C. Fernandez-Gago, I. Agudo, and J. Lopez, "Building Trust from Context Similarity Measures", *Computer Standards & Interfaces, Special Issue on Security in Information Systems*, vol. 36, pp. 792-800, 2014.
http://doi.org/10.1016/j.csi.2013.12.012
NICS Lab. Publications: https://www.nics.uma.es/publications

Building Trust from Context Similarity Measures

Carmen Fernandez-Gago, Isaac Agudo and Javier Lopez
Network Information and Computer Security (NICS) Lab,
University of Málaga, 29071, Málaga, Spain
{mcgago,isaac,jlm}@lcc.uma.es

Abstract

Trust is an essential feature of any system where entities have to collaborate among them. Trust can assist entities making decisions about what is the best entity for establishing a certain collaboration. It would be desirable to simulate behaviour of users as in social environments where they tend to establish relationships or to trust users who have common interests or share some of their opinions, i.e., users who are similar to them to some extent. Thus, in this paper we first introduce the concept of context similarity among entities and from it we derive a similarity network which can be seen as a graph. Based on this similarity network we define a trust model that allows us also to establish trust along a path of entities. A possible applications of our model are proximity-based trust establishment. We validate our model in this scenario.

1 Introduction

Online communities are becoming more and more popular as it is shown by the growth on the number of users. The types of relationships that are taking place in an online community might vary. Sometimes users have to interact among them and some other times they might only be interested in knowing the opinion of others about a certain issue. Due precisely to the growth of these networks there are a huge variety of users registered, not all of them honest or reliable enough. When information is to be shared or a user is interested in knowing the opinions of others, uncertainty might be a problem. It would be desirable to have a tool that aid users overcome this uncertainty, and it is then when trust can play an essential role. The question that arises is to whom to trust in an online community and how to establish that trust. Moreover, this trust cannot be a general concept or a general value that can be obtained by a user and valid for all its functions in the network. Trust has to depend on specific contexts where the user or entity is involved. It is a well-known example that establishing that one might trust a dentist to take care of your teeth but not for repairing your car. There is not a consensus on the definition of trust. Many different definitions can be found in the literature, depending in most of

the occasions, on the type of system where they are applied and on the problem they address. However, it is becoming an essential feature to be considered in any system where entities have to collaborate among them. Trust can assist the decision-making process by determining which is the most suitable entity to collaborate with in a system.

In a social setting, people tend to trust those who are close to them in taste, opinion or hobbies, i.e., those who they believe are similar to them. It becomes then natural to try to simulate this behaviour in a computational setting. Nowadays, social networks are experiencing the growth of the so-called recommender systems [22, 13]. These systems use the opinions of members of a community to help others identifying what products or information they might be interested in following other users' opinions or tastes.

In this paper we intend to establish trust in computational settings by following a similar approach as the one users adopt in social settings. That is, we are interested in finding which users are similar to a given one for a specific context in an online community. Following this similarity between users a function for deriving trust is defined. As a first step towards determining similarity between users we create similarity networks that can be represented by a graph. Then, we derive trust by using the similarity graph. This similarity graph will have a trust graph associated to it. Once the trust graph is also established we calculate trust by following a specific trust propagation model [2], and the similarity values we have calculated earlier.

We introduce the concept of similarity networks and design a formal model for them. Similarity between entities or users is defined with respect to a specific context that is relevant to the environment where these entities are. We define trust on top of this concept. This differs from another attempts to relate trust and similarity [29] where the assumption is that certain correlation already exists between trust and similarity and it is shown through some empirical examples. On the contrary, our only assumption is that users are related through a numerical value that defines how far in terms of a specific context they are from each other. This numerical value is a distance in our case. Then, similarity can be calculated by using this distance and a certain threshold that determines the size of the network. It could be the case when there are users who are not related at all since the similarity network depends on the threshold.

There are many cases where our model can be used and proven useful. We have presented two cases scenarios, a profile matching in social networks and in an emergency scenario where the entities involved are interested the most suitable entity that could deal with the situation in a satisfactory manner.

The structure of the paper is as follows. Section 2 reviews the related work. The first step towards building our trust model is the definition of similarity networks, which is done in Section 3. The trust model we propose is based on a specific type of trust models called *Propagation models*, which are presented in Section 4. The proposed model is introduced in Section 5. Some use cases, together with the analysis we performed and the validations of them are presented in Section 6 and 7 respectively. Section 8 concludes the paper and outlines the future work.

2 Related Work

Many definitions of trust have been provided along the years. The concept is complex and it spans across several areas such as psychology, sociology, economics, law, and more recently, computer science. The vagueness of this term is well represented by the statement 'trust is less confident than know, but also more confident than hope' [18].

Gambetta [6] defines trust as 'a particular level of the subjective probability with which an agent will perform a particular action [...] in a context in which it affects our own action'. McKnight and Chervany [17] explain that trust is 'the extent to which one party is willing to depend on the other party in a given situation with a feeling of relative security, even though negative consequences are possible'. For Olmedilla et al. [21], 'trust of a party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X)'. Ruohomaa and Kutvonen [23] state that trust is 'the extent to which one party is willing to participate in a given action with a given partner, considering the risks and incentives involved'. Finally, Har Yew [9] defines trust as 'a particular level of subjective assessment of whether a trustee will exhibit characteristics consistent with the role of the trustee, both before the trustor can monitor such characteristics (or independently of the trustor's capacity ever to be able to monitor it) and in a context in which it affects the trustor's own behavior'. Moyano et. al [20] define trust as a subjective, context-dependent property that is required when (i) two entities need to collaborate (i.e. there is a dependence relationship between them and there exists the willingness to collaborate), but they do not know each other beforehand, (ii) and when the outcome of this collaboration is uncertain (i.e. entities do not know if they will perform as expected) and risky (i.e. negative outcomes are possible). In this situation, trust acts as a mechanism to reduce the uncertainty in the collaboration and to mitigate the risk. As risk increases (either the probability or the impact of negative outcomes), trust becomes more crucial.

Similarity graphs have been used in many applications such as for example matching text in documents [3, 26]. Finding similarity between texts becomes important for instance for document clustering or web searching. Jeh and Widom [10], however, proposed an approach called *SimRank* that could be applied to any domain where similarity can be applied object to object. Our approach to define similarity graphs is context-independent but among other differences with this approach we consider also the case where more than one similarity graph or network can be used.

The trust model we introduce here can be seen as a propagation trust model, based on the model presented in [2], where the authors use a sequential and a parallel operator in order to compute trust along a path. These types of models assume that trust and reputation values exist regardless how to compute them. They provide ways to calculate indirect trust relations by propagating these trust values along trust chains or trust paths. Advogato [15] is also an example of a trust propagation model that computes a reputation flow through a network

where members are nodes and edges are referrals between nodes. Appleseed [27] is another trust propagation model that uses local groups in order to infer trust in them. Subjective logic [12] uses a discounting operator to compute opinions along different trust paths and a consensus operator to combine them into a final opinion.

The main contribution of this paper is the use of similarity for inferring trust. There are some approaches that consider the fact of how similar entities are for the purpose of deriving trust, however aiming in most of the cases at trust-based recommendation systems. In our approach, the purpose is not on the recommendation phase, although in a step beyond it could be used for this as well. The goal of Recommender Systems [22] is to suggest to the users of a system items they might be interested in. A particular type of recommender systems are those based on Collaborative Filtering (CF) [5], which try to automatize the search for similar users by taking into account their opinions and ratings on the items. However, these systems present some weaknesses that Massa et al. try to overcome in [16] by enhancing Recommender Systems by using trust information. They propose the so-called Trust-aware Recommender System. The idea of this enhancement is not to search for similar users as CF do but to search for trustable users by exploiting trust propagation on the trust network. Thus, if a user A trusts users B and C, A will be recommended the items that B and C appreciate.

There are in the literature some other trust-based recommender systems [7, 28, 25] that assume similarity is already present when deriving trust between users. The relationship between similarity and trust is investigated in [29]. The authors show that these relationships exit. They made the hypothesis that there is a dependency in between these two concepts and show some empirical evaluations on the FilmTrust social network [1]. The main difference between this work and our proposal is that we do not assume the existence of similarity between users. On the contrary, we propose a model for calculating it according to certain criteria such as distance in between these users for a specific context. In [8] the authors show that the existing relationship is not only in between trust and similarity but more specifically in between trust and profile similarity, which includes many aspects of similarity with other users and not only an overall similarity.

3 Similarity Measures and Networks

As we have pointed out in Section 2 there is no a unique definition of the concept of trust. We observe that for these definitions there are several factors that might affect trust. The context where the entities are interacting is one of them as it can also be the task they are to perform. Together with the importance of the context or task to be carried out another important factor to be taken into account is how similar trustor and trustee are or how similarly they are behaving in a specific context.

In real live interactions, a natural mechanism to build trust upon is simi-

larity. People tend to trust other people that behave like them, whereas the opposite is not generally true, i.e. we do not necessarily distrust people that behave in a different way. Our intention is to simulate this behaviour that is common in social settings. We thus present a trust model that bootstraps trust as a function that considers as input the similarity of two entities regarding a particular context. A specific context might also be influenced or compose of many different factors or characteristics that can be measured and that are meaningful for the process of deriving trust. For example, in a social network environment where users exchange pictures or statements one of this influencing factors could be 'liking a picture'.

We need to determine determine similarity between users for these specific factors and how similarity networks are derived. First, we need to introduce the essential concepts and methods.

Let us assume a network where different entities establish certain relationships. We denote the set of entities of the network or system by $\mathcal{E}$. Each entity e in $\mathcal{E}$ can measure several factors that influence their behaviours. We will call these factors characteristics.

Definition 1 (Measurable Characteristic)**.** *A measurable characteristic is a metric space* (C, d) *with a minimal element denoted by* 0*.*

This definition identifies the Characteristic itself with its measure set. As an example, if we consider the temperature as a measurable characteristic in our system, the corresponding C in the metric space will be all the values that correspond to different measurements of temperature in a given setting for a certain entity. That is, a subset of the real numbers, i.e., $C \subset \mathbb{R}$, where each element in C is represented by c and corresponds to the different values given for a certain measurable characteristic. Since the measures are referred to the same characteristic the only thing that distinguish them is in what instant they were taken. Thus, we can say that measures depend on time. Then, a measure, c is really the output of a function that maps a time instant into an acceptable value from the measure set, i.e. $f : T \longrightarrow C$, where T is the time domain and $f(t) = c$.

Let $\mathcal{C}$ be the set of measurable characteristics in a given system, where its elements are C_j for $1 \leq j \leq m$ and m is the total number of measurable characteristics.

Definition 2. *We define the Measured Context,* $\overrightarrow{C_e}(t)$*, of an entity* e *as the vector of all the measured characteristics considered for* e *in a specific time instant,* t*. That is,* $\overrightarrow{C_e}(t) =< c_e^i(t) >$ *for* $i = 1 \cdots m$*.*

$\overrightarrow{C_e}(t)$ can also be seen as a function, $\overrightarrow{C_e}(t) : T \longrightarrow \prod_{i=1}^{m} C_i$

In order to calculate how similar two entities are we have to take into account how far these measurable characteristics are for them in different situations. In a one dimension setting and in a more realistic setting as it could be a multidimensional setting

3.1 One Dimension Setting

Let us assume first the case where we need to determine how similar two entities are with respect to only one characteristic, i.e., $\overrightarrow{C_e}(t)$ is a one-dimensional vector $\overrightarrow{C_e}(t) = c_e(t)$.

In order to determine how similar two entities are, we use the distance in the measurable characteristic set. This distance represents how close two measures are at a given instant of time. The distance between two entities e and f in time instant t for a common measurable characteristic, $C \in \mathcal{C}$ is defined as $d(c_e(t), c_f(t))$. For simplicity, if the time and the characteristic are known we will denote this distance as $d(e, f)$.

Definition 3. *We define $\mathcal{G}_\delta \subset \mathcal{E} \times \mathcal{E}$ as the set of all pairs of entities such as the distance between them is lower or equal than a given threshold δ, i.e.,*

$$\mathcal{G}_\delta = \{(e, f) \in \mathcal{E} \times \mathcal{E} \text{ such that } d(e, f) \leq \delta\}$$

The closer two entities are the more similar they are. Therefore,the smaller the distance the higher the similarity is. Formally, we can define similarity as follows:

Definition 4. *A similarity measure for a set $\mathcal{G} \subset \mathcal{E} \times \mathcal{E}$ is a function, Δ associated to a distance, d, where $\Delta : \mathcal{G} \to [0, 1]$ verifies the following properties:*

- $\Delta(e, e) = 1$
- $\Delta(e, f) = \Delta(f, e)$
- $\Delta(e, f) \leq \Delta(e, g)$ *if* $d(e, f) \geq d(e, g)$

The idea of deriving a similarity function, Δ, from a distance is not new. In the literature we can find many approaches of which we can highlight the following two: $\Delta = e^{-d}$ [24] (e is the exponential function) or $\Delta = \frac{1}{1+d}$ [4].

We propose a choice for this function that depends on a specific value δ. This δ could be seen as a way to bound the distance. For any $\delta > 0$, the function $\Delta_\delta : \mathcal{G}_\delta \to [0, 1]$ is defined as follows:

$$\Delta_\delta(e, f) = 1 - \frac{d(e, f)}{\delta}$$

All the three possible definitions for similarity functions work for unbounded distances, i.e, when it holds that for any $\lambda > 0$ there are two elements x and y such as $d(x, y) > \lambda$. The main difference between our definition and the two other ones is that in our case δ is an upper bound for the distance, i.e. for any $e, f, e', f' \in C$ such as $d(e, f) > \delta$ and $d(e', f') > \delta$, $\Delta(e, f) = \Delta(e', f') = 0$. This means that our similarity measure does not differentiate when the distance is beyond the δ bound. In this case the similarity is set to 0 interpreting then that the entities are not similar at all.

However, in any of the different cases it for the three similarity functions the following holds,

$$\lim_{d \to \infty} \Delta = 0$$

3.2 Multidimensional Setting

In Section 3.1 we have considered only the case where one measurable characteristic is considered. However, in real scenarios this might not be the case. It might occur that, for example, in a social network where an entity has to interact with certain users one needs information about them related to more than one characteristic such as *movies I like* or *museums I like*. Even though the two characteristics might not seem to be very related, the user building his graph of trust might be interested in these two specific characteristics.

Let us assume a scenario where there are more than one measurable characteristic to be considered, $c_j(t)$ for a time instant t and $j = 1, ..., m$.

For those distances d_i that are bounded we need to normalize them by dividing by the upper bound δ_i, i.e., $\overline{d_i} = d_i \delta_i^{-1}$. For the distances that are not bounded we have to use an equivalent bounded distance, e.g., $\overline{d_i} = \frac{d_i}{d_i+1}$.

We can define a new distance for the new combined characteristic $\widehat{C} = \prod_{j=1}^{m} C_j$,

$$\begin{array}{rccc} \widehat{d}: & \widehat{C} \times \widehat{C} & \longrightarrow & [0,1] \\ & (\{c_i\}_{i=1}^m, \{c_i'\}_{i=1}^m) & \longmapsto & \displaystyle\sum_{i=1}^{m} \alpha_i \overline{d_i}(c_i, c_i') \end{array}$$

where $\sum_{j=1}^{m} \alpha_j = 1$. α_i are representative of the relevance of each characteristic as all the distances have been normalized.

Once a new measurable characteristic, $(\widehat{C}, \widehat{d})$, has been defined we can derive a similarity measure by using Definition 4. Even though the values for α_i have been normalized in order to represent meaningful characteristics it could be that as the new similarity is defined as a summation the result goes beyond the bound. Therefore, we might need to be more strict in these cases and imposing a lower bound for each characteristic instead of the one defined above.

3.3 Similarity Networks

The entities in the network will form a network where they have established some relationships based on the similarities that can be calculated as described in Sections 3.1 and 3.2. From these definitions of similarity we can build the similarity network, which can be defined as follows.

Definition 5. *The similarity network for a characteristic C in instant $t \in \mathcal{T}$ with threshold δ, $\mathbf{S}^c_{t,\delta}$, is defined as a the set of elements $(e, f, \Delta_\delta(e, f)) \in \mathcal{E} \times \mathcal{E} \times \mathbb{R}$ where $\Delta_\delta(e, f) > 0$, is the proposed similarity measure for characteristic C.*

Example Let us a consider a similarity network where their users are connected and a different value for δ is set. Thus, if the vertical axis corresponds

to the values of δ, we can see in Figure 1 how the similarity network becomes smaller as δ increases. The edges of the connected nodes in the network are represented in the figure by a thicker line.

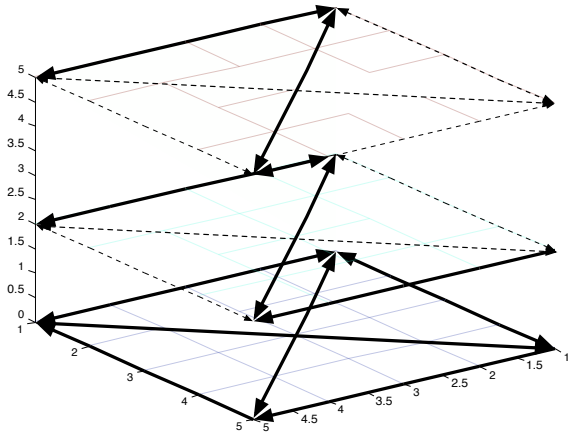

Figure 1: Similarity Graph for Different Values of δ

3.4 Similarity Networks and Graphs

In the previous section we defined how a similarity network can be obtained. According to this definition a similarity network can be seen as a labelled graph. If we consider a graph in which edges represent some kind of similarity or special relationship, we can easily transform it into a characteristic, forcing the similarity network to be always a sub-graph of it.

Let us consider first an example of a graph such as in Figure 2:

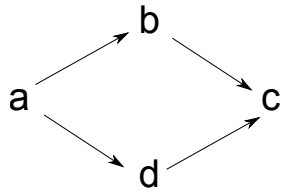

Figure 2: Simple sample graph

We can construct the following characteristic $C = \{Aa, Ab, Ac, Ad\}$, the distance in C is the graph distance, defined as the minimal number of hops needed in order to reach one node from another one, i.e. $d(Aa, Ab) = 1$, $d(Aa, Ac) = 2$, $d(Ab, Ac) = 1$, $d(Aa, Ad) = 1$ and $d(Ad, Ac) = 1$.

We can now use Δ_2 in order to enforce that two entities are similar with regards to the defined characteristic. In this case $\Delta_2 = 0$ if the nodes are

connected by more than 1 vertex and $\Delta_2 = 0.5$ if they are neighbours. Using this similarity measure in a multidimensional setting with $\widehat{\Delta}$ will result in a similarity graph that is in fact a sub-graph of the initial graph.

Moreover, our model also supports mobility of nodes because the characteristics are measured in time.

4 Propagation Trust Models

In Section 3 we have presented a way to determine similarity networks. These networks can be seen as a labelled graph. Taking advantage of the graph structure it would be very convenient to have an appropriate model for deriving trust that use then as the underlying method. Trust in a virtual community can be modelled by using a trust graph, that is, a graph where the vertices are the entities of the network and the edges are the trust relationships established between two entities.

The trust model that can fit into these requirements are propagation models. Propagation models often assume that several trust relationships have already been established and quantified, although this is not always the case. They aim to create new trust relationships by disseminating the trust values information to other entities. For example, Advocato [14] is a reputation model that allows users of the community to provide a ranking for other users. However, it is also a propagation model, since it allows computing a reputation flow through a network where members are nodes and edges are referrals between nodes. New trust values are often computed by means of operators, and in several models, we find two of them: a concatenator and an aggregator. The former is used to compute trust along a trust path or chain, whereas the latter aggregates the trust values computed for each path into a final trust value. For example, in [2] the authors use a sequential and a parallel operator in order to compute trust along a path. Subjective logic [11] uses a discounting operator to compute opinions along different trust paths, and a consensus operator to combine them into a final opinion.

In order to define the trust graph it has to be refereed to a certain context. This is what we will define next as trust domain.

Definition 6 (Trust Domain)**.** *A trust domain is a partially ordered set* $(TD, <, 0)$ *where every finite subset of* TD *has a minimal element in the subset and 0 represents the minimal element of* TD*.*

A particular trust domain is the interval $[0, 1)$, where 0 means that there is no trust and 1 means full trust. By 0 we do not mean distrust but absence of trust evidences. Distrust can be modelled as trust or believe that certain entity will behave dishonestly.

5 A Similarity-based Trust Model

Let us assume a network where the nodes interact among them. Trust can be modelled by using a directed graph where the vertices are the entities and the edges correspond to the trust relationships between the entities that the edge links.

Our definition of trust states that trust can be defined as the level of confidence that an entity s places on another entity t for performing a given task in a proper and honest way. The confidence level may vary depending on the task, or as we called it in previous sections a measurable characteristic. Assuming that the level of confidence is a real number and that for each task there is only one associated trust value in our reasoning system, the trust graph is a weighted digraph.

Trust can be tailored to tasks [19] but in our case we use a broader concept, Context. The context includes all relevant information for the trust making decision.

In order to define the function for deriving trust we need to use the definition of trust statement.

Definition 7 (Trust Statement)**.** *A trust statement is an element* $(Trustor, Trustee, c, t, Trust_{c,t})$ *in* $\mathcal{E} \times \mathcal{E} \times \mathcal{C} \times \mathcal{T} \times \mathcal{TD}$*, where* $\mathcal{E}$ *and* $\mathcal{C}$ *are the sets defined above and* $\mathcal{TD}$ *is a Trust Domain whose elements* $Trust_{c,t}$ *are calculated as a function,* f *such that* $Trust_{c,t} = f(\Delta_{c,t}(e, f))$*.*

Note that $\Delta_{c,t}(e, f)$ is the similarity measure corresponding to each $C \in \mathcal{C}$.

One important property of function f is that it is monotonically increasing. This means that the smaller the distance in between elements is the more similar they are and therefore the value of trust that one of them places on the other one is higher as well.

5.1 Trust Evaluations

In Definition 7 we have introduced how trust is calculated in between two entities by means of the similarity measures. However, in a trust graph representation, as we are considering, trust should be calculated along a path. Therefore, we need to use specific functions that allow us to calculate these trust values. For this reason we introduce the notion of trust evaluation.

Definition 8. *A trust evaluation is a function* $\mathcal{F} : \mathcal{E} \times \mathcal{E} \times \mathcal{C} \times \mathcal{T} \longrightarrow \mathcal{TD}$ *where* $\mathcal{E}$*,* $\mathcal{C}$*,* $\mathcal{T}$ *and* $\mathcal{TD}$ *are the sets defined above.*

We will concentrate on local trust evaluations that can be decomposed into Sequential and Parallel Trust Functions. This decomposition will allow us to operate with the evaluations that could be applied to a variety of scenarios. For convenience we will assume that the elements of $\mathcal{E}$ and $\mathcal{C}$ are given and we can omit them.

Definition 9 (Sequential Trust Function)**.** *A sequential trust function is a function,* $f : \bigcup_{n=2}^{\infty} \overbrace{\mathcal{TD} \times \cdots \times \mathcal{TD}}^{n} \longrightarrow \mathcal{TD}$*, that calculates the trust level associated to a path or chain of trust statements, such that* $f(v_1, \ldots, v_n) = 0$ *if, and only if,* $v_i = 0$ *for any* $i \in \{1, \ldots, n\}$*, where* $v_i \in \mathcal{TD}$ *and* $\mathcal{TD}$ *is a trust domain.*

Each path of trust statements in a graph G is represented as the chain, $e_1 \xrightarrow{v_1} e_2 \xrightarrow{v_2} \cdots \xrightarrow{v_{n-1}} e_n \xrightarrow{v_n} e_{n+1}$, where e_i are entities in $\mathcal{E}$ and v_i are respectively the trust values associated to each statement.

Definition 10 (Parallel Trust Function)**.** *A parallel trust function is used to calculate the trust level associated to a set of paths or chains of trust statements. It is defined as,* $g : \bigcup_{n=2}^{\infty} \overbrace{\mathcal{TD} \times \cdots \times \mathcal{TD}}^{n} \longrightarrow \mathcal{TD}$*, where* $\mathcal{TD}$ *is a trust domain and*

1. $g(z_1, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_n) = g(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)$ *if* $z_i = 0$
2. $g(z) = z$

The sequential and parallel trust functions verify certain properties as stated in [2]. Some examples of functions that verify those definitions are the following:

- **Sequential**. Minimum, Product, Maximum, Mean.
- **Parallel**. Minimum, Maximum, Mean, Median.

$$\mathcal{F} = g(\{p_i\}_{i=0}^{n}) \text{ where } p_i = f(\{v_j\}_{j=0}^{m})$$

That is, we first apply the sequential function f to each path connecting entities e and f, obtaining the partial values p_i. Finally, we apply the parallel function to the set of values $\{p_i\}_{i=0}^{n}$.

6 Use Cases

The model presented is this paper can be easily adapted to specific scenarios. In particular, we are going to consider an emergency scenario where trust is established by a proximity-based relationship. The experimental results from its validation is going to be shown in Section 7.

6.1 Proximity-based trust establishment

By proximity-based trust establishment we understand that two entities are to trust each other if their locations are close enough. Location can be easily modelled as a characteristic and the similarity measure will indeed establish a kind of threshold for distance between two entities that are to trust each other.

Location can be encoded as a GPS coordinate or as descriptive areas. In any case, it is important to define the distance associated to the characteristic set. For the GPS coordinates, the distance can be measured as the linear distance or it could measure actual distances using specific routes. For descriptive areas, in case they can be located in a map we can measure the distance between their gravity center or use a similar technique and, in case they are abstract locations we can make use of discrete distances that represent levels of proximity, e.g. 1 for close, 2 for far and 3 for very far. The precision will determine the granularity of the corresponding similarity networks.

Proximity on its own can be used to establish trust but it can also be combined with other characteristics. If there are two different characteristics C_1 and C_2, being C_2 the location, we can use the $\widehat{\Delta_\delta}$ similarity function in such a way that no matter how similar two nodes are regarding C_1, they will not be connected in the similarity network unless some threshold for the location distance, dependant on δ, holds. When they are below the threshold the other characteristics, C_1, could make the difference.

6.1.1 Emergency Situation

Let us consider a situation where an emergency comes up, for instance, someone is suffering a heart attack. It would be desirable to find sanitary staff that can attend the ill person as soon as possible. It might be that there are more than one person around and some of them are capable to deal with the situation. We would be however interested in finding the most suitable person close by.

In this kind of situation we are assuming the users are connected through a type of social network for emergencies. Each of the entities involved in this social network carries with them a device (for instance an Android mobile device) that allows them to connect to a central server that is in charge of gathering values related to position, similarity and trust. This same central server will also be in charge of providing the information related to these values that the entities might request.

In our scenario we are assuming someone suffering from a heart attack. The affected entity would not share any information related to his health records under normal circumstances, but in this case he decides to trust all suitable entities in the surroundings. Our model will help him determine what is the most trusted entity, and therefore the most suitable one, for attending him on this emergency.

The measurable characteristics we are considering are the location and the medical speciality of the entity in the surrounding. The medical specialities that are going to be relevant are *cardiologist*, *specialist other than cardiologist*, *general practitioner*, *nurse* and *non-medical staff*. Let the entity that is requesting medical aid be e and the entities holding medical specialities f. The characteristic is defined as follows:

$$c_t(e) = \begin{cases} 1 & if \quad e \text{ is a cardiologist} \\ 0.9 & if \quad e \text{ is a specialist other than a cardilogist} \\ 0.8 & if \quad e \text{ is a general practitioner} \\ 0.7 & if \quad e \text{ is a nurse} \\ 0 & if \quad e \text{ is non-medical staff} \end{cases}$$

This is a standard approximation where all the values relevant to the use case are close between them; whereas non-medical staff are significantly separated from the rest, i.e $d(e, f) \geq 0.7$ when e is any entity who is a medical staff and f is not.

6.2 Another Possible Application: Profile Matching

Defining trust relationships in social networks is often complicated and prone to errors. Our model could automatically define trust relationships based on similarity measures. In this way, users only need to choose some relevant characteristic that define them and the corresponding similarity measures.

Computations in our model can be performed in a centralized way, by the social network provider or distributively, where each user computes their own similarity networks based on the available information.

Let us consider a social network about films where users can even rate the movies they like. We can then define a measurable characteristic, c_i, as 'Movies I like'. One of the users on this site, e is interested in a movie l. For the sake of simplicity we are going to consider two other users that can be compared to e (f, g, h). A possible graph representation for these users can be seen in Figure 3

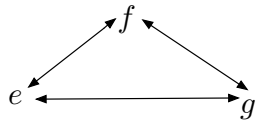

Figure 3: Graph for users in a movie rating site

The first step is to calculate the similarity between these entities. If the set of movies that a specific user likes is denoted by C_i where i is the entity we could measure the similarity between two entities as $\sharp(C_i \cap C_j)$, where $\sharp$ denotes the cardinality of the intersection set.

Let us assume that we set the threshold mentioned in Section 3 to $\delta = 5$ and that after analysing the sets of movies that the users like and the cardinality on the intersections are calculated, the results on the similarities are as follows:

$$\begin{aligned} \Delta_{e,f} c_i &= 5 \\ \Delta_{f,g} c_i &= 6 \\ \Delta_{e,g} c_i &= 4 \end{aligned}$$

A possible available path for computing trust in between e and g will be the one that follows the direct link that exists between them. However, since $\Delta_{e,g}c_i = 4$, which is below the threshold δ this path is discarded and then if we are interested in computing trust between e and h we should follow the other path, whose all the similarity values are beyond the threshold.

7 Experimental Results

In order to acquire a better understanding of how the model proposed in Section 5 works we run some experiments for the use case presented in Section 6. In these experiments we analysed how the behaviour of the model depending on the parameters δ_i, α_i and the choices for the sequential and parallel operators in a multidimensional setting.

We have chosen a scenario with the characteristics that we described in Section 6, where there are ten entities involved. The positions of these entities have been represented in Figure 4.

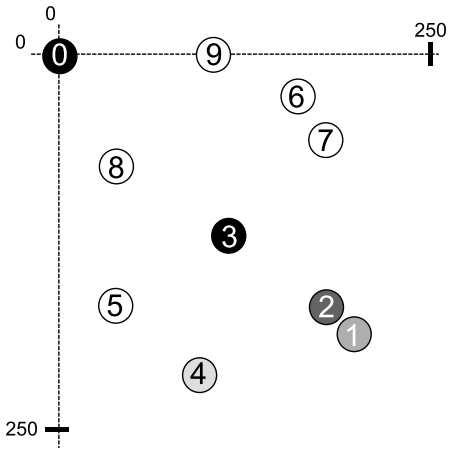

Figure 4: Distribution of the Position of the Entities

In this scenario entity 0 is the person who is having the emergency. Entity 3, represented in black, is a cardiologist. We have used a scale of darker to lighter grey to represent the most suitable staff for attending these specific emergency, where darker grey means more suitable than a lighter one. In our case these entities are 2, 1 and 4. Thus, 2 is specialist other than a cardiologist, 1 is a general practitioner (GP in the following) and 4 is a nurse. Nodes from 5 to 7 are non-medical staff.

In this particular scenario we defined the distance associated to the positions as:

$$d_P(e, f) = \frac{d_e(c_t(e), c_t(f))}{d_{max}}$$

where d_e is the euclidean distance and d_{max} is the measure of the diagonal of the box where all entities are placed, in our case case $d_{max} = 250\sqrt{2}$.

For the characteristic related to the speciality we can define the distance as

$$d_S(e, f) = |(c_t(e) - c_t(f)|$$

Those two distances are bounded by a threshold of 1. We defined the following weights for the multidimensional distance, $\alpha_P = 0.9$ and $\alpha_S = 0.1$, i.e. $d(e, f) = 0.9d_P(e, f) + 0.1d_S(e, f)$ when we want to prioritize the position with respect to the medical speciality of the entity. We also invert the weight to give more relevance to the speciality over the position.

We filter the similarity graph using the approach mentioned in 3.2. We used two parameters δ_E and δ_P to denote the upper bounds for the speciality and position distance respectively. In this case $\delta = \alpha_E\delta_E + \alpha_P\delta_P$

We increase the parameters δ_E and δ_P in steps of 0.1 from 0.1 to 1.

For each pair of values we solve the trust graph we could use different combinations of sequential and parallel functions, as we outlined in Section 5. We have chosen the combination $< Min, Max >$ as representative for this example. For all the possible paths it chooses the one with the highest value by applying the maximum. We follow the weakest link approach by assigning the weights of the paths as the minimum value for each edge.

Relevance given to the speciality with $< Min, Max >$ In Figure 5 we can see the evolution of the trust values for two entity 1, who is a GP. We can observe that the trust value for this entity reaches a high value when the threshold for the speciality is almost 1 but even for lower values of this threshold there is a value of trust assigned to this entity even if it is small. This can be explained by the fact that the speciality of entity 1 is an optimal one to attend the emergency, even though he is not a cardiologist. However, this is not the case for the position, since in this case the prevalence is given to the speciality. For the position the trust values are different than 0 only when the threshold for position starts to grow. The reason is because entity 1 is the furthest one with respect to the entity requiring assistance.

Let us now consider the case of a very close entity (9) who is non-medical staff. In Figure 6 we can observe that the trust values, as expected, are considerably low in general since the speciality of this node is not relevant at all. Only for high values of the speciality threshold (0.7) the values of trust are not 0, however very low. Since this is an entity very close to entity 0 the trust values start to be other than 0 even for a not very high value . However, since we are giving more relevance to the speciality than to the position the trust values for this entity will not reach high values.

8 Conclusion and Future Work

In this paper we have presented a trust model based on similarities. In order to find the similarities between users we built similarity networks that depend

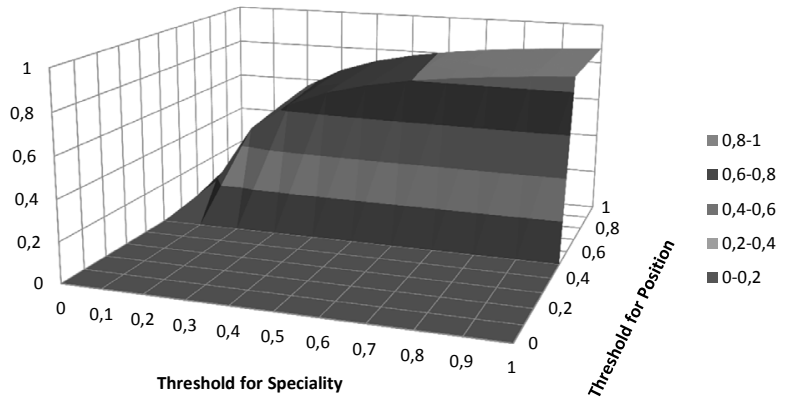

Figure 5: Trust Values for Node 1 for $< Min, Max >$ and the Speciality as Relevant Characteristic

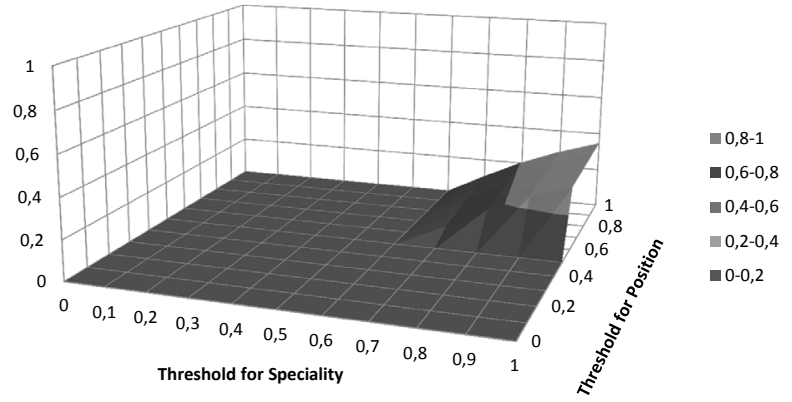

Figure 6: Trust Values for Node 9 for $< Min, Max >$ and the Speciality as Relevant Characteristic

on specific contexts where the entities are interacting. The similarity networks lead to the deployment of a similarity graph that is used for deriving the trust graph we are interested in. This trust graph determines the trust along a path of entities.

The main difference of our model with respect to our models that compare similarity and trust is that we do not assume that the similarity exists beforehand but we developed a way to calculate it and use it for deriving trust later on.

We have shown an example of a use case where our model could be applied. We have used a multidimensional setting where we combine proximity-based trust establishment and a profile matching. In this scenario we have assumed that an emergency arises and our model aids at finding the most suitable person

to deal with it depending on the position and role of the persons involved in the scenario. This scenario has been validated and the results showed that by adjusting the thresholds and weights we can customize the behaviour of our trust model.

In the future, we will apply our model and the implementation to other use cases. In particular, it will be very interesting how this model can be used in Internet of Things scenarios.

Acknowledgements

This work has been funded by and by the European Commission through the research project NESSoS under grant agreement number 256980.

References

[1] http://trust.mindswap.org/filmtrust/.

[2] Isaac Agudo, Carmen Fernandez-Gago, and Javier Lopez. A model for trust metrics analysis. In *Proceedings of the 5th international conference on Trust, Privacy and Security in Digital Business*, TrustBus '08, pages 28–37, Berlin, Heidelberg, 2008. Springer-Verlag.

[3] T. Andreasen, H. Bulskov, and R. Knappe. Similarity Graphs. In S. Tsumoto E. Suzuki N. Zhong, Z.W. Ras, editor, *14th International Symposium on Methodologies for Intelligent Systems, ISMIS*, volume 2871 of *LNAI*, pages 668–672, Maebashi, Japan, October 2003.

[4] Shyam Boriah, Varun Chandola, and Vipin Kumar. Similarity measures for categorical data: A comparative evaluation. In *Society for Industrial and Applied Mathematics, SIAM*, 2008.

[5] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[6] Diego Gambetta. Can we trust trust? In *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, 1988.

[7] Jennifer Golbeck. Generating predictive movie recommendations from trust in social networks. In *Proceedings of the 4th international conference on Trust Management*, volume 3986 of *iTrust'06*, pages 93–104, Berlin, Heidelberg, 2006. Springer-Verlag.

[8] Jennifer Golbeck. Trust and nuanced profile similarity in online social networks. *ACM Trans. Web*, 3(4):12:1–12:33, September 2009.

[9] Chern Har Yew. *Architecture Supporting Computational Trust Formation.* PhD thesis, University of Western Ontario, London, Ontario, 2011.

[10] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 538–543, New York, NY, USA, 2002. ACM.

[11] A. Jøsang. A Logic for uncertain Probabilities. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems.*, 9(3):279–311, 2001.

[12] Audun Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.

[13] Joseph A. Konstan. Introduction to recommender systems: Algorithms and evaluation. *ACM Trans. Inf. Syst.*, 22(1):1–4, 2004.

[14] R. Leiven. *Attack Resistant Trust Metrics.* PhD thesis, University of California at Berkeley, 2003.

[15] Raph Levien. *Attack Resistant Trust Metrics.* PhD thesis, University of California at Berkeley, 2004.

[16] Paolo Massa and Paolo Avesani. Trust Metrics in Recommender Systems. In John Karat, Jean Vanderdonckt, and Jennifer Golbeck, editors, *Computing with Social Trust*, HumanComputer Interaction Series, chapter 10, pages 259–285. Springer London, London, 2009.

[17] D. Harrison McKnight and Norman L. Chervany. The meanings of trust. Technical report, University of Minnesota, Management Information Systems Research Center, 1996.

[18] Keith W Miller, Jeffrey Voas, and Phil Laplante. In Trust We Trust. *Computer*, 43:85–87, 2010.

[19] Francisco Moyano, Carmen Fernandez-Gago, Isaac Agudo, and Javier Lopez. A Task Ordering Approach for Automatic Trust Establishment. In *Proceedings of the 2012 International Symposium on Engineering Secure Software and Systems (ESSoS 2012)*, volume 7159 of *LNCS*, pages 76–89, Eindhoven, The Netherlands, February 2012. Springer.

[20] Francisco Moyano, Carmen Fernandez-Gago, and Javier Lopez. Building trust and reputation in: A development framework for trust models implementation. In *pre-proceedings of the 8th International Workshop on Security and Trust Management*, Pisa (Italy), 2012.

[21] D. Olmedilla, O.F. Rana, B. Matthews, and W. Nejdl. Security and trust issues in semantic grids. In *Proceedings of the Dagsthul Seminar, Semantic Grid: The Convergence of Technologies*, volume 5271, 2005.

[22] Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, March 1997.

[23] Sini Ruohomaa and Lea Kutvonen. Trust management survey. In *Proceedings of the Third international conference on Trust Management*, iTrust'05, pages 77–92, Berlin, Heidelberg, 2005. Springer-Verlag.

[24] R. N. Sheppard. Toward a universal law of generalization for psychological science. *Science*, 237:1317–1323, 1987.

[25] Mozhgan Tavakolifard, Peter Herrmann, and Svein J. Knapskog. Inferring trust based on similarity with tillit. In Elena Ferrari, Ninghui Li, Elisa Bertino, and Yücel Karabulut, editors, *IFIPTM*, volume 300 of *IFIP Advances in Information and Communication Technology*, pages 133–148. Springer, 2009.

[26] Lu Zhang, Chunping Li, Jun Liu, and Hui Wang. Graph-based text similarity measurement by exploiting wikipedia as background knowledge. *World Academy of Science, Engineering and Technology*, 59:1548–1553, 2011.

[27] C. N. Ziegler and G. Lausen. Propagation Models for Trust and Distrust in Social Networks. *Information Systems Frontiers*, 7(4-5):337–358, December 2005.

[28] Cai Nicolas Ziegler. *Towards Decentralized Recommender Systems*. PhD thesis, Albert-Ludwigs-Universität Freiburg, Germany, 2005.

[29] Cai-Nicolas Ziegler and Jennifer Golbeck. Investigating interactions of trust and interest similarity. *Decision Support Systems*, 43(2):460 – 475, 2007.