information present in the network. This proposed metric avoids several of the problems which exist in alternative attempts of providing such a metric in the literature so far. In particular, it does not consider loops in the network, and does not rely on the introduction of *ad hoc* damping factors.

Since the metric is defined by simple equations, we were able to investigate some of its properties analytically and studied its behaviour in simple examples. Further we investigated the robustness of the metric against selfish attitudes of agents. Using the concept of betweenness centrality to differentiate between more important vertices, we considered a global strategy in which the direct trust values are proportional to the centrality of the source vertex, but the average in-trust of the vertices is the same for all vertices. We showed that for random networks, the PTT metric leads to average trust values which never exceed the average in-trust set beforehand. This means that the selfish strategy is not capable of fooling most nodes on the network showing the robustness against manipulation of our metric. Furthermore, using

centrality based trust repartition, a non-linear behaviour was identified, where, if the average direct trust is high enough, a percolating network of full trust will be formed, which increases drastically the average trust in the network.

As future work, it would be interesting to study the effect of other, more realistic degree distributions, such as scale-free networks, and to study the role of authorities (hubs) in the propagation of trust. More sophisticated ways of distributing the direct trust D could also be considered, possibly omitting the strict condition that all vertices have the same in-trust. To expand the field of application, it is possible to replace the factor D_{kt} in the definition of the Pervasive Trust Transitivity in equation (24) by another value. That may be interesting if agents want to get a trustworthy opinion about an attribute different from trust. A desirable practical test of the quality of our metric would also be the application to real networks such as the web of trust of “Pretty Good Privacy”.

References

[1] Amaral, L. A. N., Scala, A., Barthélémy, M. and Stanley, H. E., Classes of small-world networks, *Proc. Natl. Acad. Sci* USA **97**, 11149 - 11152 (2000).

[2] Huberman, B. A., *The Laws of the Web*, MIT Press, Cambridge, MA (2001).

[3] Moreno, J. L., *Who shall survive?*, Beacon House, Beacon, NY (1934).

[4] Martinez, N. D., Artefacts or attributes? Effects of resolution on the Little Rock Lake food web *Ecological Monographs* **61** 367-392 (1991).

[5] White, J. G., Southgate, E., Thompson, J. N. and Brenner, S., The structure of the nervous system of the nematode C. Elegans, *Phil. Trans. R. Soc. London* **314**, 1 - 340 (1986).

[6] Velminski, W. *Leonhard Euler. Die Geburt der Graphentheorie* Kulturverlag Kadmos, Berlin (2008).

[7] Farkas, I. J., Jeong, H., Vicsek, T., Barabási, A.-L., and Oltvai, Z. N., The topology of the transcription regulatory network in the yeast, Saccharomyces cerevisiae, *Physica A* **381**, 601-612 (2003)

[8] Zimmermann, P., *The Official PGP User's Guide*. MIT Press (1995).

[9] Jøsang, A., An Algebra for Assessing Trust in Certification Chains, *Proceedings of the Network and Distributed Systems Security Symposium*, The Internet Society, San Diego (1999)

[10] Brinkmeier, M., and Schank, T., Network Statistics, in *Network Analysis* by Brandes, U. and Erlebach, T. (editors), Springer-Verlag Berlin Heidelberg (2005)

[11] Erdős, P. and - Rényi, A., On random graphs, *Publicationes Mathematicae* **6**, 290-297 (1959)

[12] Milgram, S., The small world problem *Psychology Today* **2**, 60-67 (1967).

[13] Travers, J. and Milgram, S., An experimental study of the small world problem, *Sociometry* **32**, 425-443 (1969).

[14] Schputt, D. J., Rossel, A. g. D., The Tinky-Winky algorithm on complex non-deterministic high-clustered social networks, *WYSIWYG-YCGIIYRW* **6a104**, Adolf-Spieß-Verlag Kranichstein (2011).

[15] Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., *Introduction to Algorithms*, MIT Press, 2nd edition (2001).

[16] Koschützki, D. et al., Centrality Indices in *Network Analysis* by Brandes, U. and Erlebach, T. (editors), Springer-Verlag Berlin Heidelberg (2005)

[17] Brin, S. and Page, L., The anatomy of a large-scale hypertextual Web search engine, *Computer Networks and ISDN Systems* **30**, 1-7 (1998), 107 - 117.

[18] google.com - Corporate Information - Technology Overview. http://www.google.com/corporate/tech.html

[19] Schollmeier, R., A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications, *Proceedings of the First International Conference on Peer-to-Peer Computing*, IEEE (2002).

[20] Kamvar, S. D., Schlosser, M. T. and Garcia-Molina, H., The EigenTrust Algorithm for Reputation Management in P2P Networks, *WWW '03: Proceedings of the 12th International Conference of the World Wide Web*, ACM Press, pp 640-651 (2003).

[21] Walter, F. E., Battiston, S. and Schweitzer, F., Personalised and Dynamic Trust in Social Networks, *Proceedings of the third ACM conference on Recommender systems*, NY (2009).

[22] Dijkstra, E. W., A note on two problems in connexion with graphs, *Numerische Mathematik* **1**, 269 - 271, (1959).

[23] SciPy and Numpy - scientific computing with python.
http://www.scipy.org/

[24] Matplotlib - python 2D plotting library.
http://matplotlib.sourceforge.net/

[25] graph-tool by Tiago de Paula Peixoto - python module for statistical analysis of graphs.
http://graph-tool.forked.de/

Typesetting with LaTeX 2_ε on Padua.