Computational models of trust and reputation in online social networks

Sana Hamdi

NNT : 2016SACLL001

THESE DE DOCTORAT
de
LA FACULTE DES SCIENCES DE TUNIS
et de
L'UNIVERSITE PARIS-SACLAY
préparée à Télécom SudParis en Evry

ÉCOLE DOCTORALE N°580

Ecole Doctorale Sciences et Technologies de l'Information et de la Communication

Spécialité de doctorat : Informatique

Par

Mme Sana Hamdi

Computational Models of Trust and Reputation in Online Social Networks

Thèse présentée et soutenue à Evry, le 22 Janvier 2016 :

Composition du Jury :

Mme. K. Wegrzyn-Wolska, Professeure, ESIGETEL, Rapporteuse
Mme. H. Ben Ghezala, Professeure, ENSI La Manouba, Rapporteuse
Mme B. Le Grand, Professeure, Université Paris-Sorbonne1, Examinatrice
Mme Y. Bourda, Professeure, CentraleSupélec, Examinatrice
M. M.M. Gammoudi, Professeur, ISAMM La Manouba, Examinateur
Mme. A. Bouzeghoub, Professeure, Télécom SudParis, Directrice de thèse
M. S. Ben Yahia, Professeur, FST Tunis, Directeur de thèse
Mme A. Lopes Gancarski, Maître de conférence, Télécom SudParis, Encadrante

To my parents Kamel and Afifa

I am especially thankful for your love and your continuous
support. Thanks for always believing in me.
I know you wait this special day from my birth.

To my husband Nizar

I am particularly indebted for your unconditional trust,
support and sincere love.

To my brother Ahmed

To my sisters Meriem and Sonya

To my niece Alma

I love you!

To my family in law

I dedicate this work especially **to my bambino Skander**
you accompanied me, in my belly, during writing this manuscript.
I just adore you!

Acknowledgements

Throughout my Phd study period, many people have kindly provided me with their help and unlimited support. It is a pleasure to convey my most profound gratitude to them all and I am particularly indebted to the following people.

First of all, I am grateful to my supervisor Pr. Sadok BEN YAHIA, for his continuous support and his constant guidance during my PhD study years in the LIPAH laboratory. Your encouragement and personal guidance have provided a good basis for the present thesis. Your perpetual energy and enthusiasm in research had motivated me so much. I am very fortunate to have you as a supervisor. Thank you for everything, it was a truly great experience working with you!

Second, I would like to express my sincere appreciation to my supervisor Pr. Amel BOUZEGHOUB, for her warmly reception and insightful discussions and instructions during my research study. Under her supervision the quality of my work has constantly improved, not to mention the invaluable support and countless contributions to this work. Thank you for having oriented and supported me with patience and encouragements, for your great sense of responsibility and professionalism. Thank you also for being understanding and for your support in many difficult times I lived during my PhD life.

This work would have not been possible without the continuous support, advice, and encouragement from Mrs. Alda LOPES GANCARSKI. Throughout my PhD life, she has always been tireless in giving me invaluable comments and advice. Asides from work, she has also been exceptionally understanding and sympathetic with other problems I have encountered during this time. I express my greatest appreciation toward your guidance throughout the period of my study.

I wish to thank also the members of the dissertation jury for accepting the invitation and their valuable time and feedback. My special appreciation goes to Pr. Henda Ben Ghezala, Pr. Katarzyna Wegrzyn-Wolska, Pr. Yolaine Bourda, Pr. Bénédicte Le Grand and Pr. Mohamed Mohsen Gammoudi.

Many thanks to all my colleagues and friends inside Faculty of Sciences of Tunis

and outside also for the excellent and truly enjoyable ambiance.

My endless thanks to my husband and my family, for their endless care, understanding, support, and advice on my personal life.

Abstract

Online Social Networks (OSNs) have known a dramatic increase and they have been used as means for a rich variety of activities. In fact, within OSNs, users are able to discover, extend, manage, and leverage their experiences and opinions online. However, the open and decentralized nature of the OSNs makes them vulnerable to the appearance of malicious users. Therefore, prospective users face many problems related to trust. Thus, effective and efficient trust evaluation is very crucial for users' decision-making. It provides valuable information to OSNs users, enabling them to make difference between trustworthy and untrustworthy ones. This thesis aims to provide effective and efficient trust and reputation management methods to evaluate trust and reputation of OSNs users, which can be divided into the following four contributions.

The first contribution presents a complex trust-oriented users' contexts and interests extraction, where the complex social contextual information is taken into account in modelling, better reflecting the social networks in reality. In addition, we propose an enrichment of the Dbpedia ontology from conceptualizations of folksonomies.

We second propose the IRIS (Interactions, Relationship types and Interest Similarity) trust management approach allowing the generation of the trust network and the computation of direct trust. This model considers social activities of users including their social relationships, preferences and interactions. The intention here is to form a solid basis for the reputation and indirect trust models.

The third contribution of this thesis is trust inference in OSNs. In fact, it is necessary and significant to evaluate the trust between two participants whom have not direct interactions. We propose a trust inference model called TISON (Trust Inference in Social Networks) to evaluate Trust Inference within OSNs.

The fourth contribution of this thesis consists on the reputation management in OSNs. To manage reputation, we proposed two new algorithms. We introduce a new exclusive algorithm for clustering users based on reputation, called RepC, based on trust network. In addition, we propose a second algorithm, FCR, which is a fuzzy extension of RepC.

For the proposed approaches, extensive experiments have been conducted on real or random datasets. The experimental results have demonstrated that our proposed algorithms generate better results, in terms of the utility of delivered

results and efficiency, than do the pioneering approaches of the literature.

Key words: Social networks; direct trust; indirect trust; reputation.

Publications

[1] **Sana Hamdi**, Alda Lopes Gancarski, Amel Bouzeghoub et Sadok Ben Yahia. Semantic Clustering of Users based on Shared Conceptualizations in Folksonomies. In Proceedings of the 3rd IEEE International Conference on Computational Aspects of Social Networks (CASoN 2011), Salamanca, Spain, October 2011. **[indexed IEEE, DBLP]**.

[2] **Sana Hamdi**, Alda Lopes Gancarski, Amel Bouzeghoub et Sadok Ben Yahia. IRIS: A Novel Method of Direct Trust Computation for Generating Trusted Social Networks. In Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. (TrustCom 2012), Liverpool, UK, June 2012. **[indexed IEEE, DBLP], ERA rank A conference**.

[3] **Sana Hamdi**, Alda Lopes Gancarski, Amel Bouzeghoub et Sadok Ben Yahia. Enriching Ontologies from Folksonomies for eLearning: The DBpedia case. In Proceedings of the 12th IEEE International Conference on Advanced Learning Technologies. (ICALT 2012), Rome, Italy, July 2012. **[indexed IEEE, DBLP], ERA rank B conference**.

[4] **Sana Hamdi**, Alda Lopes Gancarski, Amel Bouzeghoub et Sadok Ben Yahia. Enriching the DBpedia Ontology with Shared Conceptualizations from Folksonomies. In Proceedings of the 36th IEEE Signature Conference on Computers, Software, and Applications. (CompSac 2012), Izmir, Turkey, July 2012. **[indexed IEEE, DBLP], ERA rank B conference**.

[5] **Sana Hamdi**, Amel Bouzeghoub, Alda Lopes Gancarski et Sadok Ben Yahia. Trust Inference computation for online Social Networks. In Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. (TrustCom 2013), Melbourne, Australia, July 2013. **[indexed IEEE, DBLP], ERA rank A conference**.

[6] **Sana Hamdi**, Alda Lopes Gancarski, Amel Bouzeghoub et Sadok Ben Yahia. Linking Trust in Social Networks with the Semantic Web: FOAF Case. In Proceedings of the International Machine Learning and Data Analytics Symposium. (MLDAS 2015), Doha, Qatar, Mars 2015. **The Best Paper Award.**

[7] **Sana Hamdi**, Alda Lopes Gancarski, Amel Bouzeghoub et Sadok Ben Yahia. TISON: Trust Inference in trust-oriented Social Networks. Transactions on Information Systems. In Press, Corrected Proof, 0 (2015), –. **ERA rank A journal.**

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

With the growing popularity of the Internet, open, large-scale distributed applications are becoming increasingly prevalent [Yao, 2004]. Some of the most exciting new activities on the web are social, with social networks and collaborative interactions [Sherchan *et al.*, 2013]. In fact, in recent years, we have seen dramatic increases and growing popularity of Online Social Networks (OSNs). As far as OSNs are maturing, issues that center around proper use of such networks are also growing and making headlines. In this respect, the area of trust management for OSNs is of increasing importance, especially given the exponential growth of online communities.

Trust has been thoroughly studied by researchers in psychology, philosophy and sociology; research in these fields shows that trust is a subjective view that varies greatly among people, situations and environment. However, this variance of trust has been overlooked in most of OSNs [Noorian et Ulieru, 2010]. At the moment, trust in OSNs is reduced to simple personalized access control methods which all tend to take a very simplified view of trust for all users. On the one hand, trust cannot be expressed with the variance of users' activities, it always should be personalized by users. On the other hand, it is very difficult, if not impossible, to personalize it manually, especially with a big number of users. We feel there is the compelling need for an innovative and automatic design for the model of trust in OSNs that considers the subjective view of users. This model should be able to convert the users' behaviours and interactions on freely and confidently trust opinions.

This thesis presents my research on the topic of trust management, especially for the social applications. Based on the properties of OSNs, we are addressing issues previously unresolved by the current state of the art with new trust management models for evaluating direct trust (trust a user has w.r.t. another one (s)he knows), indirect trust (trust a user has w.r.t. another user he does not know) and reputation.

This chapter describes the motivation and outlines the contributions of this work. It begins by briefly reviewing the notion of OSNs and trust in Section 1.1. Section 1.2 examines new challenges posed by the social applications. Section 1.3 outlines the contributions of this research described on a chained approach to trust management. This chaining is reflected in the structure of this thesis, which is described in Section 1.4.

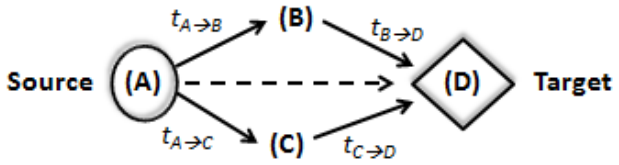


Figure 1.1: An example of trust-oriented social network

1.1 Social Networks and Trust Management

In recent years, there is a dramatic growth in number and popularity of online social networks. There are many networks available with more than 200 million registered users such as Facebook, MySpace, Twitter, Orkut, etc. People may connect, discover and share their experiences and opinions by using these OSNs. However, the user faces many problems related to trust. For example, (s)he needs to evaluate trust in a service provider before making a choice, or evaluate trust in a stranger user before accepting his friendship request. This kind of problems attracts the attention of the researchers about the importance of managing trust in OSNs, especially with their dramatic growth. In fact, establishing trust among the OSN users plays a vital role in improving the quality of services and enforcing security for the social activities. Thus, the task here is to predict algorithms evaluating the trust concerning a particular user (target user) based on a database of user rates or evaluations. Because of the different tastes of different people, they rate differently according to their subjective tastes. Moreover, these algorithms poorly perform when there is insufficient previous common rating available between users; commonly known as cold start problem [Bhuiyan *et al.*, 2010a]. To overcome these problems, a direct trust based approach is fundamental to assume a trust network among users and compute inferred trust as well as their reputations in the network.

1.2 Challenges in the Trust Management of OSNs

An OSN can be modelled as a directed graph **G** where nodes represent users, whereas edges represent relationships of a certain type between them, and edge's

label represent trust levels (e.g, $t_{A\to B}$ and $t_{A\to C}$ in Fig. 1.1). The edge direction denotes which node specified the trust value and the node for which the trust value has been specified. The trust ratings on edges are the weights of the direct trust relationships between users. In OSNs, as each user usually interacts with many others, multiple trust paths may exist between nonadjacent users from the source user (e.g., A) to the target one (e.g., D) (e.g., paths $A \to B \to D$ and $A \to C \to D$ in Fig. 1.1). If there exists at least one trusted path linking two unknown users (e.g., A and D are linked by two trusted paths), there may exist a trusted connection between them. All such trusted links form a trust network from a source to a target (e.g., the trust network from A to D in Fig. 1.1). Direct and indirect trust degrees between users can be used to build users' reputations.

This thesis will focus on the four significant challenging problems: *(i)* direct trust computing and trust network extraction; *(ii)* trust path selection and trust transitivity; *(iii)* reputation management in OSNs; and *(iv)* the enrichment of the Semantic Web with trust and reputation information.

1.2.1 Trust Network Extraction

Most interactions between two users in OSNs can be broken down into the scenario shown by Fig. 1.2: Alice (`A`) is a service truster who does not directly know John (`J`), the service provider. J is the target whose trust value has to be evaluated, and `A`'s question is "Can I trust `J`? and who should I ask for?". Thus, to satisfy the request of the truster `A` about the trustee `J` and to provide for her a proper answer, the network should be converted in a trusted one by computing direct trust relations between related users.

Therefore, in OSNs, such a trust network is fundamental and critical for trust evaluation between two non-adjacent users or for reputation evaluation, as it contains some important intermediate users, the trust relations between those participants and the social context. All of them have worth of cite influences on the trust computing between two unknown users in OSNs and on the reputation evaluation.

In the literature, there have been several existing trust evaluation approaches for trust evaluation between two non-adjacent users or reputation evaluation.

Figure 1.2: A scenario of trust request

However, they all assume that the trust network between two adjacent users has been identified. Therefore, given any two users whom have direct interactions in a large-scale social network, extracting the trust degree between them becomes a fundamental and essential step before performing any trust propagation methods. Such a task is called *trust network extraction*.

1.2.2 Trust Transitivity

In Fig. 1.2, Alice asked if she can trust Jhon or not and requested "who should I ask for?". Thus, an effective inferred trust evaluation algorithm is expected and an optimal trusted path selection method should take place.

In an extracted large-scale trust-oriented social network, there could be tens of thousands of social trust paths between a source participant and the target one ([Liu, 2013] and [Kunegis *et al.*, 2009]). Evaluating the trustworthiness of the target participant based on all these social trust paths can incur huge computational time. Alternatively, we can search an optimal path yielding the most trustworthy trust propagation result from multiple paths. This is called *the optimal social trust path selection problem* that is known to be a challenging research problem in OSNs ([Liu, 2013] and [Hamdi *et al.*, 2013]).

After extracting the trust network and selecting the trustworthy social trust paths from the trust network, the computation of the value of trust for the target user requires an understanding of how trust is propagated along a social trust

path, which is a critical and challenging problem in OSNs [Hamdi *et al.*, 2013]. In the literature, several trust transitivity models have been proposed ([Adamatti *et al.*, 2013], [Golbeck, 2005], [Jiang et Wang, 2011], [Taherian *et al.*, 2008]), but the following drawbacks are worth of mention:

1- These existing trust transitivity models do not fully consider many important social contextual information, e.g., social relationships, social interactions and preferences or interests that have significant influence on trust transitivity ([Hamdi *et al.*, 2012a], [Lichtenstein et Slovic,], [Liu, 2013]).

2- Although different trust evaluation criteria (such as the length of considered trust paths and direct trust values) can influence trust transitivity results, the specification of such criteria is not supported by most of existing methods.

3- Trust transitivity, as formalised in existing models, does not follow the nature of trust decay illustrated in social psychology, namely, trust decays slowly in a certain number of early hops from a source participant, and then decays fast until the trust value approaches the minimum [Gimpel *et al.*, 2008], [Jøsang *et al.*, 2003].

1.2.3 Reputation Computing

A truster (e.g. Alice in Fig. 1.2) needs to have a global perception of the trustworthiness of the target user in the network (e.g. Jhon in Fig. 1.2). The trustworthiness can be evaluated from its past and current behaviours and thus it is based on direct and indirect trust. In fact, after generating the trust network and computing direct and indirect trust, a reputation evaluation method helps users to make difference between trustworthy and untrustworthy users.

Since the evaluation of users varies from a user to another one, having several users could lead to different opinions (benevolent users w. r. t. a user can be malicious w. r. t. another one). This difference of evaluations hampers the process of users' classification. An effective process of reputation evaluation algorithm is expected to tackle this problem and address users' classification as well as the selection of the most benevolent ones. In the literature, several reputation models have been proposed, but in most of these models, the reputation of a user is determined by a simple trust average ratings provided by other users for that user.

1.2.4 Linking Trust in OSNs with the Semantic Web

One of the core goals of the Semantic Web is to store data in distributed locations and use ontologies and reasoning to aggregate it. In addition, it offers a promising solution to publish information and services on the World Wide Web augmented with descriptions in a processable form understandable by both agents and machines. This will help Web agents to perform a variety of tasks on behalf of their users. However, providing, in the Semantic Web, relevant information related to trust, while offering the access to it to the OSNs' users, is rarely discussed in the literature. Therefore, an approach that exploits the Semantic Web and satisfies the needs of OSNs' users by storing important trust data in a well organized and easy to understand structure using is expected.

The research challenges presented in this section effectively set out the goals for this work. This thesis is intended to address most of the above mentioned issues in an attempt to have a trust management system adequate to properties of OSNs.

1.3 Thesis contributions

As discussed in previous sections, trust problems for distributed social applications present complex research challenges. These problems cannot be tackled in a single step due to their complexity as well as inhered relationships. A good, well-known managing practice is to divide a large problem into smaller pieces, solve each piece separately to produce a consistent solution. A novel chained approach for trust managing is proposed in this thesis. It separates trust issues into four steps, sketched by Fig. 1.3. These steps are designed to address many of the pending research challenges described in the previous section. Indeed, extracting the social impact factors including social relationships, preferences, and social interactions presents the first step for the trust management in OSNs. The second step consists of the generation of the trust network and computing direct trust values as it is of paramount importance for performing any trust path selection and inferred trust evaluation methods. Based on the solution of trust network extraction and effectively and efficiently evaluate the direct trust between two known users, the third step includes the selection of trusted paths and perform trust transitivity computation to deliver reasonable trust values. The fourth step

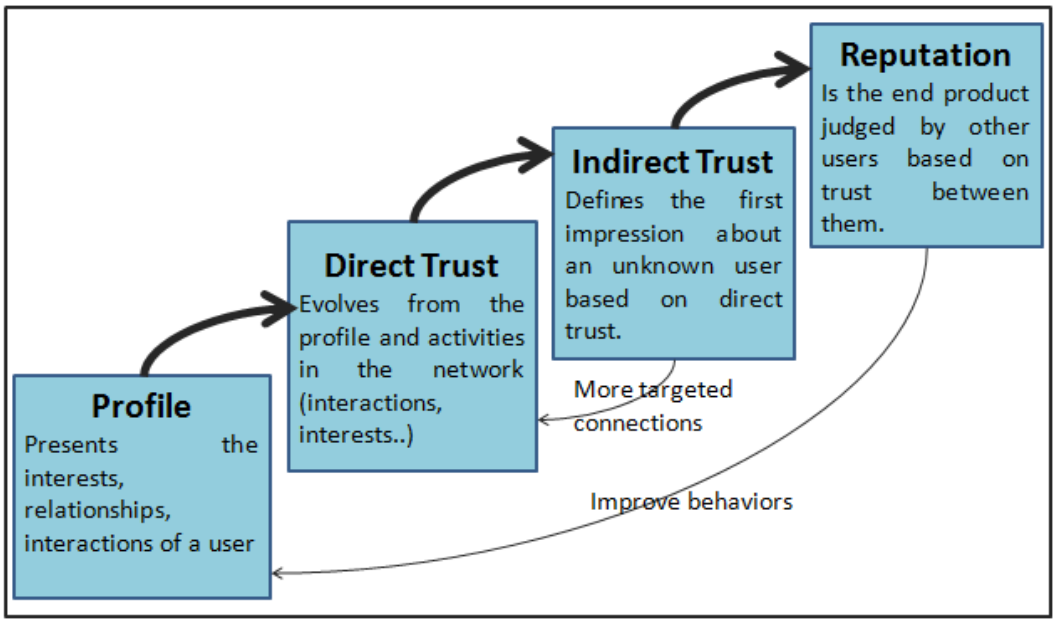


Figure 1.3: Chained steps for our trust managing models

covers the reputation management in OSNs.

The list of contributions is described below.

- The first contribution introduces a complex trust-oriented users' contexts and interests extraction, where the complex social contextual information are taken into account in modelling, better reflecting the social networks in reality. In addition, we propose an enrichment of the Dbpedia ontology from conceptualizations of folksonomies.

- The second contribution of this thesis is the generation of the trust network and the computation of direct trust.

  a. To compute the direct trust values between users having direct knowledge, we propose a new trust management approach called **IRIS** (Interactions, Relationships and Interests' Similarity). This approach considers social activities of users including their social relationships, preferences as well as interactions.

  b. We propose a novel complex trusted contextual social network structure. This new structure contains the social contextual information cited above enriched by the computed direct trust values. It can reflect

the social networks in the real world better because the above mentioned important social contextual information in the human society is modelled in the new structure.

c. Experiments conducted on real social network datasets illustrate that on average, our methods can extract trust networks with higher quality.

- The third contribution of this thesis is trust inference in OSNs. We propose a trust inference model called **TISON** (Trust Inference in Social Networks) to evaluate Trust Inference within online Social Networks.

  a. To address the scalability issue of trust path selection problem, we propose an efficient approximation algorithm, TPS for Trust Paths' Searching. We develop the TPS algorithm where a source user may refer to multiple social trust paths to obtain a more reasonable trust evaluation result of a target user. We define neighbours priority based on their direct trust degrees and then select trusted paths while controlling the path length.

  b. We develop different TIM algorithms for Trust Inference Measuring. This measure is based on: *(i)* trust propagation based on the trusted paths discovered by the TPS process; and *(ii)* trust aggregation to decide how much the source user will trust the target one.

  c. Experiments carried out on real life social network datasets illustrate that our algorithms can compute indirect trust with high quality and consumes less execution time than do the existing methods.

- The fourth contribution of this thesis consists in the reputation management in OSNs.

  a. We introduce a new clustering reputation algorithm, **RepC**, based on a trust network. This algorithm classifies an OSN's users into clusters by their trust similarity such that most trustworthy users are gathered in the same cluster. Therefore, in order to obtain a more reasonable reputation evaluation result, we propose a second algorithm, called **FCR**, which is a fuzzy extension of **RepC**.

  b. We introduce a novel approach that exploits the Semantic Web and especially the FOAF ontology [1]. In fact, using the semantics of the

[1] http://www.foaf-project.org/

FOAF ontology and applying Semantic Web reasoning techniques, we show that trust and reputation can be merged between different users from multiple social networks.

c. Experiments conducted on a real life online social network datasets demonstrate the performance of our proposed algorithms.

1.4 Dissertation outline

This thesis is organized as follows:

Chapter 2 presents a background information on Social Networks as well as the notion of trust. Current definitions are presented. A brief history of the social networks is provided. Moreover, a review of the properties of trust relationships are given and a trust classification scheme is introduced.

Chapter 3 reviews major works in the area of trust and reputation management, with the focus on social networks. It begins with an overview of the general trust problem, followed by descriptions of various trust schemes in two categories: trust (direct and indirect) and reputation. Then, for each category, a survey of the pioneering approaches, recent solutions and their problems are provided. The chapter ends with a comprehensive discussion of the state-of the art in trust and reputation management systems.

Chapter 4 presents a trust-oriented users' contexts extraction, where the complex social contextual information are taken into account in modelling, better reflecting the social networks in reality.

Chapter 5 introduces the ***IRIS*** trust management approach and highlights how ***IRIS*** considers social activities of users. The intention here is to form a solid basis for the indirect trust model.

Chapter 6 describes the ***TISoN*** model to generate and evaluate Trust Inference within online Social Networks. In this chapter, we investigate the properties of trust propagation on networks, based on the notion of transitivity discussed in

chapter 2.

Chapter 7 describes two clustering algorithms ***RepC*** and ***FCR***. These algorithms are designed to classify users through the computation of the membership degrees for each user in different clusters. The reputation management is based on direct and indirect trust defined in Chapters 5 and 6.

Chapter 8 concludes this thesis, with a summary of the main contributions and a brief discussion of potential future research and extensions.

Chapter 2

Social Networks and Notions of Trust

2.1 Introduction

The vast public interest in social networks has opened up many new spaces of possible research in computing. This research adopts OSNs as the foundation for studying trust.

This chapter introduces the concepts of online social networks in Section 2.2. We begin by defining what constitutes an OSN in Sub-section 2.2.1. Second, we present a brief history of the development of OSNs in Sub-section 2.2.2. The sizes and categorization of OSNs are discussed in Sub-sections 2.2.3 and 2.2.4. We then introduce, in Sub-section 2.2.5, the Friend-Of-A-Friend (FOAF) Project, a Semantic-Web based technology that allows users to combine information about themselves from a variety of OSNs.

Section 2.3, the second part of this chapter, reviews the basic notions of trust. Sub-section 2.3.1 defines the trust as used in several disciplines. Properties of trust are discussed in Sub-section 2.3.2. Sub-section 2.3.3 addresses some of the possible options for the values representing trust. Finally, Section 2.4 provides a summary of the chapter.

2.2 Social networks

In recent years, we have seen a dramatic increase of OSNs such as Twitter [1], Facebook [2] and MySpace [3] just to name a few, where one can set up a profile and invite friends to join the site with the purpose of sharing information and resources.

2.2.1 Definitions

There are many ways in which social networks can be automatically derived on the web. In this work, an OSN must fulfill the following criteria:

- allows users to construct a public or semipublic profile within a bounded system;

[1] https://twitter.com
[2] https://www.facebook.com/
[3] https://myspace.com/

- allows users to articulate a list of other users with whom they share a connection;
- allows users to view and traverse their list of connections and those made by others within the system.

What makes social network sites unique is that they allow users to meet "strangers" and enable them to articulate and make visible their profiles and shared resources. In fact, the term "networking" emphasizes relationship initiation, often between strangers.

While OSNs have implemented a variety of technical features, their backbone consists of visible profiles that display an articulated list of Friends who are also users of the network. Profiles are unique pages presenting users' summary of their personal details or current situation. On the one hand, after joining an OSN, an individual is asked to fill out forms containing series of questions. The profile is generated using the answers to these questions, which typically include descriptors such as age, location, interests, and an "about me" section. Most sites also encourage users to upload a profile photo. Some sites allow users to enhance their profiles by adding multimedia content or modifying their profile's look and feel. Others, such as Facebook, allow users to add applications that enhance their profile.

On the other hand, users are prompted to identify others in the system with whom they have a relationship. The label for these relationships differs depending on the site, popular terms including "Friends", "Contacts", and "Fans". Most OSNs require bidirectional confirmation for Friendship, but some do not. These onedirectional ties are sometimes labeled as "Fans" or "Followers," but many sites call these Friends as well. The term "Friends" can be misleading, because the connection does not necessarily mean friendship in the everyday vernacular sense [Boyd et Ellison, 2007].

Most OSNs also offer a mechanism for users to leave messages on their friends' profiles or to make comments to their publications. Moreover, OSNs often have a private messaging feature similar to web-mails. While both private messages and comments are popular on most of the major OSNs, they are not universally available.

Beyond profiles, Friends, comments, and private messaging, OSNs vary greatly in their features and user base. Some of them have photosharing or videosharing

capabilities; others have built in blogging and instant messaging technology. There are mobilespecific OSNs (e.g., Dodgeball), but some webbased Social Networks also support limited mobile interactions (e.g., MySpace, and Cyworld).

The visibility of a profile varies by site and according to user discretion. By default, profiles on Friendster and Tribe.net are crawled by search engines, making them visible to anyone, regardless of whether or not the viewer has an account. Alternatively, LinkedIn controls what a viewer may see based on whether she or he has a paid account. Sites like MySpace allow users to choose whether they allow their profile to be public or only visible by friends. Facebook takes a different approach: by default, users whom are part of the same network can view each others' profiles, unless a profile owner has decided to deny permission to those in their network. Structural variations around visibility and access are one of the primary ways that OSNs differentiate themselves from each other.

The public display of connections is a crucial component of OSNs. The Friend list contains links to each Friend's profile, enabling viewers to traverse the network graph by clicking through the Friends lists. On most sites, the list of Friends is visible to anyone who is permitted to view the profile, although there are exceptions. Thus, each user can receive many friend requests from strangers that have in some cases bad impacts.

When a user accepts a friend request from strangers, (s)he puts himself/herself at severe risks. The person at other side starts exchanging words very cordially and within some days he has a faith on them. Then they start the mind game and start gathering more personal information. For instance, if they get to know the email address, they could go to email account, click on "forget password" and the system will ask a security question like: "the birthday of your mother", "your first address", etc. Unknowingly, the user has already shared this information with them. So, now they have access to his email account. This task will be much easier if (s)he sets his/her date of birth, school or college name as email password. A study, conducted by the Internet security firm Webroot, found that 2 from 10 people use significant date, such as birth date, or a pet's name as password which is often publicly visible on social networks. In addition, 4 out 10 respondents shared password with at least one person [4]. Similarly, another

[4]Boulder, Colo. (2010, October 12). New webroot survey reveals poor password practices that may put consumers' identities at risk. Retrieved from http://www.webroot.com/us/en/company/press-room/releases/protect-your-computer-from-

research done by the Internet security company BitDefender [5] reveals that 75 percent of individuals use same password for social networking sites and Email in 2010. All this ignorance by users becomes the main reason behind hacking of OSNs profile and email account.

2.2.2 A Brief History

The concept of social networking dates back to 1930, when Vannevar Bush first introduced his idea about "memex" [Bush, 1996], a "device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility", and predicted that "wholly new forms of encyclopedias will appear, ready made with a mesh of associative trails running through them, ready to be dropped into the memex and there amplified".

According to the definition above, the first recognizable social network site launched in 1997. SixDegrees.com allowed users to create profiles, list their Friends and, beginning in 1998, surf the Friends lists. SixDegrees promoted itself as a tool to help people to connect with and send messages to others. While SixDegrees attracted millions of users, it failed to become a sustainable business and, in 2000, the service closed. Looking back, its founder believes that SixDegrees was simply ahead of its time [Boyd et Ellison, 2007]. While people were already flocking to the Internet, most did not have extended networks of friends whom were online. Early adopters complained that there was little to do after accepting Friend requests, and most users were not interested in meeting strangers.

From 1997 to 2001, a number of community tools began supporting various combinations of profiles and publicly articulated Friends. AsianAvenue, BlackPlanet, and MiGente allowed users to create personal, professional, and dating profiles and users could identify Friends on their personal profiles without seeking approval for those connections.

The next wave of OSNs began when Ryze.com was launched in 2001 to help people leverage their business networks. Ryze's founder reports that he first introduced the site to his friends, primarily members of the San Francisco business and

hackers

[5] http://www.bitdefender.com

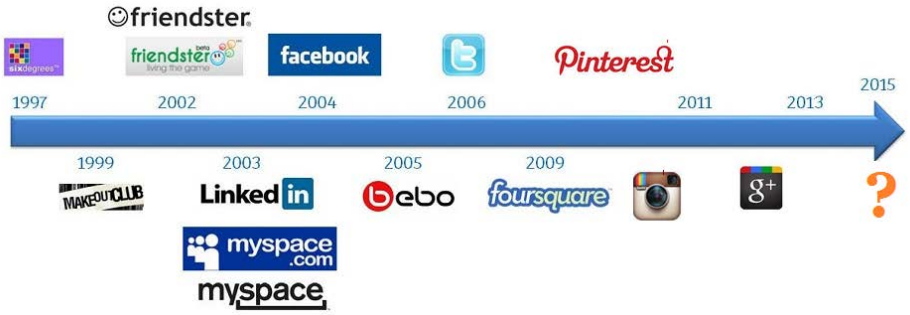


Figure 2.1: A trust-oriented social network

technology community, including the entrepreneurs and investors behind many future OSNs. In particular, the people behind Ryze, Tribe.net, LinkedIn, and Friendster were tightly entwined personally and professionally. They believed that they could support each other without competing. In the end, Ryze never acquired mass popularity, Tribe.net grew to attract a passionate niche user base, LinkedIn became a powerful business service, and Friendster became the most significant site. Friendster was one of the first of these sites to attain over 1 million members and was considered the top online social network service until around April 2004, when it was overtaken by MySpace in terms of page views according to Nielsen online institute.

From 2003 onward, many new ONSs were launched. Friendster had received competition from these sites such as hi5, LinkedIn, Facebook, Bebo and Twitter. On February, 4, 2004, Facebook was founded by Mark Zuckerberg with his college roommates. The founders had initially limited the website's membership to Harvard university students. From the end of 2006, Facebook allows anyone who claims to be at least 13 years old to become a registered user of the website. In the meanwhile, many OSNs appeared. Although they presented remarkable competitors, as Twitter and Google+, Facebook stays in the lead in terms of number of visitors.

Figure 2.1 presents a brief Timeline of the launch dates of many major OSNs.

2.2.3 Size of OSNs

The size of the social networks varied greatly. Most of OSNs have over one million members. Figure 2.2 shows the five largest OSNs.

From its foundation, Facebook knew a huge evolution and reached 1 million active users in the end of 2004. The number of users had grown broadly and exponentially to reach more than 1 billion in 2013.

Twitter rapidly gained worldwide popularity, with 500 million registered users in 2012 and 600 million in 2013, whom posted 340 million tweets per day.

Google+ is described by Google as the "social layer" that enhances many of its online properties, and that it is not simply a social networking website, but also an authorship tool that associates web-content directly with its owner/author. It is the second-largest social networking site in the world after Facebook. It reached the one billion of users by the end of 2013.

Although Instagram and Pinterest are new OSNs, both of them were launched in 2010, they rapidly gained popularity. In February 2013, Reuters and ComScore stated that Pinterest had 48.7 million users globally [McBride, 2013]. A study released in July 2013 by the French social media agency Semiocast revealed the website had 70 million users worldwide [Horwitz, 2013]. On February 27, 2013, Instagram announced 100 million active users, only two-and-a-half years after the launch of the network [Mansell, 2013]. As of September 9, 2013, the company has announced a total of more than 150 million monthly active users [Rusli, 2013].

2.2.4 Categorization

In the literature, the social networks were classified according to the purpose of their use. The categories are shown in Table 2.1.

- **Social Networks of mass:** Services that allow users to connect with other people of similar interests and background. Usually they consist of a profile, various ways to interact with other users, ability to setup groups, etc. The most popular are Facebook and LinkedIn.

- **Bookmarking Sites:** Services that allow users to save, organize and manage links to various websites and resources around the Internet. Most of them allow to "tag" links to make them easy to search and share. The most popular are Delicious and StumbleUpon.

- **Social News:** Services that allow people to post various news items or links to outside articles and then allow its users to "vote" on the items. The voting is the core social aspect as the items that get the greatest number

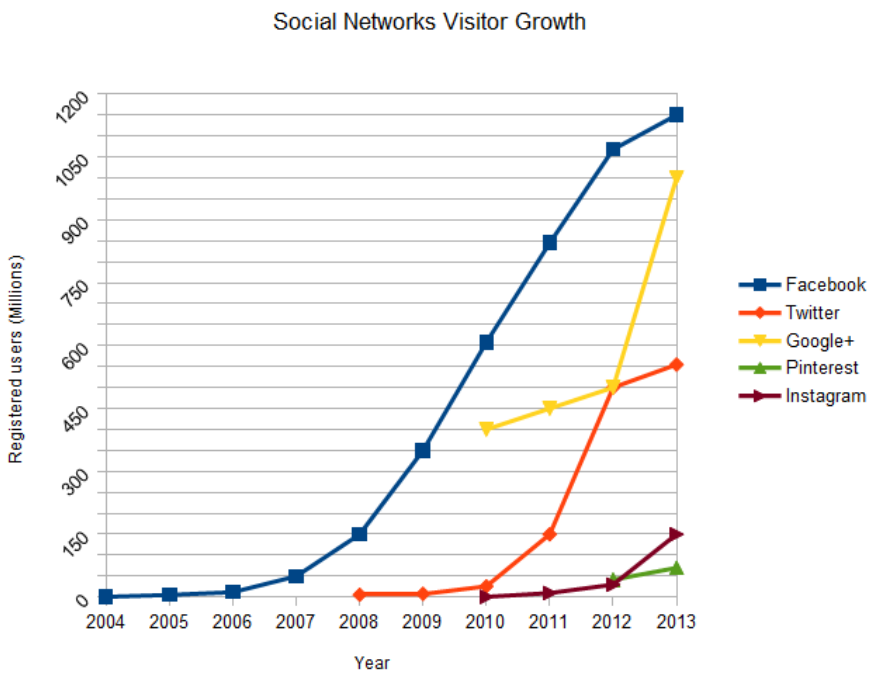


Figure 2.2: The five largest OSNs

of votes are displayed the most prominently. The community decides which news items get seen by more people. The worth of cite Social News sites are Digg and Reddit.

- **Media Sharing:** Services that allow users to upload and share various media such as pictures and video. Most services have additional social features such as profiles, commenting, etc. The most popular are YouTube and Flickr.

- **Microblogging:** Services that focus on short updates that are pushed out to anyone subscribed to receive the updates. The most popular is Twitter.

- **Blog Comments and Forums:** Online forums allow members to hold conversations by posting messages. Blog comments are similar except they are attached to blogs and usually the discussion centers around the topic of the blog post. There are many popular blogs and forums.

While these are the 6 different categories of social networks, some networks fall into multiple categories. For instance, Facebook has microblogging features with

their "status update". Also, Flickr and YouTube have comment systems similar to that of blogs.

2.2.5 Friend of a Friend (FOAF)

The billions of members in social networks do not represent unique people. In fact, many people maintain accounts at multiple social networking websites. It is desirable, for instance, to keep information intended for business networking separate from information about meeting. Members' bosses or colleagues certainly do not need to know they enjoy long walks on the beach. At the same time, users put significant effort into maintaining information on social networks [Golbeck, 2005].

Multiple social network accounts are not just for sharing parts of their lives. A person may have one group of friends who prefer Twitter, another group on LinkindIn, and have an account on Facebook to stay connected to that community. Information about the user that is distributed across several sites also would be merged. In this thesis, we need to merge these distributed information to compute direct trust. The Friend-of-a-Friend (FOAF) Project [Dumbill, 2002] is a potential solution to sharing social networking data among sites, and this section introduces how that is being done.

Background

The FOAF (Friend-of-a-Friend) vocabulary [Brickley et Miller, 2010] describes user's information and their social connections through concepts and properties using the semantic Web technologies. The FOAF vocabulary reveals basic information of users such as name, surname as well as personal information about the people that a user "knows" and his interest area. It also depicts user information regarding his social relationships by OnlineAccounts such as *YahooChatID*, *msnChatID* and user's membership information in different groups and organizations. Users resources such as images, thumbnails or logo are included in Documents and Images of the user's information.

Example 1: The following code example contains a simple FOAF description of a person:

```
<foaf:Person rdf:ID="GP">
```

Category	Examples	Characteristics
Social Networks of mass	Facebook Google+ Myspace Linkedin Orkut	Network of people Sharing of contents Entertainment and exploration
Social News	Digg Reddit scoopeo fuzz wikio TapeMoi	Hard-hitting and entertaining news Offering systems of friends' networks
Bookmarking Sites	Delicious Diigo Stumbleupon Pearltrees	Giving the meaning to bookmarks by tags and comments
Media Sharing	Youtube Dailymotion Flickr 500px Slideshare Wikipedia	Current events on real time So important channels as the TV
Micro-blogging	Twitter Tumblr FriendFeed Cif2.net Plurk Jaiku	Important source of real-time news and updates for recent crisis situations
Blog Comments and Forums	Gizmodo Forum "Com- mentça marche.net"	allow visitors to comment on the content and to directly interact with each other.

Table 2.1: Categories of OSNs

```
<foaf:givenname>Gautier</foaf:givenname>
<foaf:familyname>Poupeau</foaf:familyname>
<foaf:weblog rdf:resource="http://www.lespetitescases.net/"/>
<foaf:img rdf:resource="http://www.lespetitescases.net/got.jpg"/>
<foaf:gender>male</foaf:gender>
<foaf:knows>Christian Faure, David Larlet, Emmanuelle Bermès</foaf:knows>
<foaf:interests>Semantic Web, comics, Science fiction</foaf:interests>
```

From this snippet, a program that understands OWL and RDF is able to process the information. Using the FOAF vocabulary, the program can recognize that there is a person named "Gautier Poupeau", with a weblog and a picture online, who knows "Christian Faure", "David Larlet", and "Emmanuelle Bermès" and is interested in the Semantic Web, comics and the science fiction.

Merging and Querying Current OSNs

As introduced previously, FOAF allows us to describe personal profiles, but it can also be used to represent relationships between people. In fact, it allows people to interlink and unify the various profiles that represent themselves by automatically generating a FOAF profile which links to other existing profiles. In addition, it also interlinks distributed social networks from various platforms [Bojars *et al.*, 2008] (cf. Figure 2.3). The following snippet, sketched by Example 2, mentions three different OSNs' accounts for the same person.

Example 2:

```
:me owl:sameAs flickr:33669349@N00;
owl:sameAs twitter:terraces;
owl:sameAs facebook:foaf-607513040.rdf#me.
```

Providing such an entry point allows any RDF-compliant tool to browse one's complete social network in a simple way, i.e. retrieving relationships from Flickr, Twitter or Facebook *(i)* with standard libraries and SPARQL queries; and *(ii)* without having to crawl the Web for data since everything can be accessed from one FOAF file.

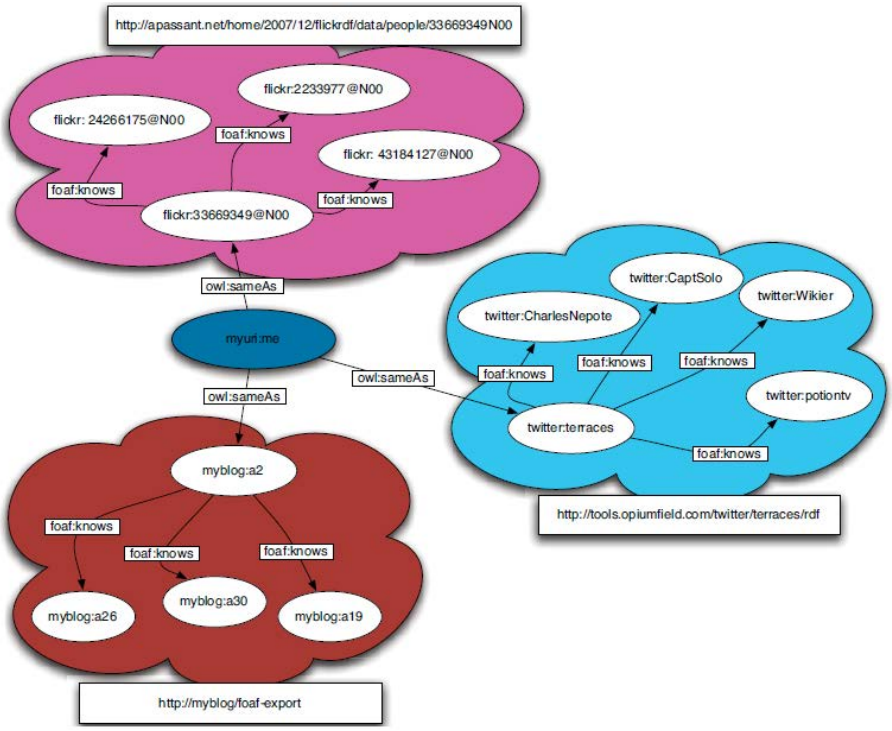


Figure 2.3: Interlinking social networks with the Semantic Web [Bojars *et al.*, 2008]

2.3 Trust

Trust is a very general topic that may be applied to virtually any context. In fact, it plays a role across many disciplines, including sociology, psychology, economics, political science, history, philosophy, and computer science [Sherchan *et al.*, 2013]. Thus, definitions vary depending on the researcher's background, outlook on life and potentially in each context where it is applied [Golbeck, 2005], [Grandison, 2003]. In the following, we first discuss trust definitions in the primary disciplines concerned with trust relationships: psychology, sociology, and computer science. Then, we describe the properties of trust and we address some of possible values representing trust.

2.3.1 Definitions of trust

It is so challenging to define trust. The literature on trust is also quite confusing, since it manifests itself in fairly different domains and forms [Chen *et al.*, 2011]. In the following, we lead a more thorough study of the trust definitions mainly stemming from psychological, social and computer sciences.

Trust in Psychology

In psychology, trust is considered to be a psychological state of the individual, where the trustor risks being vulnerable to the trustee based on positive expectations of the trustee's intentions or behaviour [Rousseau *et al.*, 1998]. Trust is considered to have three aspects: cognitive, emotive, and behavioral [Beatty *et al.*, 2011].

Trust in Sociology

In sociology, trust is defined as "a bet about the future contingent actions of the trustee" ([Dumouchel, 2005], [Sztompka, 1999]). This bet, or expectation, is considered to be trust only if it has some consequence upon the action of the person who makes the bet (i.e., trustor). Trust is considered from two viewpoints: individual and societal. At individual level, similar to the perspective from psychology, the vulnerability of the trustor is a major factor ([Rousseau *et al.*, 1998], [Molm *et al.*, 2009], [Cook *et al.*, 2005]). Trust is differentiated from cooperation in the presence of assurance (a third party overseeing the interaction and providing sanctions in case of misbehavior). However, cooperation in the pres-

ence of the shadow of the future (i.e., fear of future actions by the other party) is considered to be trusted ([Molm *et al.*, 2009], [Cook *et al.*, 2005]). In this respect, social trust has only two facets, cognitive and behavioural; the emotive aspect is built over time as far as trust increases between two individuals ([Kollock, 1994], [Lawler et Yoon, 1996]). At societal level, trust is considered to be a property of social groups and is represented by a collective psychological state of the group. Social trust implies that members of a social group act according to the expectation that other members of the group are also trustworthy [Lewis et Weigert, 1985] and expect trust from other group members. Thus, at societal level, social trust also has the institutional or system aspect of trust.

Trust in Computer Science

Trust in computer science in general can be classified into two broad categories: "user" and "system". The notion of "user" trust is derived from psychology and sociology [Marsh, 1994], with a standard definition as "a subjective expectation an entity has about another's future behaviour" [MUI, 2003]. This implies that trust is inherently personalized. In online systems such as eBay [6] and Amazon [7], trust is based on the feedback on past interactions between members ([Resnick *et al.*, 2000], [Ruohomaa *et al.*, 2007]). In this sense, trust is relational. As two members interact with each other frequently, their relationship strengthens, and trust evolves based on their experience. Trust increases between members if the experience is positive and decreases otherwise. In online systems, trust is considered to be of two types: direct trust and inferred trust. Direct trust is based on the direct experience of the member with the other party. Inferred trust is based on experiences of other members in the social network with the other party. Inferred trust is based on the propagative property of trust [Jøsang *et al.*, 2006].

2.3.2 Properties of trust

In the web-based social environment, many properties of trust are proposed ([Golbeck, 2006a], [Bhuiyan *et al.*, 2010b], [Sherchan *et al.*, 2013]). These properties identify where trust exists in social networks, and how it can be used in

[6] http://www.ebay.com/

[7] https://www.amazon.com/gp/gw/ajax/s.html

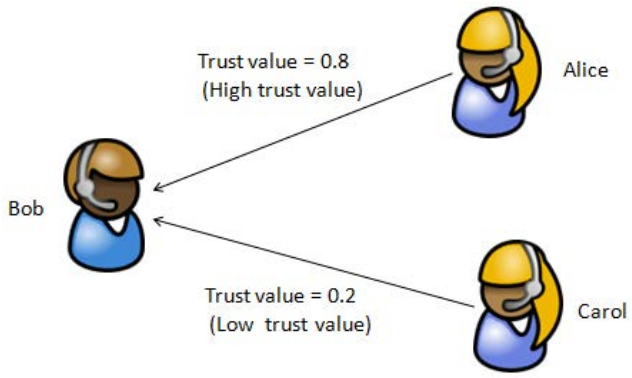


Figure 2.4: Example of 2 trusters expressing different trust in the same trustee

computation. In the following, we present the characteristics that were identified and subsequently included in this work, namely (*i*) Subjectivity; (*ii*) Propagation; and (*iii*) Asymmetry.

Subjectivity

As illustrated in social psychology ([Hardin, 2002], [Mansell et Collins, 2005]), trust is a subjective phenomenon that is defined by the psychological experiences of the individual who bestows it, reflecting subjective attitudes that affect participants' thinking based on subjective evaluation criteria which can vary in different domains. Thus, we can find two people often having very different opinions about the trustworthiness of the same person. For an example, we need only to look to politics. When we ask in Tunisia, "do you trust the current President to effectively lead the country?", the population will be split, some will trust him highly, and the others will have very little trust in his abilities.

The same for OSNs users. Each user is free to trust another one. Figure 2.4 sketches an example of 2 trusters expressing different trust in the same trustee. Alice is free to trust Bob at every level (in the example, high trust) just as Carol is free to trust Bob at every level (in the example, low trust). We respect this property when computing direct trust in Chapter 5.

Subjective trust is sometimes used in opposition to objective trust, even if there is no "objective trust" by definition and so objective trust would be better called *reputation or global reputation*[8]. In reality, the trust cannot be defined

[8]http://www.trustlet.org/wiki/Objective_trust.

objectively simply because every user is free to express a level of trust different from the level of trust expressed by the other users on a same trustee. For this reason, it is better to use the term "reputation" when referring to an aggregated value computed by a global trust metric trying to represent what the OSN as a whole thinks about a certain user. Chapter 7 presents our approach of evaluating users reputations.

Propagation

Trust is propagative, in that, as shown in Figure 2.5, if Alice trusts Bob, who in turn trusts Carol, whom Alice does not know, Alice can derive some amount of trust on Carol based on how much she trusts Bob. This is not to say, however, that trust is transitive.

Unfortunately, the propagative nature of trust is sometimes confused with the transitive nature of trust in the literature as in [DuBois *et al.*, 2011] and [Jøsang *et al.*, 2003]. However, many works assume that trust can be transitive but this transitivity needs certain constraints ([Christianson et Harbison, 1996], [Jøsang et Pope, 2005]). Thes more thorough works distinguish between different contexts of trust, and do not allow for transitivity between contexts that are semantically incompatible or inappropriate. Namely, if Alice trusts Bob in the domain of teaching java, and Bob trusts Carol in the domain of repairing a car, then the trust cannot be transitive from Alice to Carol via Bob in the domain of teaching java. However, if Alice also trusts Bob in repairing a car (in the same domain that Bob trusts Carol), then trust can be transitive from Alice to Carol in this domain and Alice can understandably cooperate with Carol in their first interaction.

The propagation is especially important as in an open network we have to interact with new or unknown agents, and it is not possible to count on our personal experience and evaluation, or on some authority's guarantees, or on the explicit recommendation of another agent [Adamatti *et al.*, 2013]. In fact, without propagation, trust metrics are unlikely to be used to reason about trust in more complex relationships. Thus, the propagation is the most studied property of trust and various trust models have used this property.

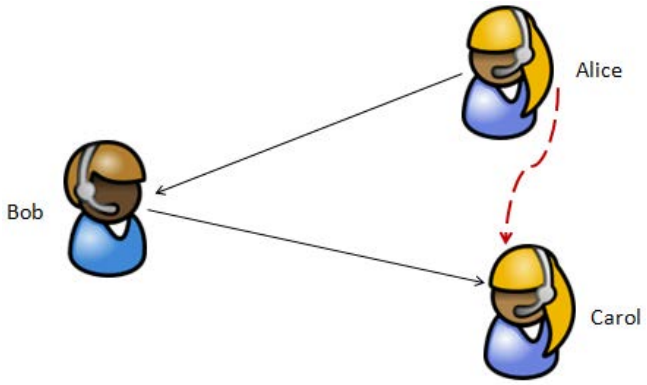


Figure 2.5: An illustration of how Bob is used in the trust inference from Alice to Carol

Asymmetry

In social networks, it is also important to note the asymmetry property of trust. For two users involved in a relationship, trust is not necessarily the same in both directions. Because individuals have different experiences, psychological backgrounds, and histories, it is understandable why two users might trust each other to different degrees [Golbeck et Hendler, 2006]. In addition, if one of the users does not act in a trustworthy manner, the other user will be forced to penalize him, leading to low mutual trust.

2.3.3 Values of trust

The trust presents information about a social relationship and, as such, in an OSN, it must be represented as a label on that relationship. There is still much freedom concerning to what form that label takes, and this section addresses some of the possible options for the values representing trust.

There are some types of relationships that easily fit in this paradigm of simply existing or not existing. For example, whether or not we know a person, if we have met a person, or if we are colleagues, is a relationship that exists or does not exist. The same for the trust, it exists a scheme for representing trust with the label "trusted", if trust exists and "blocked", otherwise. The most famous social trust network using this scheme is Epinions [9]. However, trust is not so simple. It is generally established that social trust has a range of strength ([Golbeck, 2005],

[9] http://www.epinions.com/

[Marsh, 1992]).

There are other schemes for representing levels of trust, including scales with more values (such as Richardson et al., (2003)) that used a continuous 0-1 range or with labels rather than numbers (e.g. "very low trust", "low trust", "moderate trust", "high trust", and "very high trust").

Table 2.2 shows some OSNs having some notion of trust that is expressed over a range of values or labels.

Orkut is one of the pioneering and famous Web social networks run by google that plays an important role to communicate and share private and public information in web environment. It facilitates bogging (scraping), personal networking, photo sharing, chatting, private messing, friend search. A worth of cite, a user can have access to others profile information as well as others friends networks. Orkut allows users to rate many features of their friends, including the trustworthiness with zero to three smiley faces.

Overstock [10] has been included in our list. In fact, like other social networks, Overstock encourages users to establish an online presence through personalized homepage with personal history, photos and links to friends. However, unlike typical social networks, Overstock user profiles often include their shopping preferences and return policies. Similar to eBay's Feedback Forum, buyers and sellers on Overstock rate one another at the end of each transaction, and these ratings are aggregated to form a user's feedback based reputation profile. Feedback ratings (-2 to +2) are aggregated by Overstock to form a user's "Business Rating" score. The business rating is the sum of the average rating received from each distinct transaction partner.

We include Overstock network although the "Business Rating" is not explicitly ratings of trust. However, ratings in the context of business are similar to trust in that they provide information about how much one can trust a person to produce a good outcome with respect to a business transaction.

The Epinions web of trust [11] is a who-trust-whom online social network of the general consumer review site Epinions.com. Members or reviewers of the site can decide whether to trust each other or not. All the trust relationships interact and form the web of trust which is then combined with review ratings to determine which reviews are shown to the user. This online product rating site is, so,

[10] http://www.overstock.com/

[11] http://www.epinions.com/

Website	URL	Relationship	Trust scale
Orkut	http://www.orkut.com	Trust	0-3
Overstock	http://www.overstock.com/	Business trust	-2-+2
Epinions	http://www.epinions.com/	consumer	0 or 1
advogato	http://www.advogato.org/	Social trust	4 levels

Table 2.2: The available numerical scale for rating trust in OSNs using

presented by the network of individual users connected by directed trust and distrust links. Edges have the weight +1 for trustful and -1 otherwise. A user can add another one to his web of trust whenever he trusts him. However, not adding him does not mean obligatory that he does not trust him, i.e., simply, it can mean an absence of collaboration or knowledge between them.

Advogato [12] is an online community and social networking site dedicated to free software development, and was created by Raph Levien. Because Advogato was the first website to use a robust, attack-resistant trust metric and to release the underlying code for that trust mechanism under a free software license, it has been the basis of numerous research papers on trust metrics and social networking. In fact, users certify each other in a kind of peer review process and use this information to avoid the abuses that plague open community sites. These certificates are represented as a graph, with each account as a node, and each certificate as a directed edge. Advogato performs certification to four different levels: Observer, Apprentice, Journeyer, and Master. In this thesis, we use Advogato as a testbed for social networking, and since we use trust values in the range [0, 1], we map these labels in [0,1] in the following way: Observer=0.25, Apprentice=0.5, Journeyer=0.75, and Master=1.0.

In this work, we have chosen to represent trust as a continuous variable over a specific range (here, [0, 1]). We discuss the benefits and drawbacks of such an approach here.

It is possible to imagine, for example, one user X giving a value of 50% on how much he trusts another Y. So far, we have a straightforward value placed on trust. The problem arises when that truster X tells another Z how much he trusts the trustee Y. The user Z undertakes the 50% value, but here he considers

[12] http://www.advogato.org/

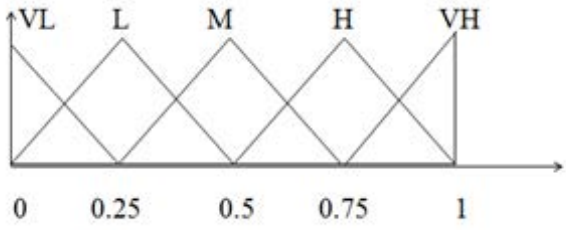


Figure 2.6: An example of stratification of trust values

50% as very high, trusting the trustee with more, perhaps, than what X considers reasonable, since, for X, 50% is more accurately an average trust value.

To contend against subjectivity in values, it is possible to use a stratification of trust. In this work, we use a kind of fuzzy logic of trust, giving each strata a label. So, for example, a trust of 1 would be labeled "very high trust". A suggested stratification is given in Figure 2.6. In fact, we associate a Triangular Fuzzy Number (TFN) that enables us to specify a range for a given trust level instead of giving it a particular discrete value. The meaning of the different linguistic values (fuzzy set) are defined as: Very Low (VL), Low (L), Medium (M), High (H), and Very High (VH), to range users from very untrustworthy to very trustworthy. The advantages of this stratification are that a trust designated as "high" by one user is acknowledged by others as a high trust. Thus, we avoid the problem of "what does a trust of 0.5, or 50%, mean? Is it high or low?", for example. More details are shown in Chapter 7.

2.3.4 Trust vs Reputation

Despite reputation is closely related to the concept of trust, it is not to be confused or treated as trust.

Many definitions of reputation exist. A general definition of this concept given in the free dictionary [13], with some modifications, is:

"*The general estimation in which a person, a group, or an organization is held by the public*".

From this definition, we can derive that reputation is based on different evaluations or opinions. Thus, contrary to the trust, reputation can not be subjective.

[13] http://www.thefreedictionary.com/reputation

Many works differentiate the concepts of trust and reputation, however, some consider that trust is computed from reputation [Bhuiyan *et al.*, 2010b], whereas others consider that reputation is computed from trust.

The first observation considers that trust is based on reputation. This is illustrated by the following plausible statement: I trust you because of your good reputation. The second observation considers that reputation is based on previous judgment to trust an individual or an organization.

In this work, as shown in Figure 1.3, we consider that along with the notion of trust, comes that of reputation. This observation reflects that reputation can be considered as a collective measure of trust which is a personal and subjective phenomenon based on personal experiences of members in an OSN. However, in the absence of personal experience, trust often can be derived from reputation.

2.4 Conclusion

As shown in this Chapter, the social networking sites knew a steady growth and popularity. This growth shows a significant change in the social and personal behaviour of Internet users. OSNs have become essential mediums of communication and entertainment among the young adults. As everything in this world, OSNs can be used for a bad purpose as well as for good. Thus, a trust and reputation management in OSNs is of paramount importance to detect malicious users. The next Chapter provides a comprehensive literature review of proposed trust and reputation models in OSNs.

Chapter 3

Literature Review

3.1 Introduction

The issue of trust has been gaining an increasing amount of attention in a number of research communities including OSNs. There are many different views of how to assess and use trust. As trust is a social phenomenon, the model of trust for the artificial world like Web should be based on how trust works between people in society [Abdul-Rahman et Hailes, 2000]. The wealthy literature growing around methodologies of using trust for OSNs gives a clear indication about the hotness of such an issue. In this work, we categorize the existing review articles managing trust in two broad categories from the point of view of the type of trust: Direct Trust and Indirect or Inferred Trust. Sections 3.2, 3.3 and 3.4, respectively, introduce the existing studies on direct trust, indirect trust and reputation management. Each Section comes to an end with a comprehensive discussion of the studied works. Finally, Section 3.5 reviews the related works dedicated to the enrichment of the FOAF ontology with trust relationships.

3.2 Direct Trust Related Issues in OSNs

Direct trust presents the trust level between two users directly connected. Indeed, in OSNs, these levels are the most important and basic information to propose trust based approaches for protecting user data or inferring trust relationships. In fact, given the direct trust relationships among users of a social network, several methods have been developed to: *(i)* recursively compute the transitive trust (indirect trust) between two non-neighbour users; *(ii)* to ship the access control policies; *(iii)* to compute the reputation of a member in the network or *(iv)* even to compute recommended ratings.

Although how to compute direct trust levels is rarely discussed in the literature, we try to survey the existing works using and considering this trust type. We classify these works to three categories. The first one summarizes works that use direct trust without detailing the computation manner. The second category includes the studied works determining direct trust but manually. The third one, covers works automatically computing direct trust. In the following, we present a survey of the approaches proposed in the different categories, we discuss their proposed trust management solutions and we unveil their drawbacks.

3.2.1 Direct Trust Computing Ignorance

Many works in the literature propose algorithms based on direct trust values to manage their trust models. However, most of these works do not detail the process of computing trust values.

Authors in [Richters et Peixoto, 2010] present an indepth analysis of trust propagation based on the notion of transitivity: if an agent A trusts agent B, and agent B trusts agent C, then, to some extent, the agent A should also trust agent C. Based on this simple concept, a trust metric to assess the reliability of any reachable agent may be inferred. They consider that direct trust is defined *a priori* and they exploit it to define inferred trust. In the same context of computing indirect trust, authors in [Kuter et Golbeck, 2007] propose a new approach that gives an explicit probabilistic interpretation for trust in social networks. They describe SUNNY, a trust inference algorithm that uses a probabilistic sampling technique to assess the user's confidence in the trust information from some designated sources. In addition, to solve the problem of "To which extent can a user trust another one on a service in a social network setting", the authors in [Jiang et Wang, 2011] propose the SWTrust framework, in which they focus on generating small trusted graphs from large OSN. To tackle the key challenge of efficiently discovering short trusted paths, they propose an algorithm for processing a large social network. Authors in [Golbeck, 2006b] also propose a relationship trust computing method, called Tidal, and how those relationships can be used in designing interfaces. As a matter of fact, they present FilmTrust, a website that relies on trust in Web-based social networks to provide predictive movie recommendations. Indeed, trust values assigned by the users of the network for users whom rated a movie are used as weights to compute the movie's average rating. Being so, this weighted average of movie ratings reflects the users' opinion, since the direct or indirect trust values reflect to what extent a user trusts the opinions of other users rating the movie.

An access control mechanism for Web-based social networks was proposed in [Carminati et Ferrari, 2009]. In this work, the authors adopted a rule-based approach for specifying access policies to the resources owned by network participants, and where authorized users are denoted in terms of the type, depth, and trust level existing between nodes directly connected in the network. Similarly, another work, presented in [Abdessalem et BenDhia, 2011], proposed a

reachability-based access control model for OSNs where access control policies are expressed as reachability queries in terms of constraints on the type, direction, distance, and on the trust level according to a given utility between nodes. Yet in the same context, the authors in [Ali *et al.*, 2007] introduced a social access control strategy inspired by a multi-level security [Benantar, 2006] for protecting data in social networks. Instead of clearance levels, they used trust levels to annotate objects and subjects. The trust level of an object is specified by the creator. The trust level of a subject is obtained from an existing trust modelling process. Reading a data object is controlled using the relative trust values of subjects and objects.

All these works do not pay attention to computing direct trust values; In fact, all of them consider direct trust as defined a priori, with random values or they suppose that these values already exist and they do not present how to compute these trust levels.

3.2.2 Manual Management of Direct Trust

Most of OSNs, which allow users to trust each other, propose a manual method by requesting users for instance, to affect zero to five stars to his friend. Orkut, for example, is an OSN that allows users to rate one another's trustworthiness with zero to three smiley faces. Zero smiley face means that the trustor distrusts the trustee, however, giving him three smiley faces, the maximum number of faces a trustor can give, means that he trusts totally the trustee.

3.2.3 Direct Trust Computation

The work carried out in [Singh et Tomar, 2009], proposes an approach computing the similarity score between two-user profiles on the OSN Orkut. This similarity score may be then used as a trust between users. It is measured on the basis of personal information (contact, geographical, educational and professional information), the shared interest (including mutual communities) and mutual social connection (mutual friends). In fact, as shown in Equation 3.1, the similarity score between users U and their friends F_i is the sum of the different criteria presented in Figure 3.1.

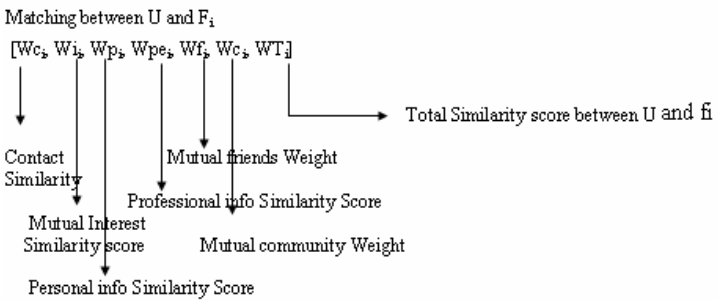


Figure 3.1: The different criteria used for the similarity weights in [Singh et Tomar, 2009]

$$WT_{U \to F_i} = Wc_i + Wi_i + Wp_i + Wpe_i + Wf_i + Wcu_i \tag{3.1}$$

A reputation-based model is proposed in [Nepal *et al.*, 2011] where authors propose a social trust model, called STrust, with the aim of building trust communities. In this trust model, they separate the interactions into two groups: popularity and engagement based interactions. The popularity based interactions are in general based on the trustworthiness of a member in the community. Similarly, the engagement based interactions are in general based on how much a member trusts other ones in the community.

3.2.4 Discussion

At a glance, Table 3.1 sketches the surveyed approaches based on direct trust values using the following criteria:

- **Approach's Aim**: this criterion presents the aim of using direct trust. It can be used to provide access control, to compute indirect trust or to manage a personal reputation;
- **Direct Trust Computing**: this criterion checks whether, in the given approach, the direct trust is computed or not;
- **Direct Trust Metrics**: this criterion gives the metrics considered to compute the direct trust;

	Approach's Aim	**Direct Trust Computation**	**Direct Trust Metrics**	**Direct Trust Range Values**
Orkut Büyükkökten, 2004	Social Networking	Yes	Manually	[0,3]
Ali et al., 2007	Access Control	No	-	[0,1]
Kuter and Golbeck, 2007	Indirect Trust	No	-	[0,1]
Carminati and Ferrari, 2009	Access Control	No	-	[0,1]
Golbeck, 2009	Movie Recommendation	No	-	[1,10]
Singh and Tomar 2009	User profile Investigation	Yes	Users' Interests Personal information	[0, $+\infty$]
Richters and Peixoto, 2010	Indirect Trust	No	-	[0,1]
Abdessalem and Ben Dhia, 2011	Access Control	No	-	[0,1]
Jiang and Wang, 2011	Indirect Trust	No	-	[0,1]
Nepal, et al., 2011	Reputation	Yes	Users' Interactions	[0,1]

Table 3.1: Comparison of surveyed works exploiting direct trust

- **Direct Trust Range Values**: this criterion yields the range of the possible values of the direct trust.

Several approaches managing trust in social networks were based on direct trust values. Some works used the direct trust to solve trust-inference problems ([Jiang et Wang, 2011], [Kuter et Golbeck, 2007] and [Richters et Peixoto, 2010]), others are interested in managing the access control policies ([Abdessalem et BenDhia, 2011], [Ali *et al.*, 2007] and [Carminati et Ferrari, 2009]), while [Golbeck, 2006b] focuses on creating predictive movie recommendations and [Nepal *et al.*, 2011] concentrates on managing users' reputations. Whereas [Singh et Tomar, 2009] investigates users profiles to measure similarity scores between users in OSNs.

Most of the existing trust computing algorithms are based on direct trust values to manage their trust models. However, these works ([Abdessalem et BenDhia, 2011], [Ali *et al.*, 2007], [Carminati et Ferrari, 2009], [Golbeck, 2006b], [Kuter et Golbeck, 2007], [Richters et Peixoto, 2010]) suppose that these values already exist and they do not present how to compute these trust levels. Consequently, there is a large gap in their trust model's definition. Authors in [Jiang et Wang, 2011] admit that the direct trust is based on users' interests, but they did not explain how to compute the interests similarity, nor, even, refer any previous work being interested on managing direct trust. Direct trust values computed respectively, in [Nepal *et al.*, 2011] and [Singh et Tomar, 2009]. However, this computation is only based on users' interest and personal information metric in [Singh et Tomar, 2009], as well as it is only based on past interactions of users in [Nepal *et al.*, 2011].

As mentioned above, Orkut allows the users for trusting each other manually. However, this method suffers from many drawbacks:

1- In an OSN, the user can have hundreds of friends and to evaluate the trust of each contact, he has to spend a lot of time assigning these trust values.

2- The user can have a high propensity to trust (or can be naive) and can be likely to give high ratings to anyone. If this is the case, less trustworthy people may receive higher ratings because of this user.

3- The user can feel bad about assigning low ratings, and gives high ones (even though the information is kept private), and the opposite can be true.

4- The user only bothers to rate people that are very trustworthy, and does not spend time on anyone else. In this case, the user may providing very accurate

information about the people who have been rated.

5- The user does not understand the scale, and may be mis-assigning ratings. For example, if the user incorrectly treats the middle of the scale (a rating of 0.5) as neutral and lower ratings as expressions of distrust, then the all ratings from that user will be skewed.

Regarding the range values of direct trust, most of the reviewed works used a continuous [0-1] range, where 0 denotes complete distrust and 1 absolute trust between the trustor to the trustee user. In [Singh et Tomar, 2009], the trust values stand within the range [0, $+\infty$], since a measured score that corresponds to the trust level is not more than a sum of different values. In this case, the user can not make difference between benovelent or malicious users. For example with a value of 20, a user can not answer to the question: "Is the trustee considered as trustworthy or untrustworthy?". We think that these scores can be used rather for ranking users and not for evaluating their trust. Otherwise, these values should be normalized to facilitate their understanding by the users.

In this thesis, in Chapter 5, we introduce a novel approach, $IRIS$, that aims at computing direct trust in OSN. Thus, a management trust model will be able to apply its trust computing algorithms with real, neither random, nor manual, direct trust values that make these algorithms more reliable. Moreover, our method considers users' interactions, interests as well as their relationship types. The computed direct trust values belong to the continuous [0, 1] range that is straightforward applied by the surveyed methods. It is worth of mention that our approach could be easily adapted to take any social network as input and generate its correspondent trusted social network.

3.3 Trust inference in OSNs

The structure of the social network as well as explicit trust values between directly connected users can be used for the computation of the trust inference estimation. In the following, we review a collection of trust inference mechanisms designed for determining inferred trust values on OSNs.

3.3.1 The TidalTrust algorithm

The TidalTrust algorithm [Golbeck, 2005] considers the trust values to be numbers in a continuous range [0, 10]. A social network is considered as a directed graph used to represent trust relationships. Each edge has a label within the range [0, 10]; i.e., 10 means full trust and 0 means full distrust. Authors considered two restrictions in their algorithm leading to more accurate results in many cases. First, they showed that trust values inferred through shorter paths may be more accurate. So, they only considered the shortest paths from source to sink in their inference algorithm. Second, they extracted from their analysis that the most trustworthy information usually comes from the highest trusted neighbors. Thus, they computed a trust threshold for trust network in their algorithm and applied it when combining trust values. This means that, in the combination process to compute the trust t_{os} between the source o and the sink s (Eq. 3.2), only neighbors having an associated trust value greater than or equal the threshold (max in Eq. 3.2) are considered.

For the simplest case, suppose that node o in the trust network is interested in computing its trust value to node s which is not directly connected to it. Nodes o and s are connected by one or more trust paths of length 2. First, all these shortest paths are discovered. Second, they are processed in order to set a trust threshold max, which is used to discard trust paths consisting of edges with a trust value less than max. Then, the predicted trust existing between o and s, denoted $t_{o,s}$ is computed as follows:

$$t_{o \to s} = \frac{\sum_{j \in \mathcal{ADJ} | t_{o \to j} > max} t_{o \to j} t_{j \to s}}{\sum_{j \in \mathcal{ADJ} | t_{o \to j} > max} t_{o \to j}} \tag{3.2}$$

In Eq. 3.2, $t_{o \to j}$ (resp. $t_{j \to s}$) denotes the direct trust value existing between nodes o and j (resp. j and s), whereas $\mathcal{ADJ}$ denotes the set of nodes with an incoming edge exiting from o. If the distance between o and s is greater than 2, then the formula above is applied recursively, until the trust between o and s is computed.

3.3.2 The RN-Trust algorithm

To infer the trust value from o to s, the authors in [Taherian *et al.*, 2008] consider the trust network modeled by the graph G as a resistive network, called Res. They modeled each trust relationship between two nodes in G by a resistor in Res such that the more the trust value, the less the value of the corresponding resistor is. Thus, they use a mapping function to compute the resistance of each resistor.

Given t as the trust value from v and v' in G, the corresponding resistor with resistance R between v and v' in Res is calculated as $R = -log(t)$. To compute the inferred trust value $t_{o \to s}$ between the source node o and the sink node s, they first compute the equivalent resistance between o and s, $R(o, s)$, to have $t_{o \to s} = 10^{-R(o,s)}$. For a given path, $R(o, s)$ can be considered as the combination of two series' resistors $R(o, v)$ and $R(v, s)$ which are serially connected as shown in Eq.3.3.

$$R(o, s) = R(o, v) + R(v, s) \tag{3.3}$$

Whenever there exist two paths, the inferred value $R(o, s)$, is equivalent to the combination of two parallel connected resistors $R_1(o, s)$ and $R_2(o, s)$ as shown in Eq.3.4.

$$R(o, s) = \frac{R_1(o, s) \times R_2(o, s)}{R_1(o, s) + R_2(o, s)} \tag{3.4}$$

Notice that, in RN-Trust, all paths, and not only the shortest paths, are considered to compute the final inferred trust value.

3.3.3 The SWTrust algorithm

In [Jiang *et al.*, 2014], a framework is proposed to generate trusted graphs for trust evaluation in an OSN to develop various algorithms for building a trust network and generating a trusted graph. To compute the inferred trust, the authors propose an algorithm, called *Distributed Breadth-first Search* (DBFS), to look for trust paths between the source and the sink. In this work, every path linking the source with the sink's neighbors with length less than or equal to L is

considered as a trust path [Wang et Wu, 2011]; L represents a fixed value to limit the depth of the paths' search. For each path, the direct trust values are multiplied and the average of the paths' values is used to compute the inferred trust between the source and the sink. In fact, the obtained average value is multiplied by the average of direct trust values between the sink and its neighbours to retrieve the inferred trust value.

3.3.4 The MQCTT model

In [Liu, 2013], a general concept of Quality of Trust Transitivity (QoTT) is proposed. This new concept takes into account different attributes as trust, social relationships and preference similarity to select social trust paths and guarantee a certain level of quality in trust transitivity. After the selection of trust paths, the authors propose a new Multiple QoTT Constrained Trust Transitivity (MQCTT) model to compute indirect trust values. Indeed, basing on the trust properties: subjectivity [Hardin, 2004], transitivity [Christianson et Harbison, 1997], [Jøsang et Pope, 2005] and "decay" [Christianson et Harbison, 1997], [JÃ¸sang et Pope, 2003], the transitive trust values between users are computed.

3.3.5 Discussion

Despite the popularity of TidalTrust, it flogs out some drawbacks. First, it is worth of mention that, with the restriction on the paths' length, some useful information may be lost. Yet, the most important problem is about single paths between the source o and the sink s. In fact, if only one path exists between o and s, then $t_{o \to s} = t_{j \to s}$, being j the last node in this path and the neighbour of s. Suppose that there is only a long path between the source and the sink, and all nodes in this path have trust value equal to 1 (low trust) to their neighbours, except the neighbour of the sink, having a trust value 10 (full trust) to the sink. With TidalTrust, the computed trust value from o to s is 10. However, it is clear that the inferred trust value should be smaller since the quality of the path is very bad (a long path with very low trust values) and there is no relation between the intermediary trust values with the final inferred trust.

The RN-Trust algorithm [Taherian *et al.*, 2008] tries to solve some of TidalTrust problems considering all paths to compute the inferred trust between two unrelated users. There is no limit for the path length nor a threshold for the

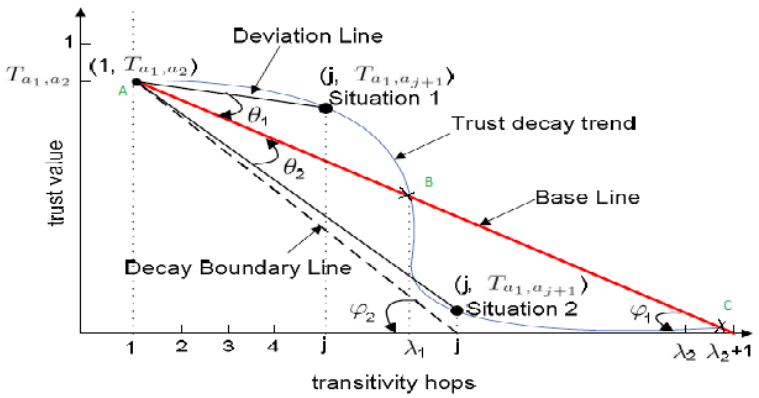


Figure 3.2: Trust transitivity model in [Liu, 2013]

trust values between intermediary nodes. Nevertheless, it risks to use many unimportant information by considering the weakest trust values which decrease the inferred trust value. Indeed, suppose that there are two paths from the source to the sink, such that each path contains one intermediary node. The trust values in the first path are successively 10 and 1; in the second path, the trust values are 1 and 10. It is clear that the first path is trustworthy but the other one is not. Despite this big difference, RN-Trust does not consider it and takes both of them as trustworthy.

The algorithm SWTrust [Jiang et Wang, 2011] is developed to perform six methods to measure inferred trust values and determine the best one. By experiments, the authors showed that trust values inferred through the Wave function, that averages all inferred trust values for all paths, are the most accurate ones. Thus, the most trustworthy information comes from considering all available information. This finding decreases the inferred trust quality since it considers, for example, the opinions of malicious nodes. Another problem results from this function that averages inferred values, between 0 and 1, significantly decreasing the inferred value that tends towards zero. This would not help the truster to take a decision about its trustees.

As mentioned above, the transitive trust presented in the MQCTT model is based on the "decay" property. In fact, as depicted in Fig. 3.2, the curve of the trust transitivity is defined with 3 points defined as A(1,T_{a_1,a_2}), B(λ_1, y_1) and C(λ_2, y_2), where a_1 is the sink, a_2 is the target, T_{a_1,a_2} is the direct trust from a_1 to a_2, the abscissas λ_1 and λ_2 are given by the source user to indicate when the

target user begins to become stranger and becomes a stranger, respectively. The ordinate y_2 is near to 0, however y_1 is not specified but it presents the ordinate of the intersection point between the curve and the Base Line drawn on Fig. 3.2. Then, indirect trust values along the curve are determined by the intersection angle θ (i.e. θ_1 in situation 1 in Fig. 3.2 and θ_2 in situation 2). However, with only the known points A, B and C, and respecting the decay property, many shapes can take place, as e. g. those in Fig. 3.3, and the question that can raise is which one we must consider.

We should also note, that a shape presents only one trusted path, while authors propose an algorithm H-OSTP-K that extracts the most trusted K paths. However, the authors do not show which one is considered. If many paths are taken into account, they do not describe how they are used to compute the indirect trust.

In addition, the different attributes of the quality of trust (QoT), as the direct trust, are not used to compute the transitive trust. They are only considered on extracting the trusted paths with the H-OSTP-K algorithm. Furthermore, this latter has not considered the correlation between the QoT attributes, and the utility function (Eq. 3.5) used to compute the utility of a path is too simple to embrace all QoT attributes.

$$F_{p(a_1,...a_n)} = \omega_T * T_{p(a_1,...a_n)} + \omega_r * r_{p(a_1,...a_n)} + \omega_\rho * \rho_{p(a_1,...a_n)} \tag{3.5}$$

In Eq. 3.5, the trust attributes T, r and ρ present respectively Trust between participants, Role Impact Factor and Social Intimacy Degree. ω_T, ω_r, ω_ρ are their respective weights to compute the utility of the path. However, how to choose these weights or to help source participants to precise them, is not mentioned by the authors.

In the following, we unveil, through an illustrative example, the main weaknesses of the surveyed algorithms.

3.3.6 Analysis by examples

In the following, we apply the reviewed algorithms on a sample trust network shown in Fig. 3.4. The results of applying TidalTrust, RN-Trust and SW-Trust are shown in Table 3.2, Table 3.3 and Table 3.4 respectively. To apply RN-Trust and SW-Trust, the trust values are scaled down to the range of

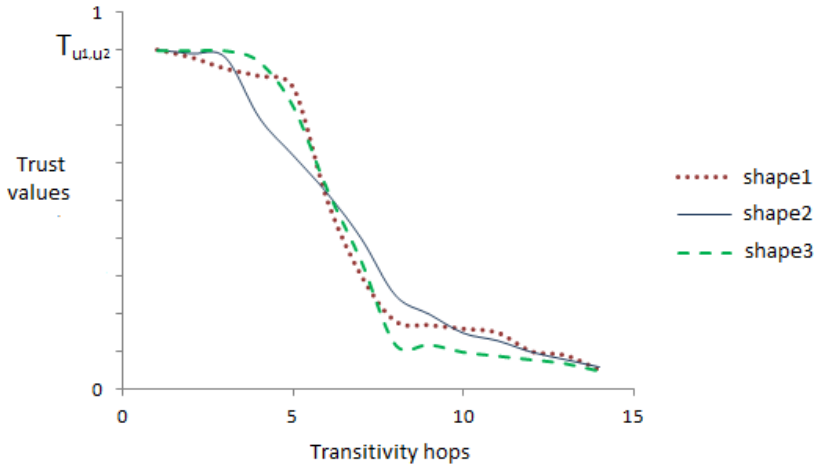


Figure 3.3: Different shapes give different transitive trust values

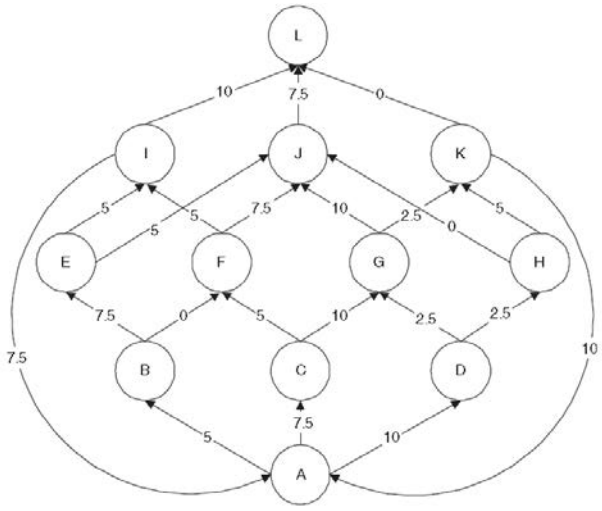


Figure 3.4: A sample trust network from [Taherian *et al.*, 2008]

	A	B	C	D	E	F	G	H	I	J	K	L
A	-	5.00	7.50	10.00	**7.50**	5.00	2.50	2.50	5.00	**10.00**	2.50	7.50
B	7.50	-	7.50	10.00	7.50	0.00	5.70	2.50	5.00	5.00	2.50	8.80
C	7.50	5.00	-	10.00	7.50	5.00	10.00	2.50	5.00	10.00	2.50	7.50
D	10.00	5.00	7.50	-	7.50	3.00	2.50	2.50	5.00	5.00	3.80	3.00
E	7.50	5.00	7.50	10.00	-	3.00	5.70	2.50	5.00	5.00	2.50	8.80
F	7.50	5.00	7.50	10.00	7.50	-	5.70	2.50	5.00	7.50	2.50	7.50
G	10.00	5.0	7.50	**10.00**	7.50	3.00	-	2.50	5.00	10.00	2.50	7.50
H	10.00	5.00	7.50	10.00	7.50	3.00	5.70	-	5.00	0.00	5.00	0.00
I	7.50	5.00	7.50	10.00	7.50	5.00	5.70	2.50	-	10.00	2.50	10.00
J	-	-	-	-	-	-	-	-	-	-	-	7.50
K	10.00	5.00	7.50	10.00	7.50	5.00	2.50	2.50	5.00	10.00	-	0.00
L	-	-	-	-	-	-	-	-	-	-	-	-

Table 3.2: Results of TidalTrust for the trust network in Fig. 3.4.

$[0, 1]$. In addition, the outputs are scaled up to the range of $[1, 10]$ for better comparison with TidalTrust.

To evaluate the results of respectively TidalTrust, RN-Trust and SW-Trust algorithms, we analyse the trust values from some sources to some sinks and we discuss if the results are reasonable with regard to properties of trust in real OSNs.

As already mentioned, TidalTrust has a major disadvantage on calculating trust through a single path. In fact, it considers only the shortest paths to infer trust. In the case of the existence of only one path between the origin and the sink, the inferred trust value will be strongly equal to the direct trust given to the sink by its adjacent in this path: $t_{o\to s} = \frac{\prod t_i \times t_{adj\to s}}{\prod t_i} = t_{adj\to s}$. We call this problem *the unique path problem* and we show through the two next examples that, in this case, the returned inferred values are not acceptable.

Let us consider the trust value from A to E in Fig. 3.4. It is clear that there is only one path from A to E which is $A \to B \to E$. TidalTrust gives value 7.5 for the trust from A to E. However, regarding the trust values in this path, at a glance, the trust value from A to E should be less than that calculated by TidalTrust. If A has not full trust on B, how can it obtain the same trust

	A	B	C	D	E	F	G	H	I	J	K	L
A	-	5.00	7.5	10.00	3.70	3.70	**7.90**	2.50	4.30	**8.10**	4.00	6.90
B	2.8	-	2.10	2.80	7.50	0.00	2.20	0.70	3.70	4.90	1.10	5.00
C	4.70	2.30	-	4.70	1.80	5.00	10.00	1.20	3.80	10.00	3.60	8.00
D	3.00	1.50	2.30	-	1.10	1.10	2.50	2.50	1.30	4.20	3.00	3.50
E	3.70	1.90	2.80	3.70	-	1.40	2.90	0.90	5.00	5.00	1.50	6.70
F	3.70	1.90	2.80	3.70	1.40	-	2.90	0.90	5.00	7.50	1.50	7.30
G	2.50	1.20	1.90	2.50	0.90	0.90	-	0.60	1.10	10.00	2.50	7.70
H	5.00	2.50	3.70	5.00	1.90	1.90	3.90	-	2.20	0.00	5.00	3.50
I	7.50	3.70	5.60	7.50	2.80	2.80	5.90	1.90	-	6.00	3.00	10.00
J	-	-	-	-	-	-	-	-	-	-	-	7.50
K	10.00	5.00	7.50	10.00	3.70	3.70	7.90	2.50	4.30	8.10	-	0.00
L	-	-	-	-	-	-	-	-	-	-	-	-

Table 3.3: Results of the RN-TRUST for the trust network in Fig. 3.4.

from B to E?. In addition, if we focus on the trust value from G to D. The TIDALTRUST algorithm returns the value of 10, i.e., G gives a full trust to D. Note that there is also only one path from G to D which is $G \rightarrow K \rightarrow A \rightarrow D$. Normally, suggesting G to totally trust D means that all direct trust values in the concerned path are full trusts and equal to 10. However, it is easy to remark that see that the first trust value between G and K is $2.5 << 10$. When G has not full trust on K, how can TIDALTRUST obtain full trust from G to D?

It is worth of mention, that not only considering shortest paths can reduce the number of trust paths to one, but also considering the highest trust values can reduce the number of trust paths to one. Thus, suffering from what we call *unique path problem* will obviously increase for TIDALTRUST. Indeed, this insufficiency is highlighted if we pay attention to the computation of the trust value from A to J. Many paths exist between them, we cite $A \rightarrow B \rightarrow E \rightarrow J$; $A \rightarrow B \rightarrow F \rightarrow J$; $A \rightarrow C \rightarrow F \rightarrow J$; $A \rightarrow C \rightarrow G \rightarrow J$; $A \rightarrow D \rightarrow G \rightarrow J$ and $A \rightarrow D \rightarrow H \rightarrow J$. By taking into account the highest values, e. g., for values greater than 7.5, only $A \rightarrow C \rightarrow G \rightarrow J$ will be considered. TIDALTRUST returns 10 as trust value, even though A has not full trust to C. So, how can it obtain full trust from A to J?

Even though RN-Trust addresses the *unique path problem*, i.e., it gives for example 3.7 for the trust from A to E which seems to be more acceptable then 7.5, it, nevertheless, suffers from other problems that we try to resume in the following.

Let us consider the trust value from A to G; there are two paths between them which are respectively $A \to C \to G$ and $A \to D \to G$. The RN-Trust algorithm gives the value 7.9 for the trust of A to G. However, when considering the first path, the inferred trust value should be equal to or less than 7.5. Considering now the second path, when A has full trust on D and D gives 2.5 to G, how can A give 7.9, which is so higher than 2.5, to G? Another problem also rises when we try to compute the trust value from A to J. Many paths exist between them and the RN-Trust algorithm gives 8.1 as the inferred trust value from A to J. We analyse each path independently to prove that none of these paths allows to obtain a trust value equal to 8.1. Beginning by $A \to B \to E \to J$, the value should be no greater than 5, and it is also the case for $A \to B \to F \to J$, because the direct trust from A to B is 5. Considering $A \to C \to F \to J$ and $A \to C \to G \to J$, the value should be no greater than 7.5, because it is the only direct trust value coming from the source A in these paths. Regarding to $A \to D \to H \to J$, the value should be 0, because it is the only direct trust information concerning J. Finally, it should be less than 2.5 for the path $A \to D \to G \to J$. As we see, when following each of the paths, we do not obtain a value higher than 7.5. So, when combining the results of the paths, the obtained value should be less than 7.5, but what to say about 8.1?

Inferred trust values in SWTrust do not suffer from this kind of problems, since these values are very low regarding the existing direct trust values. These low values result from the multiplication of the intermediate trust values between the origin node and the sink one. In different circumstances, these indirect trust values are meaningless and do not help the origin node to decide whether to trust or to distrust the sink. In fact, authors in [Jiang et Wang, 2011] propose a threshold equal to 5 on a scale of $[1, 10]$ as the best choice when needing to predict to trust or not; in that case, it is too rare to trust someone, even if (s)he really deserves trust. As shown in Table 3.4, none of the nodes can be trusted since they all have an inferred trust values lower than or equal to 5. In addition, if we consider the trust value from C to J. Many paths exist between

	A	B	C	D	E	F	G	H	I	J	K	L
A	-	5.00	7.50	10.00	3.70	1.56	5.46	2.50	1.23	1.87	1.56	1.16
B	1.40	-	1.05	1.40	7.50	0.00	0.76	0.35	1.87	0.75	0.21	0.41
C	2.10	1.10	-	2.18	0.82	5.00	10.00	2.50	0.11	**0.45**	0.93	0.23
D	0.93	0.46	0.70	-	0.35	0.14	2.50	2.50	0.11	0.11	0.45	0.93
E	3.75	1.875	2.81	3.75	-	0.58	2.05	0.93	5	5	0.58	0.94
F	3.75	1.875	2.81	3.75	0.14	-	2.05	0.93	5.00	7.50	0.58	1.19
G	2.5	1.25	1.875	2.50	0.93	0.40	-	0.62	0.31	10.00	2.50	0.94
H	5.00	2.50	3.75	5.00	1.875	0.78	2.73	-	0.625	0.00	5.00	0.82
I	7.50	3.75	5.62	7.50	2.81	1.17	4.10	1.875	-	1.40	1.17	10.00
J	-	-	-	-	-	-	-	-	-	-	-	7.50
K	10.00	5.00	7.50	10.00	3.75	1.56	5.46	2.50	1.25	1.875	-	0.00
L	-	-	-	-	-	-	-	-	-	-	-	-

Table 3.4: Results of the SW-TRUST for the trust network given in Fig. 3.4.

them. The SWTRUST algorithm gives 0.45 as the inferred trust value between C and J. Regardless the set of paths between C and J, there is a trustworthy path $C \rightarrow G \rightarrow J$. So, the value should be high and even equal to 10, and not unimportant as 0.45.

As mentioned above, when computing the transitive trust, the MQCTT model suffers from different problems due to the decay property of trust where trust is divided into three phases (slow decay phase, fast decay phase then slow decay phase as in Fig. 3.2). In fact, many cases exist where trust does not decay quickly, as in the fast decay phase. Let us take the example in Fig. 3.5. Suppose the source user assigns 2 and 5 for both λ_1 and λ_2. Thus, the indirect value that the source agent u_1 affects to the target u_5 is near to 0. However, having all direct trust values between u_1 and u_5 equal to 10 can not imply a very weak value near to 0. We think it is better when u_1 considers the target as stranger than totally distrusting him.

The most reasonable setting for a local trust metric is the one in which every user runs it from its personal point of view [Hamdi *et al.*, 2012a], this property of subjectivity is not respected by most of the related works. So, in Chapter 6, we introduce TISON for large OSNs, in which each user can select the next hop

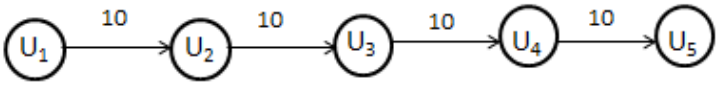


Figure 3.5: Trust transitivity example

based on his own knowledge of connected neighbourhood as well as his behaviour.

3.4 Reputation Management in OSNs

An important objective in this thesis is managing reputation in OSNs. In Chapter 7, we specify first, the data from OSNs that are used to determine the reputation of OSNs' users; Then, we define how these data can be structured using simple or fuzzy clustering methods, since clusters are able to translate the human thinking to computer-understandable models [ZHAO, 2012].

Thus, we split the following into two subsections: The first subsection reviews the clustering algorithms and applications. The second subsection scrutinizes the literature work on reputation management in OSNs.

3.4.1 Clustering Analysis

Clustering analysis groups objects based only on data that describes the objects and their relationships. The goal is that the similarity between objects within a cluster should be higher than those outside the cluster. The greater the similarity within a cluster and the more the difference between clustering are, the better the clustering is.

Different methods of clustering data are described in the literature. In this Section, we distinguish various categories of clustering: Hierarchical versus partitioning and exclusive versus fuzzy [Jain et Dubes, 1988].

Hierarchical methods

A hierarchical clustering is obtained when clusters are permitted to have sub-clusters. This clustering is a set of nested clusters that are organized as a tree also known as a dendrogram. Each node in the tree (except for the leaf nodes) is the union of its children (sub-clusters) and the root of the tree is the

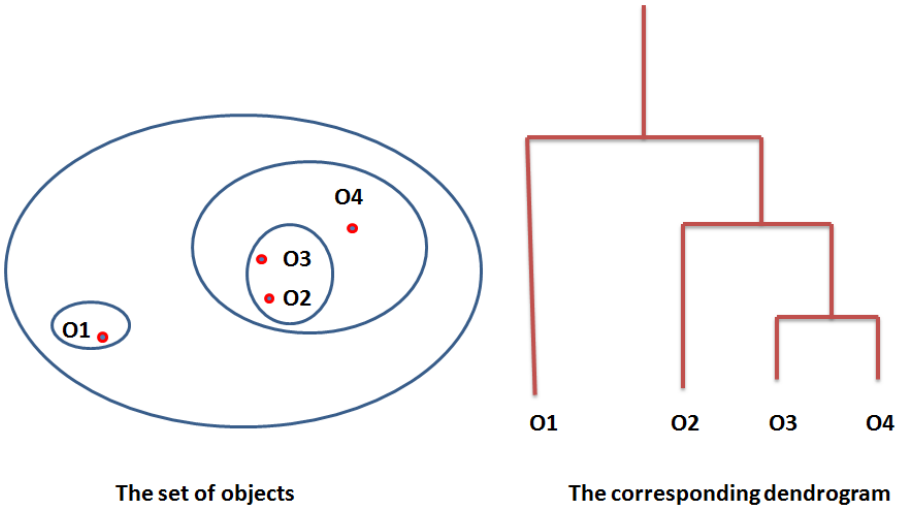


Figure 3.6: Hierarchical Divisive Clustering

cluster containing all the objects. Hierarchical clustering methods are classified into agglomerative (bottom-up) and divisive (top-down) ([Jain et Dubes, 1988], [Kaufman et Rousseeuw, 1990]). In one hand, the agglomerative clustering algorithms start with the points as individual clusters and at each step, merge the closest pair of clusters. These algorithms do not scale well, their time complexity is at least $O(n^2)$, where n is the number of total objects, moreover, they can never undo what was done previously. In the other hand, divisive clustering algorithms start with one, all-inclusive cluster, and recursively split a cluster until a stopping criterion is achieved (each cluster contains a point or there is the requested number k of clusters). As shown in Figure 3.6, the dendrogram can be cut at different levels to yield different partitions of the data objects.

Partitioning methods

A partitioning clustering is simply a division of a database of objects into non-overlapping clusters such that each object is in exactly one cluster. An example of a partitioning clustering is illustrated in Figure 3.7.

Partitioning methods have advantages in applications requiring large data sets for which the construction of a tree is computationally expensive. A problem accompanying the use of a partitioning algorithm is the choice of the number of desired output clusters. Iterative optimization partitioning algorithms are

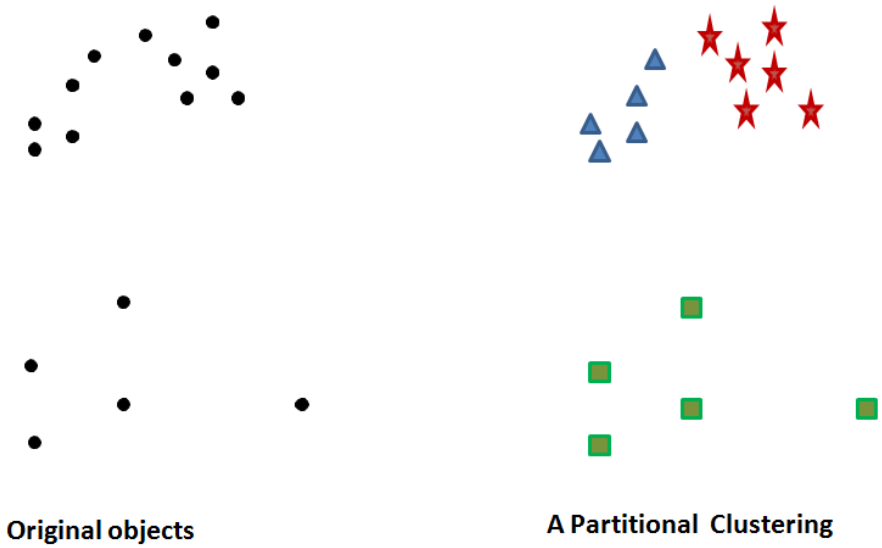


Figure 3.7: Partitioning Clustering

subdivided into k-medoids and k-Means methods [Berkhin, 2002].

K-Means Methods

The k-Means algorithm [Hartigan et Wong, 1979] is by far the most popular clustering tool used in industrial and scientific applications because it is easy to implement, and its time complexity is linear. The name k-Means comes from representing each of k clusters C_j by the mean (or weighted average) c_j of its points (objects), the so-called *centroid* [Berkhin, 2006]. The k-Means algorithm aims to partition the objects into mutually exclusive clusters where objects within each cluster remain as close as possible to each other but as far as possible from objects in other clusters. The distances used in clustering in most of the times do not actually represent the spatial distances.

The pseudo-code of the k-Means algorithm is sketched by Algorithm 1. After setting a number k to define the number of desired clusters, the algorithm chooses k clusters starting centroids randomly as initial estimates of the cluster centroids (line 2). In line 4, the algorithm assigns each object x_i to the closest cluster centroid by calculating the distances between x_i and the k centroids. Then, it updates and recomputes the cluster centroids as the mean of the objects assigned to it, using the current cluster assignment as mentioned in line 5. The algorithm repeats the previous 2 steps until a convergence criterion is met (line 6). Typical convergence criteria are: no (or minimal) reassignment of patterns to new cluster

Algorithm 1: Basic K-means Algorithm

```
1 begin
2     Select k points as initial centroids;
3     repeat
4         Create k clusters by assigning each point to the closest centroid;
5         Recompute the centroid of each cluster;
6     until convergence criterion is met;
```

centroids, or minimal decrease in squared error.

However, k-Means suffers from a major disadvantage, it is sensitive to the selection of the initial clusters and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen. In addition, k-means works conveniently only with numerical attributes and can be negatively affected by a single outlier [Jain *et al.*, 1999]. Figure 3.8 shows seven two-dimensional objects. If we start with objects T, U and V as the initial means around which the three clusters are built, then we end up with the three clusters {T}, {U, V} and {W, X, Y, Z} as shown in Fig. 3.8(a). The squared error criterion value is much larger for this partition than for the best partition {{T, U, V}, {W, X}, {Y, Z}} shown in Fig. 3.8(b), which yields the global minimum value of the squared error criterion function for a clustering containing three clusters. The correct three-cluster solution is obtained by choosing, for example, T, W, and Y as the initial cluster means.

K-Medoids Methods

In k-medoids methods a cluster is represented by one of its object (medoid) which is the most appropriate within it. When medoids are identified, clusters are defined as subsets of objects close to respective medoids, and the objective function is defined as the averaged distance or another similarity measure between an object and its medoid.

Representation by k-medoids has two advantages. First, it covers any attribute types, and, second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is lesser sensitive to the presence of outliers.

An early version of k-medoid methods is the algorithm CLARA (Clustering LARge Applications) [Kaufman et Rousseeuw, 1990]. CLARA knows a progress

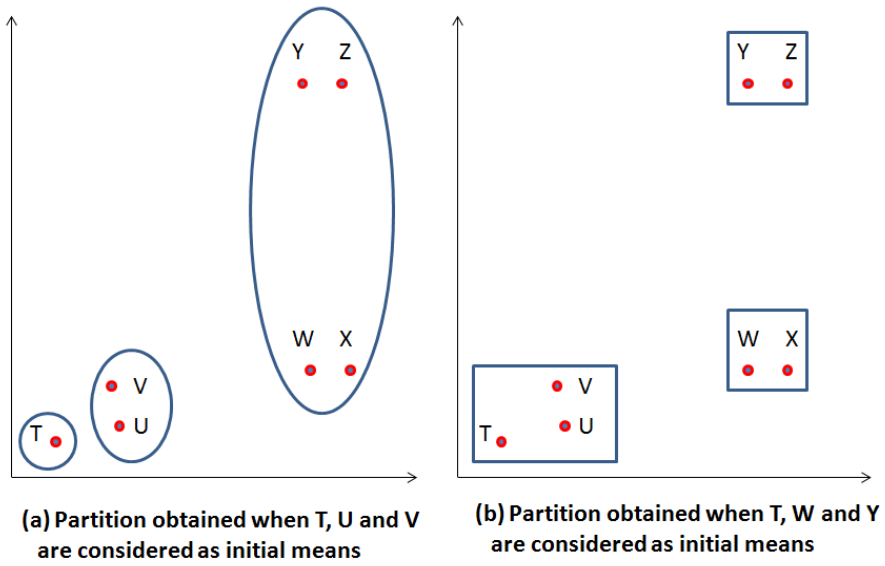


Figure 3.8: The k-means algorithm is sensitive to the initial partition.

with the algorithm CLARANS (Clustering Large Applications based upon RANdomized Search) in the context of clustering in spatial databases [Ng et Han, 1994]. The complexity of CLARANS is about $O(n^2)$, where n is the number of objects. CLARANS were also extended in [Ester *et al.*, 1995], where a database interface for clustering in large spatial databases is proposed, this interface is based on the spatial geographic access method R*-tree [Beckmann *et al.*, 1990].

Fuzzy clustering

Traditional clustering approaches generate a set of clusters; each object belongs to a unique cluster. Hence, the clusters in a hard clustering are disjoint. Fuzzy clustering extends this notion to associate each object with every cluster using a membership function [Zadeh, 1965]. The most popular algorithm is Fuzzy c-means (FCM). It is a data clustering technique where each object belongs to a cluster to some degree that is specified by a membership grade. This technique was originally introduced by Dunn in 1973 [Dunn, 1973] and then developed by Bezdek in 1981 [Bezdek, 1981] as an improvement on earlier exclusive clustering methods. The main advantage of fuzzy c-means clustering is that it allows gradual memberships of data objects to clusters measured as degrees in [0,1] (the membership weights of one object in the different clusters must sum to 1). This

Algorithm 2: The Fuzzy c-means Algorithm

```
1 begin
2     Initialize U = u_ij matrix, U^0;
3     k ← 1;
4     repeat
5         At step k, compute the vectors of centroids C^k = [c_j] with U^k
          respecting Eq. 3.6;
6         Update U^k, U^{k+1} respecting Eq. 3.7;
7         k ← k + 1;
8     until U^{k+1} − U^k < ε;
```

yields the flexibility to express that data objects can belong to more than one cluster [Bora et Gupta, 2014].

The pseudo-code of the FCM algorithm is sketched by Algorithm 2. In line 2, FCM selects an initial fuzzy partition of the N objects into N' clusters by selecting the $N \times N'$ membership matrix U. An element u_{ij} of this matrix represents the grade of membership of object x_i in cluster c_j, such as $u_{ij} \in [0, 1]$. Then, FCM computes the centroids of the N' clusters (cf. line 5). In fact, each new centroid is recomputed as the mean of the objects assigned to its corresponding cluster, multiplied by their membership degrees as described in Eq. 3.6. The process is repeated until elements in U do not change significantly.

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m . x_i}{\sum_{i=1}^{N} u_{ij}^m}; j \in [1, N'] \tag{3.6}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} (\frac{\|x_i - c_j\|}{\|x_i - c_k\|})^{\frac{2}{m-1}}} \tag{3.7}$$

Figure 3.9 shows an example of execution of the FCM algorithm. In the last step (e), FCM produces the two fuzzy clusters C1 and C2 depicted by ellipses where each cluster is a fuzzy set of all the objects. The objects have membership values in [0,1] for each cluster. For example, fuzzy cluster C1 could be compactly described as {(1, 1.0); (2, 1.0); (3, 1.0); (4, 0.48); (5, 0.0); (6, 0.0); (7, 0.0)} and C2 could be described as {(1, 0.0); (2, 0.0); (3, 0.0); (4, 0.52); (5, 1.0); (6, 1.0); (7, 1.0)}.

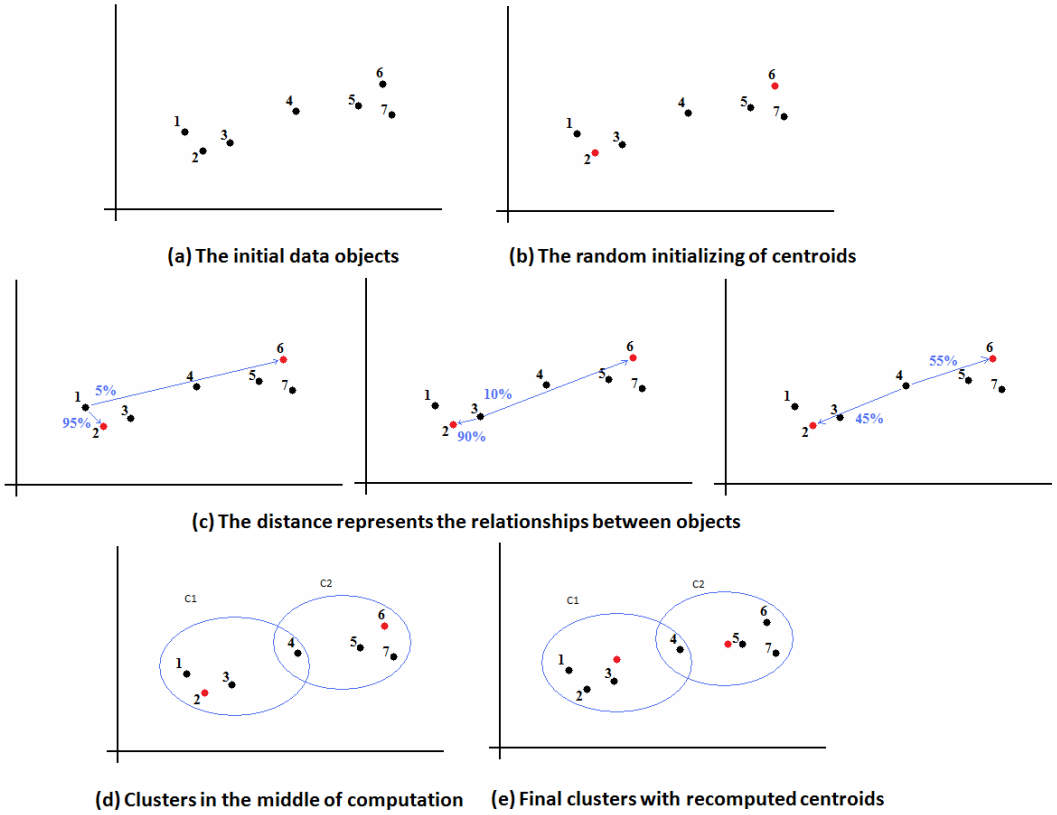


Figure 3.9: Execution Example of Fuzzy C-Means Algorithm

3.4.2 Online Reputation Management

The reputation management in OSNs deals with monitoring and influencing the online record of a person, a product or an organization. OSNs offer increasingly simple ways to publish and disseminate personal or opinionated information, which can rapidly have a disastrous influence on the online reputation of some of the entities [Portmann, 2012]. In adddition, OSNs are environments where a distributed reputation system, i.e. without any centralised function, is better suited than a centralised system. In a distributed system there is no central location for submitting ratings or obtaining reputation scores of others. In the literature, several approaches are designed to describe how to identify the reputation of users. In the remainder, we present and describe a set of some of the most representative reputation approaches for distributed networks.

3.4.3 EigenTrust

One of the most compared and cited reputation models for distributed networks is EigenTrust [Kamvar *et al.*, 2003]. The algorithm affects to each agent a unique global trust value in a P2P file-sharing network, based on his history of uploads, achieving thus a decreasing in the number of downloads of inauthentic files.

The local trust value, s_{ij}, is defined in Eq. 3.8.

$$s_{ij} = sat(i,j) - unsat(i,j) \tag{3.8}$$

Where $sat(i,j)$ is the number of satisfactory transactions agent i has had with agent j. Equally, $unsat(i,j)$ is the number of unsatisfactory transactions.

A probability distribution p (with $p_i \in [0,1]$) is defined over pre-trusted agents. For instance, if some set of agents P are previously known to be trusted, then $p_i = \frac{1}{|P|}$ if $i \in P$, and $p_i = 0$ otherwise. With a definition like this, a normalized local trust value $c_{ij} \in [0,1]$ can be defined as shown in Eq. 3.9:

$$c_{ij} = \begin{cases} \frac{max(s_{ij},0)}{\sum_j max(s_{ij},0)} & if \ \ \sum_j max(s_{ij},0) \neq 0 \\ p_j & otherwise \end{cases} \tag{3.9}$$

Accordingly, if an agent does not know or trust anybody, he will choose to trust the pre-trusted agents. The global reputation of agent i is defined with EigenTrust in terms of the local trust values affected by other agents to agent i,

weighted by the global reputation of the assigning agents. So the aggregation of normalized local trust values is computed as:

$$t_{ik} = \sum_{j} c_{ij} c_{jk} \tag{3.10}$$

t_{ik} is the trust that agent i gives to agent k based on asking his friends. Let C be defined as the matrix $[c_{ij}]$ and t_i as the vector containing the values t_{ik}, then $t_i = C^T c_i$.

The agent i may wish to ask his friends' friends in order to get a wider view. In such a situation we would have that ${t_i}^2 = (C^T)^2 c_i$. If he continues in this way (i.e., ${t_i}^n = (C^T)^n c_i$), he will achieve a complete view of the network after n iterations.

The trust vector t_i will converge to the same vector for every agent i, if n is large enough. In other words, it will converge to the left principal eigenvector of C. Namely, t is a global trust vector in this model whose elements, t_j, quantify how much trust the system as a whole places in agent j.

Finally, in order to avoid malicious collectives in P2P networks, the reputation value (global trust value) is re-defined as:

$$t^{(k+1)} = (1 - \alpha) C^T t^{(k)} + \alpha p \tag{3.11}$$

where α is some constant less than 1 and $t^{(0)} = p$.

The EigenTrust reputation management algorithm is among the most successful and famous reputation systems. However, it suffers from some drawbacks. A main disadvantage of this algorithm is its reliance on a set of pre-trusted agents which causes peers to center around them. As a consequence, other peers are ranked low despite they are honest, marginalizing their effect in the system [Kurdi, 2015]. Another drawback of Eigentrust linked to the used normalization. In fact, the normalized trust values do not distinguish between an agent with whom agent i did not interact and an agent with whom agent i has had poor experience. The c_{ij} values are relative, and there is no absolute interpretation. That is, if $c_{ij} = c_{ik}$, we know that the agent j has the same reputation as the agent k in the eyes of the agent i, but we don't know if both of them are very reputable, or if both of them are mediocre.

3.4.4 SemanticWeb

A trust and reputation model specific for social networks is presented in [Zhang *et al.*, 2006]. The trustworthiness between two users is computed by searching all the paths that connect themselves; then, for each path the ratings associated with each edge are multiplied; finally, all the scores are added (normalizing that aggregation).

Let n be the number of paths from agent A to agent B. D_i denotes the number of users between A and B on the i^{th} path. The set of B's friends or neighbours is called M, m_i denotes B's direct friend or neighbour on the i^{th} path. w_i denotes the weight of the i^{th} path. The weight of each path is computed as follows (giving a higher weight to shorter paths):

$$w_i = \frac{\frac{1}{D_i}}{\sum_{i=1}^{N} \frac{1}{D_i}}; \tag{3.12}$$

The reputation of B from A's point of view is computed as follows:

$$R_{A \to B} = \sum_{i=1}^{N} T_{m_i \to B} \times \prod_{i \to j} R_{i \to j} \times w_i; \tag{3.13}$$

Where the reliable factor $R_{i \to j}$ denotes to which degree i believes directly in j's opinions or behaviours.

In this work, the authors did not compute a global value reflecting the reputation of one user in the whole network, their model only computes the reputation of each user based on the opinion of other user. We can consider these computed scores, simply, as indirect trust between two users. In addition, these computed values are essentially based on a direct trust ($R_{i \to j}$). However, authors did not show or mention how to calculate them nowhere.

3.4.5 TAUCA

TAUCA (Temporal And User Correlation Analysis) is an anomaly detection scheme composed and assessed for securing feedback based online reputation frameworks introduced in [Liu et Sun, 2010]. TAUCA is not a reputation computation model itself but it helps reputation systems to identify malicious users who try to manipulate the systems by submitting false reviews and to recover reputation scores. TAUCA first, uses a change detector to detect suspicious time intervals in which attack may be present. Then, it identifies the suspicious group of users by computing Euclidean distance of ratings given by each couple of users.

Finally, TAUCA removes ratings from malicious nodes from the system and reduces the bias in the recovered reputation scores.

3.4.6 PatrolF

The authors in [Tajeddine *et al.*, 2011] introduce PATROL-F (comPrehensive reputAtion-based TRust mOdeL with Fuzzy subsystems) as a comprehensive model for reputation-based trust incorporating fuzzy subsystems to protect interactions between agents in distributed systems. This reputation model incorporates many important concepts in order to compute an agent reputation in distributed systems, e.g, direct experiences and reputation values, the agent credibility, the decay of information with time based on a decay factor, first impressions and an agent system hierarchy.

PATROL-F is based on three fuzzy subsystems. The first one is used to set the importance factor of an interaction and related decisions. Decide and choosing which data is necessary or indispensable, or which data is needed more quickly, is a concept close to humans that fuzzy logic can model. Moreover, there is the region of uncertainty where an agent is not sure whether to trust or not (when the reputation of an other agent is greater than the absolute mistrust level ϕ, but less than the absolute trust level θ). It is in this region where the fuzzy techniques are effectively applied.

Finally, for the result of interaction (RI) value, fuzzy logic can be used to capture the subjective and humanistic concept of "good" or "better" and "bad" or "worse" interaction. RI becomes the result of several concepts effectively combined to achieve a more representative value. The decay factor is computed based on the difference of an agent's values of RIs between successive interactions.

3.4.7 FR Trust

A fuzzy reputation model for trust management in a semantic P2P Grid is proposed in [Javanmardi *et al.*, 2015]. Authors use fuzzy theory, in a trust overlay network named FR TRUST that models the network structure and the storage of reputation information. In fact, they present a reputation collection and computation system for semantic P2P Grids. The system uses the fuzzy set theory to compute a peer trust level, which can be either: Low, Medium, or High.

The model is specifically targeted for semantic P2P environments where peers

are clustered based on their similarities semantically identified. On a P2P grid system, there are virtual groups to which peers belong and each virtual group is represented by a special peer called *group coordinator*. After having some interactions with a peer in question in a virtual organization, peers report their own evaluations about the peer to the group coordinator which decides whether the peer is benevolent or malicious based on a threshold. The model is centralized in a sense that there is some authorities who collect reputation scores, but it is also distributed in a way that each node act as judges for the peer and report individual scores to the super node. There is also a special agent called *trust agent* who is responsible for storing reputation scores and computing global reputation value for nodes.

A main drawback of this model is its reliance to explicit topological restrictions. Indeed, FR TRUST can only be used in a confined architecture where group coordinators and trust agents exist.

3.4.8 REMSA

Authors in [Lee et Oh, 2015] introduced a new model named REMSA for reputation computation in OSNs. The proposed model considers the information associated to users to model how reputation is spread within the social network. In REMSA, each user updates reputations values affected to his neighbours based on the history of interactions and by considering the frequency of interactions in recent history. In addition, ReMSA, uses a voting mechanism to aggregate neighbours' opinions when updating reputation values. The voting process is recursive and aims to reach to every user in the network.

3.5 Linking Trust and Reputation with the Semantic Web

One of the core goals of the Semantic Web is to store data in distributed locations and use ontologies and reasoning to aggregate it. In addition, it offers a promising solution to publish information and services on the World Wide Web augmented with descriptions in a processable form understandable by both agents and machines. This will help Web agents to perform a variety of tasks on behalf of their users, such as information discovery and integration. In this thesis, we

are interested in trust and reputation management tasks in OSNs.

Many works, discussed in Sections 3.2 and 3.3, were developed proposing algorithms to help users to evaluate other ones as friends ([Hamdi *et al.*, 2012a]) or strangers ([Hamdi *et al.*, 2013], [Jiang et Wang, 2011], [Richters et Peixoto, 2010], [Lesani et Montazeri, 2009]). However, it is still unpractical to run these algorithms of trust evaluation each time a trust requester asks about the reliability of another user. It would be much easier if the requester gets a direct access to trust information. Thus information concerning trustworthiness should constitute a common knowledge base that is not only helpful for each user to know those whom he can trust, but also it encourages users to have benevolent behaviours. In addition, having this information publicly available is very useful for the Semantic Web or trust and security on OSNs researchers to evaluate their contributions. In the following, we provide a comprehensive literature review of how the Semantic Web is exploited to store trust information between the users of OSNs.

3.5.1 Trust module for defining trust relationships in FOAF

Vitiello, in [Vitiello, 2002], proposed a module for defining trust relationships in FOAF and allowing a user to describe the trust between one user and another one. To determine the trust relationship between individuals in FOAF circles, the author proposes various properties to use:

- trustsNever ($-\infty$)
- trustsNone (0)
- trustsMinimally (1)
- trustsAveragely (2)
- trustsHighly (3)
- trustsImplicitly ($+\infty$)

These values of trust can be interpreted as the probability that the trustee acts as expected. The *Never* description can then be interpreted as distrust: the trustee will most likely not act as expected. The value *None* can be interpreted as a sort of unknown trust level: the truster does not know whether (s)he can trust the trustee. The last *Implicitly* description appears meaning the total trust.

The vocabulary specifies nothing about trust transitivity or reputation, it is therefore only suited to specify direct Trust.

Although Vitiello aims at defining trust relationships between individuals, he asserts that he is only interested on the description of the RDF schema and not in determining the algorithm used in calculating the extended distant levels of trust. Moreover, the imposed numerical values cited above ($-\infty$, 0, 1, 2, 3, $+\infty$) should be mapped since the existing trust algorithms do not output these values. However, most of algorithms may require trust limits that cannot be mapped on six levels.

3.5.2 The Trust ontology

Golbeck in [Golbeck, 2005] introduces an RDF schema, designed to extend foaf:Person, which allows users to indicate a level of trust for people they know. The defined trust schema adds new properties with a domain of foaf:Person. Each of these properties specifies one level of trust on a scale of 1-9. The levels roughly correspond to the following:

1. Distrusts absolutely
2. Distrusts highly
3. Distrusts moderately
4. Distrusts slightly
5. Trusts neutrally
6. Trusts slightly
7. Trusts moderately
8. Trusts highly
9. Trusts absolutely

Using the trust ontology, the different trust ratings (i.e. "distrusts Absolutely", "trusts Moderately", etc.) are properties of the "Person" class, with a range of another "Person". In addition, users can specify trust levels for a person manually. Although this manual method keeps the subjective property of trust, it suffers from many problems already enumerated in Subsection 3.2.4.

With the trust ontology, the trust is considered transitive. In fact, Golbeck and Hendler in [Golbeck et Hendler, 2006] have developed an algorithm for comput-

ing trust transitivity. However, the trust ontology does not distinguish between referral trust (direct) and functional trust (transitive) and treats them as equal.

In this thesis, we propose to use normalized trust levels, since most of trust algorithms in OSNs generate values standing within the range [0, 1]. In addition, we add trust levels automatically while keeping the subjectivity of users. Moreover, we do not settle for enriching FOAF files with trust between people, but we define a new property presenting the reputation of each person.

3.6 Conclusion

This chapter, provided an overview of the research in trust and reputation management in OSNs. We have reviewed and analysed the advantages and disadvantages of the existing works of different aspects of trust management, including direct trust, inferred trust, reputation and linking trust with the semantic Web. From the result of this review, we can surmise that the maturity of a generally trust mechanism is yet to be realised. In fact, to address some of the current limitations, we introduce, in next chapters, novel approaches to compute trust levels between users of OSNs having direct and indirect relationships as well as their reputations.

Chapter 4

Social Trust-Oriented Extraction of Users' Interests

4.1 Introduction

In Chapter 3, we have introduced the existing methods computing the trust information between two participants in OSNs. However, OSNs contain complex important social information, including social relationships and interests which has not been included in most existing trust management methodologies in social networks.

This chapter presents a complex trust-oriented users' contexts extraction, where the complex social contextual information is taken into account in OSNs modelling, better reflecting the OSNs in reality. In fact, social softwares and on-line communities, such as social networking, folksonomies and blogs, are fostering a steady increase in user participation, engaging users and encouraging them to share more and more information, resources and opinions. The huge amount of information resulting from this emerging phenomenon gives us rise to investigate, understand, and exploit the knowledge about the user interests, preferences and needs.

4.2 The Influence of Social Context on Trusted Social Connections

In Computer Science, based on the statistics on Flickr, an online photo sharing social network [Mislove *et al.*, 2007], any two users in photo sharing usually have similar preferences. This example illustrates that the preference and interests of two users have influence on their social interactions in different domains (e.g., in photo sharing), and thus can badly affect social connections and trust between users. This feature also has been validated by Social Psychology theory ([D.J., 2009], [Gimpel *et al.*, 2008], [Luhmann, 1979] and [Zajonc, 2001]). Next, we give an example to illustrate the influences of social context on social connections.

As depicted in Fig. 4.1, A is a teacher in the Department of Computer Science of Faculty of Sciences in Tunis (Tunisia), and (s)he has social interactions with B as B is a popular student of A. D is a tennis coach and lives in the same city than both A and B. B does not have direct interactions with D. In the OSN, suppose

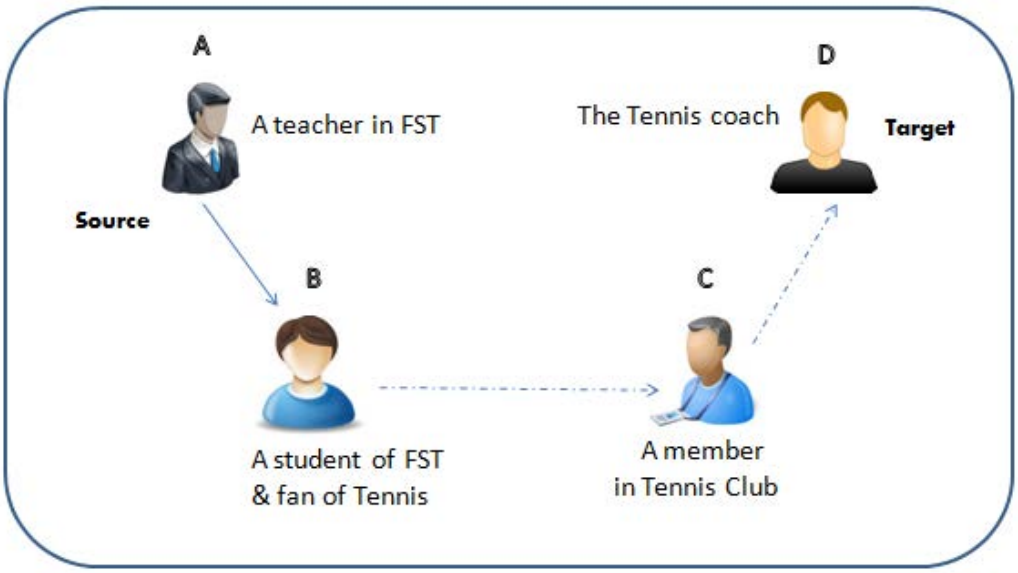


Figure 4.1: The influence of social preferences on trusted social connections

that A is looking for a tennis coach, (i.e., A is the source and D is the target), we can see that both B and D have a high probability to have a social connection as both of them like playing Tennis and thus they may be connected via some other members in a Tennis club (C).

From this example, we can see that the similarity of social interests between two unknown users can affect the social interactions and thus affect their trusted social connections. Based on this property, we can have a social connection based on the social context similarity between users.

4.3 Bridging Social Resource Sharing Systems and Folksonomies

These last years, the social tagging within Web 2.0 is known as the main means of large-scale data classification. Indeed, it allows the Internet users to store, to share and to look for their favorite links. Within a social tagging system, users register resources or their URLs found of interest. These lists of tagged resources are then open to the public or to a particular network. Indeed, within the last years, social software on the Web, such as Flickr, Delicious, Youtube, has received a tremendous impact with regard to hundred of millions of users. Thanks to these

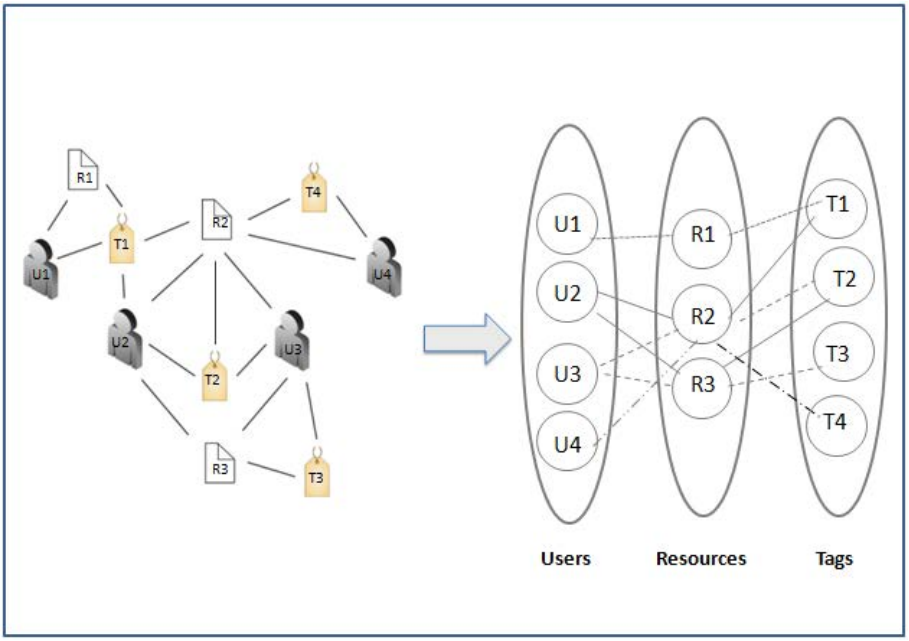


Figure 4.2: Illustration of the folksonomy's structure

sites, we can reach a multitude of bookmarks which are generally associated to freely chosen tags. The collective nature of these sites makes it possible to view bookmarks added by other users, and to produce a user-generated classification called *folksonomy*.

The word *folksonomy* is a blend of the words *taxonomy* and *folk* and stands for conceptual structures [Jaschke *et al.*, 2008]. It was created by the information architect T. Vander Wal. A folksonomy also, known as *social tagging* or *social indexing*, is a decentralized, spontaneous and collaborative system of classification, based on an indexation made by non-specialists. It is defined as the practice to share links between the various users of Internet. The Internet users can use their wished mode of classification by tags. It is a complete directory including bookmarks which allows any Internet user for sharing of favourites with other users. By building up his/her own file, the user can then register, on the network, the information that s/he wants to reveal, including his favourite links, some descriptions, notes and tags. Fig. 4.2 shows the structure of a folksonomy.

The wealth of folksonomies can be of use to classify users according to their domain of interest and enrich the knowledge bases. In this chapter, the classification of users in groups of interests is based on the shared resources as well as the way the users tag these resources.

4.4 Related Issues in Users' Interests Extraction

The classification of users from their domain interests relies on the recognition of their published tags' meaning. The idea consists on associating semantic DBpedia entities to these tags [Hamdi *et al.*, 2011]. Hence, for helping the tags' meaning retrieval, the DBpedia ontology needs to be dynamically enriched with shared conceptualizations from folksonomies. That's why we are also interested, in this thesis, in enriching the DBpedia ontology from the folksonomy tags and resources by adding new concepts, relations or instances. Thus, we need to study different related works in three research areas: *(i)* Associating semantics to tags (Section 4.4.1); *(ii)* Ontologies enrichment (Section 4.4.2); and *(iii)* Semantic representation of users Interests (Subsection 4.4.3).

4.4.1 Associating Semantics to Shared Conceptualizations in Folksonomies

The spread of tagging and the derivation of folksonomies is providing valuable data sources and environments for studying various user-related issues, such as online behaviour, tagging patterns, incentives for sharing, social networking, and opinion formation [Szomszor *et al.*, 2008a]. However, since tagging can be with any terms users wish to use, some works, like in [Guy et Tonkin, 2006], suggest that users should be educated about how to better tag, and that systems should implement procedures to check for problematic tags and suggest alternatives. Other approaches are more interested in improving tag quality and identifying their semantics.

We identify three groups of approaches according to if they are based on *(i)* ontologies; *(ii)* clustering techniques; or *(iii)* on a hybrid approach mixing clustering techniques and ontologies. In general, clustering-based approaches goal is to group related tags in the hope that such grouping will indirectly expose a meaning for their tags. Hence, these approaches do not formally define the meaning of tags or their relations. Ontology-based approaches aim at stating the meaning of the tags and their relations by means of associating semantic entities to tags. Hybrid approaches objective can be either: 1) to group tags using semantic information; or 2) to associate semantic entities to tags using as context

Criterion / Work	Approach type	Dataset	Semantic meaning recognition	Disambiguation
Mika, 2007 [Mika, 2007]	Clustering techniques	Delicious	Yes	No
Angeletou et al., 2008 [Angeletou *et al.*, 2009]	Ontology	Flickr	Yes	Yes
Szomszor et al., 2008 [Szomszor *et al.*, 2008b]	Ontology	Flickr + Delicious	Yes	No
Giannakidou et al., 2008 [Giannakidou *et al.*, 2008]	Hybrid	Flickr	Yes	No

Table 4.1: Comparison of surveyed approaches for associating semantics to tags

groups of tags [Garcia-Silva *et al.*, 2011].

Tag co-occurrence is a well known measure of tag relatedness that can be measured when two tags are used to annotate the same resource regardless of the annotator, or when two tags are used by the same user regardless of the resource. Authors in [Mika, 2007] exploit these tag concepts' co-occurence measures to find groups of tags. More exactly, an algorithm of clustering of graph was applied in order to group all semantically connected tags with specific terms. However, this approach does not explicitly deal with disambiguation problems; In fact, it is not clear in this approach how ambiguous tags can affect or be reflected in the generated ontologies.

Authors in [Angeletou *et al.*, 2009] associate semantic entities from online ontologies to tags as a way to formally define their meaning. They use the semantic search engine Watson [1] to relate the expanded set of tags to ontological entities. For each tag, several ontological entities may be retrieved and integrated in order to group similar ontological entities. Tags and their contexts are taken as input to the disambiguation activity. In this activity, if a tag has more than one sense in WordNet, then the hierarchy of its senses as extracted from WordNet is used to calculate the similarity with the senses of all tags in the tag set and thus disambiguating them. The authors mention that some tags and their context were not found in the WordNet hierarchy of senses, and thus the disambiguation activity failed [Garcia-Silva *et al.*, 2011].

[1] http://watson.kmi.open.ac.uk/

Another ontology-based approach, presented in [Szomszor *et al.*, 2008b], tries to associate folksonomy tags with domain ontology concepts using Wikipedia [2] categories as an intermediate shared representation between tags and ontology classes. According to the authors of this approach, the advantage of using Wikipedia as a shared representation for tags is that Wikipedia is maintained collaboratively by a large user community. Thus, Wikipedia incorporates a new terminology faster than linguistic resources like WordNet. However, this approach fails when the Wikipedia's page is not directly related with the intended meaning of the tag according to its context, mainly because the approach lacks a disambiguation process. The disambiguation activity is not specified in this approach, although Wikipedia disambiguation pages are pointed out as a possible source of information to disambiguate tags [Garcia-Silva *et al.*, 2011].

Authors in [Giannakidou *et al.*, 2008] present some approaches relying on ontologies and clustering techniques whose goal is to group related tags. They proposed a statistical approach for discovering the semantics of tags by clustering tags and resources, being resources represented by their annotations, but this approach does not explicitly deal with disambiguation problems.

Table 4.1 resumes the surveyed approaches using the following criteria. The approach type criterion describes whether the approach is based on ontologies, on clustering techniques or if it is an hybrid approach. The dataset criteria is used to present the folksonomies used to test the approaches. Semantic meaning recognition criteria is used to verify if the approach finds related tags groups identifying their meaning. The last criteria, disambiguation, checks whether authors used a method to disambiguate tags or not.

In this Chapter, we introduce a new hybrid approach that groups resource's tags and enrich their meanings by associating them with relevant concepts in an online ontology. Indeed this approach aims to recognize the context of the group of tags associated to resources. We heavily rely on the existing multidomain ontology DBpedia to link tags to its concepts. To do that, we follow a pre-processing step for correcting tags and disambiguating their senses. Our methodology, that mixes tag co-occurrence with semantic similarity to recognize tags' contexts, has as goal to *(i)* cluster the folksonomy users according to their interests; and *(ii)* enrich DBpedia ontology with the folksonomy efficient informations.

[2] http://en.wikipedia.org/

Criterion / Work	Data source	Target users	Cleaning identification	Disambiguation
Mori et al., 2004 [Mori *et al.*, 2004]	Informations in the Web	Web users	No	No
Diederich and Iofciu, 2006 [Diederich et Iofciu, 2006]	DBLP keywords	People in research community	No	No
Demartini, 2007 [Demartini, 2007]	Edits' history of users	Wikipedia experts	No	Yes
Szomszor et al., 2008 [García-Silva *et al.*, 2011]	Folksonomy tags	Folksonomy users	Yes	Yes

Table 4.2: Comparison of surveyed systems for discovering users' interest

4.4.2 Discovering the Users' Interests

This Chapter is mainly concerned with learning about interests of Web users since, as described in Section 4.2, interests play a fundamental role on defining trusted social connections. A number of studies have focused on modelling user interests based on the folksonomy activity. In [Mori *et al.*, 2004], Mori *et al.* investigated extracting information from Web pages using term co-occurrence analysis to build FOAF files. Demartini, in [Demartini, 2007], suggested using the history of users' edits in Wikipedia to find out about their expertise. Such an approach will obviously only work for users whom actively edited Wikipedia pages. In contrast, Szomszor *et al.*, in the work reported in [Szomszor *et al.*, 2008b], exploited the resources of Wikipedia, but they were interested on identifying and semantically representing the general interests of users, based on what they tag and how they tag across the folksonomies. In [Diederich et Iofciu, 2006], Diederich and Iofciu proposed to use the tags associated with the specified resources to identify persons inside a community with similar interests. The idea was to identify user interests based on tag clustering. The proposed work in [García-Silva *et al.*, 2011] aimed to supply an architecture that constructs a model of user interests. To do that, García-Silva *et al.* proposed an approach that examines users' interaction with folksonomies.

This overview of existing works is summarized in Table 4.2. In this table, the first column contains the reviewed approaches. The following columns are the

criteria that we followed to compare these approaches. The data source column shows the dataset used to extract users' interests. The target users are those concerned by extracting their interests. The data cleaning column indicates if the approach includes filters that take into account tag use frequency, lexical characteristics, morphological characteristics, or even the language of the tags. The last criterion checks whether authors used a method to disambiguate tags or words taken to test their approaches.

Starting with the data source used to extract users' interest, some of the mentioned related works ([Diederich et Iofciu, 2006] and [Demartini, 2007]) used a specific dataset that engenders a limited extraction of communities while others used more general informations ([Mori *et al.*, 2004] and [García-Silva *et al.*, 2011]). Regarding the target users, some works target users with multiple interests ([Mori *et al.*, 2004] and [García-Silva *et al.*, 2011]), while others concentrate only on a specific domain namely Wikipedia expertise ([Demartini, 2007]) or research community ([Diederich et Iofciu, 2006]). Only the approach proposed, in [García-Silva *et al.*, 2011], tackles the problem of word cleaning by proposing a tag filtering process. It is worth of mention, none of [Mori *et al.*, 2004] and [Diederich et Iofciu, 2006] solve the problem of words ambiguity. The author in [Demartini, 2007] uses the external semantic resource, WordNet, to disambiguate tags, while authors in [García-Silva *et al.*, 2011] used both of WordNet and Google "did you mean mechanism" for that purpose.

It is worth of mention that our work has common points with the work presented in [García-Silva *et al.*, 2011]. Indeed, we use the folksonomy tags as data source to extract its users' interests. In the same spirit of our tag cleaning, García-Silva *et al.* in [García-Silva *et al.*, 2011], proposed an approach of cleaning existing tags using a filtering process. However, if a synonymy between a pair of tags is found, then the pair is directly replaced by the synonym term without further checking, whereas one of the tags pair may have other synonyms. In our work, we propose a pre-processing step as a cleaning method that uses WordNet to extract several tags' synonyms. Besides, we disambiguate tags and recognize their semantic contexts using the DBpedia ontology with regard to tag frequency, that is is measured based on the number of times the tag has been used to annotate a resource. Another novelty of our work is that we use user-tagging activity track not only to extract users' interests, but also to cluster these users according to their shared resources and their behaviour of tagging. For that, a user semantic

Criterion / Work	Enriched Elements	Initial Requirements	Redundancy Elimination	Disambiguation
Faatz and Steinmetz, 2004 [Faatz et Steinmetz, 2004a]	Concepts with is_a relation	Google search engine, Web-based corpus, WordNet	Yes (through domain expert)	No
Navigli and Velardi, 2006 [Navigli et Velardi, 2006]	Concepts, Instances	Domain ontology, Wordnet, Structured Corpus	No	No
Passant, 2007 [Passant, 2007]	Concepts, has_tag relation	Domain ontology, folksonomy, vocabulary of SIOC ontology	Yes (through domain expert)	Yes
García-Silva et al., 2011 [García-Silva *et al.*, 2011]	Instances (only images)	folksonomy, Search Engine, DBpedia ontology	No	Yes

Table 4.3: Comparison of surveyed systems for ontology enrichment

clustering algorithm is proposed (Subsection 4.6.1).

4.4.3 Semantic Web Improvements : DBpedia Enrichment

Ontologies provide semantics, which can be used by a broad range of applications such as search and retrieval, semantically enhanced Web services and software agent communications [Faatz et Steinmetz, 2004b]. There exists several accepted workflows for ontology engineering, e.g, collaborative processes of domain experts. All state-of-the-art definitions of ontology engineering work-flows essentially are open to so-called ontology enrichment techniques. These techniques are semi-automatic processes, which generate extensions of the evolving ontology and propose these extensions to the ontology engineers. In general, the extensions may include: new concepts to be integrated, new relations to be instantiated between existing concepts, corrections of existing concepts and relations, formal explanations of how to map and merge different ontologies [Faatz et Steinmetz, 2004b].

Several methods for ontology enrichment are described in the literature. Navigli and Velardi, in [Navigli et Velardi, 2006], provided a pattern-based methodology to automatically enrich a core ontology with the definitions of a domain glossary. Applied to the domain of cultural heritage, the system populates the CIDOC CRM core ontology [Doerr *et al.*, 2007], using terms extracted from glosses con-

tained in the Art and Architecture Thesaurus (AAT). During this process, manually developed extraction patterns were also of use. The system starts by performing part of speech tagging and then named-entity recognition by applying manually created regular expressions to locate terms that can be annotated with concepts from the CIDOC CRM core ontology. Then, the system tries to locate the domain property and the range property of each annotated text portion in order to populate the ontology, with the help of manually developed constraining rules.

In [Faatz et Steinmetz, 2004a], Faatz and Steinmetz proposed an ontology enrichment with texts from the WWW. The enrichment process is based on the comparison between statistical information of word usage in a large text collection, a so called *text corpus*, and the structure of the ontology itself. The text corpus is constructed by using the vocabulary from the ontology and querying the WWW via Google.

The relation between social network and the emergence of ontologies in collaborative tagging is largely discussed in literature [Limpens *et al.*, 2009]. Gruber in [Gruber, 2005] introduced the idea of an "ontology of folksonomy", where each tagging event has related attributes (the tag, the object designed by the tag, the tagger, the concerned resource, etc.).

Passant, in [Passant, 2007], proposed an approach to enrich ontologies from folksonomies. Indeed, he associated to each tag a property attached to concepts in the ontology of an enterprise. This approach disambiguates tags by associating them to concepts from existing ontologies. However, he considered neither grammar misspellings nor flaws made by the users tagging. Furthermore, he did not propose a method to correct tags. Besides, the author only paid attention to the addition of new properties but the system was unable to enrich the ontology with new instances. It is also worth of mention that García-Silva et al., in the work of [García-Silva *et al.*, 2011], addressed the problem of how to enrich DBpedia ontology instances with candidate images retrieved from existing Web search engines. They achieve that task by expanding the semantic neighbourhood of DBpedia resources with context words, i.e., words that occurred around DBpedia resources mentioned in Wikipedia text and computing semantic relatedness between tagging information and DBpedia resources.

At a glance, Table 4.3 sketches the surveyed approaches. The second column

contains the elements (concepts, instances) with which the ontology is enriched, the third numerates the prior knowledge and external resources used for enriching an ontology, the fourth one checks whether authors eliminate redundancy or not and the last column verifies if they used a method to disambiguate words or not.

Starting with the elements enriched in the surveyed works, some works are more complete in the sense that they can enrich an ontology with both concepts and instances ([Navigli et Velardi, 2006]), while others concentrate only on adding one of them, namely instances ([García-Silva *et al.*, 2011]) or concepts ([Passant, 2007]). Considering the initial requirements, some of the mentioned related works do not try to learn synonyms ([Passant, 2007] and [García-Silva *et al.*, 2011]), while the other ones include an extraction engine of synonyms by using available resources. None of [Navigli et Velardi, 2006] and [Faatz et Steinmetz, 2004a] tackle the problem of words' ambiguity. Nevertheless, it is true that [Passant, 2007] disambiguates tags, but without considering the mistakes committed because of the free tagging. Both of [Passant, 2007] and [García-Silva *et al.*, 2011] respective approaches refer to existing ontologies to disambiguate tags. Regarding the redundancy elimination, it is checked through domain experts in some works ([Faatz et Steinmetz, 2004a] and [Passant, 2007]). This problem is not mentioned in the other presented works.

In this thesis, we propose a new approach that aims to enrich the DBpedia multi-domain ontology from existing folksonomies and thus to offer a social interoperability of the data resulting from heterogeneous sources ([Hamdi *et al.*, 2012c], [Hamdi *et al.*, 2012b]). Our enrichment process is able to add both concepts and instances. We adapt a synonyms extraction engine by means of the lexical database WordNet to associate synonyms to the folksonomy tags. Furthermore, we use Google Translator and Google "Try with this Spelling" mechanism to fit grammar misspellings and mistakes made by the folksonomy users and disambiguate the sense of tags. Moreover, our method tries to reduce the role of experts. Indeed, our enrichment algorithm automatically checks the concept redundancy and enriches DBpedia whenever a new tag or resource does not match any existing concept.

4.5 Key Notions

In this section, we briefly sketch the key notions that will be of use in the remainder of this chapter. In the following, we start by presenting a formal definition of an *ontology*.

Definition 1 :(ONTOLOGY)[Trabelsi *et al.*, 2010]

An *ontology* is a 4-tuple $\mathcal{O} = (\mathcal{C}, \mathcal{P}, \mathcal{TR}, \mathcal{NTR})$, where the disjoint sets $\mathcal{C}$ and $\mathcal{P}$ contain concept and relation identifiers, respectively. $\mathcal{TR}$ defines taxonomic (vertical) relationships between concepts, i.e., $\mathcal{TR} \subseteq \mathcal{C} \times \mathcal{C}$, and $\mathcal{NTR}$ defines non-taxonomic (horizontal) relationships between concepts, i.e., $\mathcal{NTR} \subseteq \mathcal{C} \times \mathcal{P} \times \mathcal{C}$.

Definition 2:(FOLKSONOMY)[Jaschke *et al.*, 2008]

A *folksonomy* is a set of tuples $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$, where $\mathcal{Y} \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{R}$ is a triadic relation such that each $y \subseteq \mathcal{Y}$ can be represented by a triple: $y = \{(u, t, r) \mid u \in \mathcal{U}, t \in \mathcal{T}, r \in \mathcal{R}\}$, denoting that the user u annotated the resource r with the tag t.

Example: Figure 4.2 shows a folksonomy's example $\mathcal{F}$ with $\mathcal{U} = \{u_1, u_2, u_3, u_4\}$, $\mathcal{R} = \{r_1, r_2, r_3\}$ and $\mathcal{T} = \{t_1, t_2, t_3, t_4\}$. For instance, user u_2 annotates resource r_2 with tag t_1.

Definition 3 :(TAGS' CONTEXT)

A *Tags' Context* $\mathcal{CO}$ is a sub-ontology, $\mathcal{CO} = (\mathcal{C}', \mathcal{P}', \mathcal{TR}, \mathcal{NTR})$, that checks the mapping $\sigma : \mathcal{R} \times \mathcal{T} \rightarrow \mathcal{C}' \cup \mathcal{P}'$ where each resource R_i is annotated with a set of tags T_i compatible with its true sense and each tag $t_j \in T_i$ matches the concept $c'_j \in \mathcal{C}'$ or the relation $p'_j \in \mathcal{P}'$.

Definition 4:(ENRICHED FOLKSONOMY)

An *enriched folksonomy* is a set of tuples $\mathcal{EF} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{CO}, \mathcal{Y})$, where $\mathcal{Y} \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{R} \times \mathcal{CO}$ is a quadratic relation such as each $y \subseteq \mathcal{Y}$ can be represented by a quadruple: $y = \{(u, t, r, co) \mid u \in \mathcal{U}, t \in \mathcal{T}, r \in \mathcal{R}, co \in \mathcal{CO}\}$ denoting that the user u annotated the resource r granted to the context co with the tag t.

In the following, we recall the main definitions ([Jaschke *et al.*, 2008]) related to triadic concepts and contextualized triadic concepts, that exactly mimic the structure of a *folksonomy* and an *enriched folksonomy*.

Definition 5:(TRIADIC CONCEPT)[Trabelsi *et al.*, 2011] A *triadic concept* (or a tri-concept for short) of a folksonomy $\mathcal{F} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$ is a triple $(\mathcal{U}', \mathcal{T}', \mathcal{R}')$, where $\mathcal{U}' \subseteq \mathcal{U}$, $\mathcal{T}' \subseteq \mathcal{T}$ and $\mathcal{R}' \subseteq \mathcal{R}$ with $\mathcal{U}' \times \mathcal{T}' \times \mathcal{R}' \subseteq \mathcal{Y}$ denoting that the

set of users $\mathcal{U}'$ annotated the set of resources $\mathcal{R}'$ with the set of tags $\mathcal{T}'$.

Definition 6:(INTEREST GROUP) An *Interest group* is a contextualized Tri-concept, i.e., a tri-concept associated to a context $\mathcal{CO}$. It is a tuple $\mathcal{CTC} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{CO}, \mathcal{Z})$, where $\mathcal{Z}$ is a quadratic relation and each $z \subseteq \mathcal{Z}$ can be represented as a quadruple $z = \{(u, t, r, co) \mid u \in \mathcal{U}, t \in \mathcal{T}, r \in \mathcal{R}, co \in \mathcal{CO}\}$, denoting that the user u annotated, with the tag t, the resource r according to the context co.

4.6 Global Process Phases

The first aim of the work described in the present Chapter is to supply an original process that builds a network of users grouped according to their interests by examining their interactions with a folksonomy site. Such a building process is concretely illustrated through the real-life folksonomy, collected from the social network Delicious. The process is split into two phases : the first and the third steps depicted in Figure 4.3 (tag's context recognition and semantic clustering of users, resp.).

The second aim of this work is to enrich the DBpedia ontology with shared conceptualizations from the folksonomy; this process is splet also into two phases that include the first and the second steps shown in Figure 4.3 (tag's context recognition and DBPedia ontology enrichment, resp.).

In what follows, we describe each of the three mentioned steps.

4.6.1 Tags' Context Recognition

Using free-tagging makes ambiguity handling a compulsory issue. Indeed, a tag can express various concepts and the system cannot make any difference. As illustrated by Figure 4.4, by tagging "Java", some users intend the Java island, while others mean the programming language Java. In this case of ambiguity, this tag can be associated to different concepts. To overcome this problem, we propose a tags' context recognition method minimizing the rate of ambiguity. This method is based on domain ontologies. Indeed, we consider that it is unavoidable to remove the tag's ambiguity raised when relying on a folksonomy as a knowledge representation. In what follows, we show that this method allows to determine the contexts of resources' tags in spite of the lack of information. For the previous example, undoubtedly, we should be able to determine that the

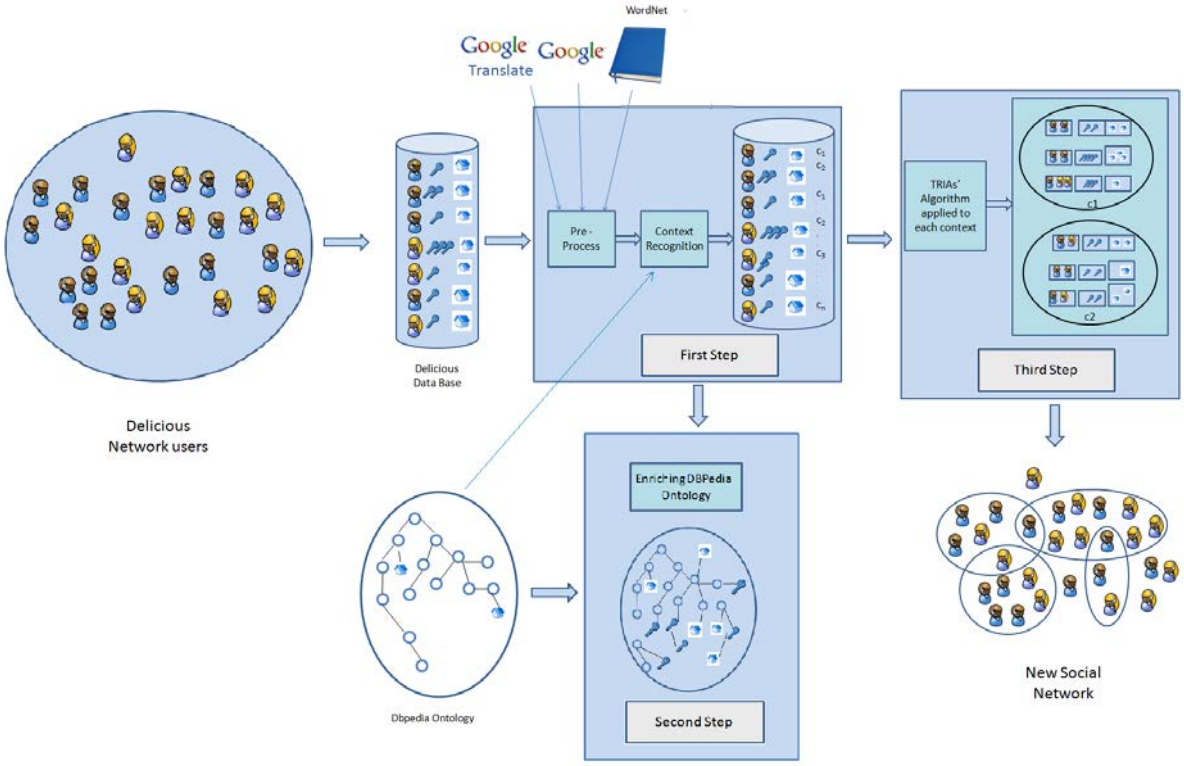


Figure 4.3: Global process steps of our approach

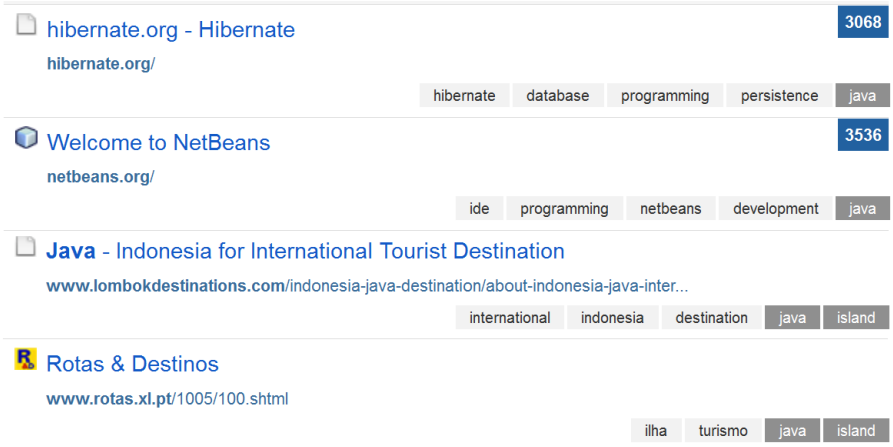


Figure 4.4: Example of tags' ambiguity on Delicious folksonomy

tag "Java" applied to the resource (http://hibernate.org/) involves Java as programming language. In the meanwhile, it means the Java Island for the resource (http://www.rotas.xl.pt/1005/100.shtml).

Pre-processing

As stated above, users can freely tag each of their bookmarks with chosen terms. Thus, they can make many grammatical, syntactic and other mistakes. To tackle this issue, we propose a pre-processing step which precedes the tags' context recognition step.

Worth of mention that we do not consider tags as objects as in [Gruber, 2005] or [NEWMAN, 2005], but as simple text strings mainly because we are not interested in inferring relationships at the tag level, but between tags and ontology concepts associated to these tags.

This pre-processing step considers the lexical database WordNet, Google *"Try with this spelling"* mechanism and Google Translator with Detect Language.

- ***Google Translate with Detect Language***: Google Translate is a free statistical translation machine tool, provided by Google, to translate a word, a section of text, a document or a Web page, into another language. Text in a foreign language can be typed, and if ***Detect Language*** is selected, then it will not only detect the language but also translates it into English by default.

- ***Wordnet*** is a system grouping nouns, verbs, adjectives and adverbs into sets of cognitive synonyms called *synsets*, each one expressesing a distinct concept.

Algorithm 3: TagsMeaningsEnrichment

Data: Set of Tags $\mathcal{TS}$ associated to a resource.

Result: Set of New Tags $\mathcal{NTS}$.

```
1  begin
2      NTS = ∅;
3      foreach ( t_i ∈ TS ) do
4          SS = ExtractSynonym ( t_i );
5          if SS ≠ ∅ then
6              NTS = NTS ∪ SS;
7          else
8              CT = CorrectSpelling (t_i);
9              if CT ≠ ∅ then
10                 foreach ( t_j ∈ CT ) do
11                     SS = ExtractSynonym (t_j);
12                     NTS = NTS ∪ SS;
13     NTS = NTS ∪ TS;
14     return NTS ;
```

Synsets are interlinked by means of conceptual-semantic and lexical relations.

- ***Google "Try with this spelling" mechanism*** returns more relevant terms than those already introduced manually and without control by users. Indeed, if a tag is not a standard tag or a jargon one, i.e., nonsense tag ("webmanagr"), then we consider possible misspellings ("webmanager") and/or compound nouns, ("web manager"). In fact, when a searched term is entered, the Google engine checks whether more relevant search results are found with an alternative spelling. Since Google's spell check is based on occurrences of all words in the Internet, it is able to suggest common spellings for proper nouns that would not appear in a standard dictionary.

In what follows, we introduce a new algorithm called TagsMeaningsEnrichment, that investigates all possible synonyms of a tag by using the external tools already quoted above. It takes as input the set of tags $\mathcal{TS}$ associated to each resource and returns a new set of tags $\mathcal{NTS}$ including all tags in $\mathcal{TS}$ and their synonyms. The pseudo-code of TagsMeaningsEnrichment is sketched by

Algorithm 4: ExtractSynonym

Data: A tag t.

Result: Set of tags $\mathcal{SS}$.

```
1 begin
2     te = TranslateToEnglish ( t );
3     SS = SearchSynonym ( te );
4     return SS ;
```

Algorithm 3.

In line 4, this algorithm invokes the ExtractSynonym function described in Algorithm 4. This function computes all synonyms $\mathcal{SS}$ for each tag t_i according to WordNet, after translating it to English thanks to Google Translate with Detect Language.

In the case of empty answer from WordNet, TagsMeaningsEnrichment checks the tag spelling with Google *"Try with this Spelling"* mechanism by invoking the CorrectSpelling procedure (line 8). This latter returns, if possible, the corrected $\mathcal{CT}$ tags set. In that case, TagsMeaningsEnrichment re-calls, in line 11, the ExtractSynonym procedure and re-looks for them in WordNet.

Let us consider the resource (*http://www.rotas.xl.pt/1005/100.shtml*) in Figure 4.4. It is tagged by the set of tags $\mathcal{TS}$ = {*java, island, turismo, ilha*}. Thus, by performing the TagsMeaningsEnrichment algorithm, the returned new set of tags will be $\mathcal{NTS}$ = { *java, island, turismo, ilha, coffee, island, programming language, land mass, zone, tourism, touristry, island, land mass, zone*}. Table 4.4 shows how the $\mathcal{NTS}$ set is obtained.

We remind that, within the tags' context recognition phase, we are looking for the context to which belong a resource and its tags. Thus, we are looking to associate the resource to a specific domain, i. e., a context. In this respect, after the description of the pre-processing step, we heavily rely on the existing ontology DBpedia [Heath et Bizer, 2011] to link tags to their domain concepts.

Linking Tags to DBpedia Concepts

Within this step, we aim at recognizing resource's contexts. The idea is to search for each term of the $\mathcal{NTS}$ set, returned by the pre-processing step, in the DBpedia ontology. The idea is already used in a previous work [Passant, 2007], in which

Function \ $\mathcal{TS}$	**java**	**island**	**turismo**	**ilha**
TranslateToEnglish	java	island	tourism	island
SearchSynonym	programming language, island, coffee	land mass, zone	touristry	land mass, zone
CorrectSpelling	-	-	-	-

Table 4.4: Example of the $\mathcal{NTS}$ Computation

authors put the focus on using some existing domain ontologies combined with the SIOC ontology for that purpose. However, we find this not practical for a general solution since it is too heavy to access, extract, interpret, and maintain available information from different ontologies. Furthermore, folksonomies are dynamic systems that steadily evolve to accommodate new terminology and trends, and information within this sites is extremely diverse and stands at the confluence of many domains. Therefore, we decided to use the DBpedia ontology since it offers a wide range covering topics and is constantly updated by the community. Indeed, this cross-domain ontology currently covers over 272 classes organized as a subsumption hierarchy and described by 1,300 different properties with domain and range definitions. DBpedia was manually created from the most commonly used infobox templates from the English edition of Wikipedia [Bizer *et al.*, 2009].

The hierarchy, and the specified parent-child or sibling relationships between concepts in DBpedia ontology, facilitate the specification of each tag's context. Indeed, for each tag ($t \in \mathcal{NTS}$), we attempt to identify its corresponding context as a set of concepts in the DBpedia ontology, each one being a sub-class, a property, or an instance. To do so, we need a toolkit able to retrieve these concepts and processing their semantic information from DBpedia. Since DBpedia's information are represented in the standard OWL language, any OWL compliant API can be used, for example, the Wonder Web OWL API [3] and the Jena ontology

[3] http://owlapi.sourceforge.net/

API [4]. We use Jena in this work since it allows, besides treating RDF, RDFS and OWL documents, supplying an inference engine providing such reasoning on the ontologies.

In order to find the most probable context for the initial $\mathcal{TS}$ set of tags, we introduce a measure of **dominance** (Eq. 4.1) to compute the score of each context found from the $\mathcal{NTS}$ set. The dominance highlights to which extent the context dominates the other ones. In this equation, $N_{\mathcal{CO}}$ denotes the number of tags from the $\mathcal{NTS}$ set corresponding to the context $\mathcal{CO}$.

$$dom(\mathcal{CO}) = \frac{N_{\mathcal{CO}}}{|NTS|} \tag{4.1}$$

The dominant concept, i. e., the one that presents the highest appearance frequency, will be considered as the Top Class of the context. In the previous example, the tag *island* dominates other concepts with a dominance value equal to 21%. So, the context of the corresponding resource tags in the DBpedia ontology is $Island$.

When the context corresponding to the dominant concept is found, an enrichment of the ontology can take place by adding the resource as an instance of the context. Besides, if a tag does not correspond to any element in the DBpedia ontology, an enrichment of the ontology with a new concept is possible.

4.6.2 The DBpedia Ontology Enrichment

Multimodal knowledge bases have been successfully used in the past for several knowledge consuming tasks. Enriching knowledge bases with diverse information makes it possible to complement and improve results of knowledge consuming tasks, including question and answering systems and recommendation processes among others. However, retrieving relevant resources and information from folksonomies in the Web to enrich a knowledge base is far from being a trivial task. In this work, we focus on enriching the popular ontology DBpedia. To do that, we propose an approach that determines the meanings of tags to recognize the suitable Tags' Context. Thus, we will be able to enrich the DBpedia ontology with relevant tags and resources. Figure 4.5 depicts the different steps and their corresponding substeps followed to achieve the DBpedia enrichment.

[4] http://jena.sourceforge.net/

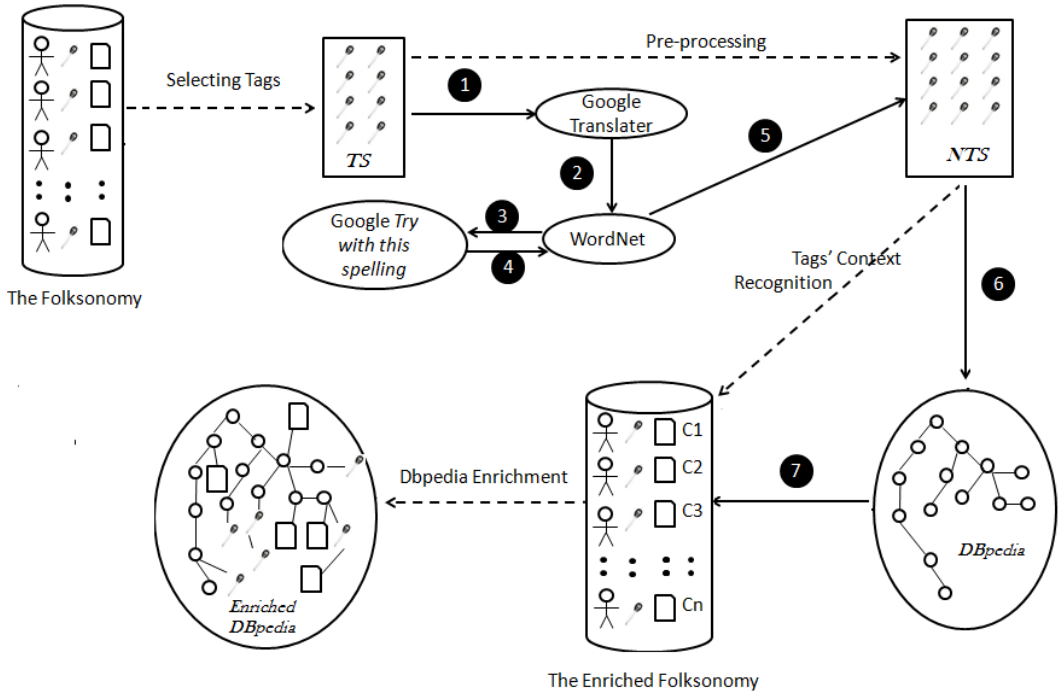


Figure 4.5: Illustration of our enrichment approach

Our DBpedia *enrichment algorithm* is a process that takes a given collection ξ as input, where $\xi = R \cup \mathcal{NTS}$: R is the set of all resources shared by the users of the folksonomy and $\mathcal{NTS}$ is the output of algorithm 3. It produces a set of propositions $P(c) \subseteq \xi$ such that for each $c \in \xi, \exists c_1 \in$ DBpedia such that $r(c, c_1)$, where the relation r underlies that there exists a semantic relationship between the enriched concept c and the existing DBpedia concept c_1.

Once new concepts are selected, we cross the DBpedia ontology and we check that the selected terms (C) do not already exist in the ontology. In fact, a basic SPARQL query will check that. For instance, the following query checks the existence of the concept *"programming language"*:

```
SELECT ?x WHERE {
?x rdfs:label "programming language"@en.  }
```

The different semantic relations that link concepts in DBpedia ontology can be seen either as vertical or horizontal. Our *enrichment algorithm* offers both possibilities to enrich DBpedia: with vertical or horizontal relations.

Enrichment with Horizontal Relations

We can classify the horizontal relations in different categories: *(i)* Similarity that gathers similar concepts with an associated similarity degree; *(ii)* Synonymy that links similar concepts with a similarity degree equal to 1, concepts are equivalent, and the denoting terms are synonyms; *(iii)* Generic Relatedness that links the other related concepts that may be of different kinds but must be defined in the Ontology; and *(iv)* Translation that gives to a concept having a source-language text an equivalent one with a target-language text.

In this work, our enrichment approach tries to minimize the expert role as much as possible. Thus, we are mainly interested in synonymy and translation relationships. Indeed, the enrichment with these two relations is automatic and does not need to be validated by a domain expert.

- Synonymy

In several dictionaries, synonymic are mixed with prolonged definitions (glosses) in a capricious way and it is not possible to distinguish them automatically [Faatz et Steinmetz, 2004a]. The terms announced as synonyms may sometimes not be actually the synonyms for the chosen term, but may represent more specific or general concepts. These dictionaries represent mere dictionaries not adhering to any particular linguistic model, even if they may represent valuable resources on their own. That's why we relied, during the pre-processing step, on the much stronger model proposed by Wordnet, which, being a structured lexical database, presents a clear distinction between words, senses and glosses, and is characterized by various semantic relations.

Our main concern here is to extend this lexical representation further by automatically deriving synonyms from WordNet. The idea is to use synonyms found in WordNet thanks to the ExtractSynonym procedure. In fact, we add each synonym of each synset that corresponds to the dominant context of the original concept. The RDFS relation *is-synonym-of* presents the ObjectProperty linking the equivalent concepts.

- Translation

The DBpedia ontology allows users to switch between 97 languages[5] to be used for labels and abstracts. This enriches DBpedia and satisfies users. In fact,

[5]http://en.wikipedia.org/wiki/DBpedia

it allows them to gain deeper insights about the different concepts. However, most of concepts have labels given in English. This is why we try to exploit the different languages used by taggers to enrich the DBpedia concepts. For example, the *Programming Language* concept[6] is related with `rdfs:label` property to the English and Portuguese names "programming language"@en and "linguagem de programação"@pt; thus, we also can enrich this concept by the French name "langage de programmation"@fr.

Enrichment with Vertical Relations

In order to retrieve relevant resources for DBpedia ontology, we propose to enrich it by a process of instantiation that takes advantage of resources (images, videos, Web pages, etc.) in existing folksonomies when they are available. In fact, we propose to associate *relevant* resources shared by users, to the DBpedia concepts in the adequate context using the `instance-of` property. In Section 4.8, we show through our evaluation how to distinguish between relevant and irrelevant resources.

4.6.3 Semantic Clustering of Users

Folksonomies do not explicitly state shared conceptualizations, nor do they force users to use the same tags. However, the usage of tags of users with similar interests tends to converge to a shared vocabulary. Our intention is not only to discover these shared conceptualizations that are hidden in a folksonomy like in [Jaschke *et al.*, 2008], but also to exploit the context's recognition step, described above, and gather users sharing resources accorded to a same context. Worth of mention, the proposed users' semantic-clustering algorithm is based on $\mathcal{TRIAS}$ algorithm presented in [Jaschke *et al.*, 2008]. The main difference is that $\mathcal{TRIAS}$ looks at mining all frequent *tri-concepts*, but in the proposed algorithm, we look for mining all frequent *contextualized tri-concepts*. Our algorithm for mining all frequent *contextualized tri-concepts* of the enriched folksonomy $\mathcal{EF} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{CO}, \mathcal{Y})$ is listed in Algorithm 5.

In line 3, the algorithm initializes the set of clusters $\mathcal{CL}$ by invoking the **GroupByContext** function. This latter groups quadruples (u, r, t, co) by the context co. Then, in lines 4 and 5, the algorithm invokes the TRIAS procedure to it-

[6] http://mappings.dbpedia.org/server/ontology/classes/ProgrammingLanguage

Algorithm 5: UserSemantiClustering

Data: $\mathcal{EF} = (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{CO}, \mathcal{Y})$

Result: Set of Contextualized Tri-Concepts $\mathcal{CTC}$

```
1 begin
2     CTC = ∅;
3     CL = GroupByContext (EF);
4     foreach ( cluster cl_i ∈ CL ) do
5         CTC = CTC ∪ TRIAS (U, T, R, Y);
6     return (CTC)
```

eratively extract the contextualized tri-concepts for each cluster of the enriched folksonomy $\mathcal{EF}$.

By extracting the contextualized tri-concepts, we have, not only investigated matching users based on the similarity of tag clusters in Delicious, but also identified the specific interest of Delicious users. All these users become members of a new, automatically constructed, social network. Each user is assigned to one or several groups if he shares resources accorded to a same context and he has the same behaviour of tagging those resources.

4.7 Illustrative Example

We present, in this section, an illustrative example to concretize our method. We make use of the real dataset collected from the folksonomy of the Delicous collaborative site.

Let us consider the simple folksonomy $\mathcal{F}_1$ showed in Table 4.6 where a set of users $\mathcal{U}_1$ are represented by a list of resources $\mathcal{R}_1$ annotated by a set of tags $\mathcal{T}_1$, with $\mathcal{U}_1 = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}\}$, $\mathcal{R}_1 = \{r_1, r_2, r_3, r_4\}$, $\mathcal{T}_1 = \{t_1, t_2, t_3, t_4, t_5\}$ and each "×" denotes a triadic relation between a user from $\mathcal{U}_1$, a resource from $\mathcal{R}_1$ and a tag from $\mathcal{T}_1$.

The URLs and labels corresponding, respectively, to resources and tags of $\mathcal{F}_1$ are described in Table 4.5.

We first try to apply the pre-processing step to recognize the tags' context of each resource. We take the example of the popular resource shared by more than 7430 Delicious users, http ://www.eclipse.org/. The tags affected to this

Resources / Tags	URLs / Labels
r_1	http ://www.eclipse.org/
r_2	http://www.rotas.xl.pt/1005/100.shtml
r_3	http://www.aptana.com/
r_4	http://islandreefjob.com/
t_1	java
t_2	programming
t_3	development
t_4	IDE
t_5	ilha

Table 4.5: URLs and labels corresponding to resources and tags in Table 4.6

resource are: *java, programming, development, ide, eclipse, de código aberto, webdesign, licenza*, etc. Most of these tags are ambiguous. For example, as said above, Java produces three different senses: it can mean the Java Island (http://dbpedia.org/page/Java), or the programming language (http://dbpedia.org/class/yago

Platform%29) or even the famous coffee (http://dbpedia.org/class/yago/JavaCoffee). Table 4.7 illustrates the disambiguation of these tags by applying the pre-processing step.

Next, to recognize the context of a resource's tags, we reach the DBpedia ontology and we search for relationships between the new tags resulting from the pre-processing step. For example, the object property *programming language*[7] has as a domain the class *software*[8]; the relation between these two tags is described in the DBpedia OWL Ontology as follows.

```
<owl:ObjectProperty
rdf:about="http://dbpedia.org/ontology/programmingLanguage">
<rdfs:label xml:lang="en">programming language</rdfs:label>
<rdfs:domain rdf:resource="http://dbpedia.org/ontology/Software">
```

[7]http://mappings.dbpedia.org/index.php/OntologyProperty:ProgrammingLanguage

[8]http://mappings.dbpedia.org/server/ontology/classes/Software

$\mathcal{U}/\mathcal{R}$-$\mathcal{T}$	r_1					r_2					r_1					r_2				
	t_1	t_2	t_3	t_4	t_5	t_1	t_2	t_3	t_4	t_5	t_1	t_2	t_3	t_4	t_5	t_1	t_2	t_3	t_4	t_5
u_1	×	×	×	×		×				×	×	×	×	×		×				×
u_2	×	×	×								×	×	×	×						
u_3	×	×	×								×	×	×	×						
u_4						×				×						×				×
u_5						×				×						×				×
u_6						×				×						×				×
u_7	×	×	×	×		×				×	×	×	×	×		×				×
u_8	×	×	×	×							×	×	×	×						
u_9																				
u_{10}	×	×	×			×														

Table 4.6: The folksonomy $\mathcal{F}_1$

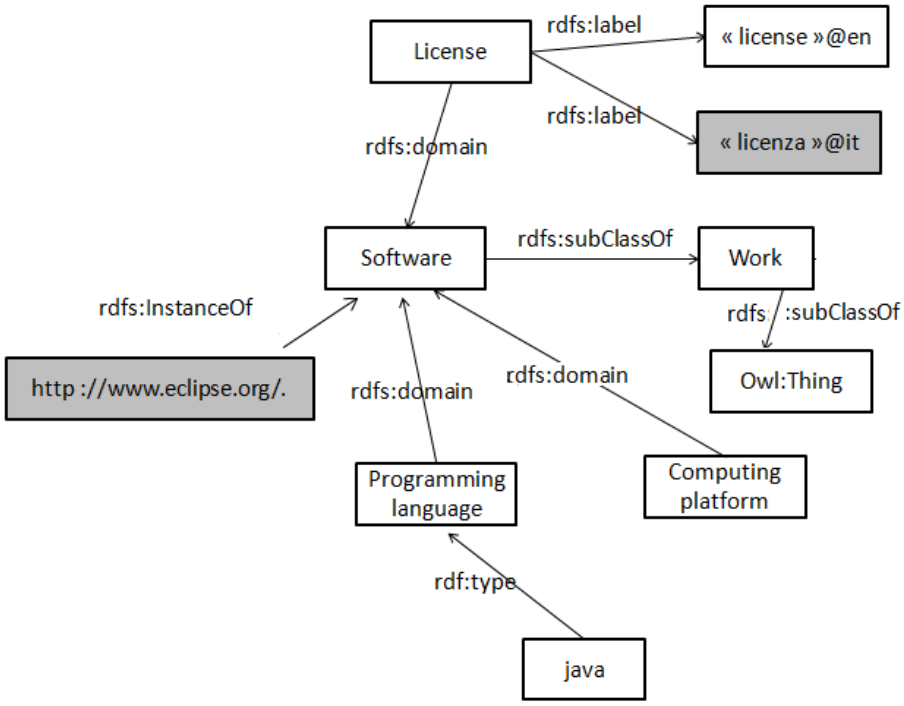


Figure 4.6: Example of enriching DBpedia with a new instance and a new concept

```
</rdfs:domain>
</owl:ObjectProperty>
```

The dominant concept found for this resource is "*Software*". So, Software is the top class defining our context. An enrichment of the DBpedia ontology takes place (cf Figure 4.6) by adding this populated resource as an instance of Software class, and by adding the concept "licenza":@it, that is the italian traduction of "license", with the *rdf:label* relation to the ObjectProperty *license* of the class *Software*.

After applying the pre-processing step for all tags annotating the different resources and recognizing the tags' context of each resource, we obtain the enriched folksonomy $\mathcal{E}F_1$ as illustrated in Table 4.8.

We present now how to obtain the new social network with the adequate clusters of users [Hamdi *et al.*, 2011]. We apply the proposed users' semantic-clustering algorithm. The execution of this algorithm with the disambiguated enriched folksonomy allows to extract the contextualized Tri-concepts $\mathcal{CTC}_\infty$, $\mathcal{CTC}_\in$ and $\mathcal{CTC}_\ni$ with:

TagSet	**NewTagSet**
java	{java, programming language, island, coffee}
programming	{programming, programming language, scheduling}
development	{development, growing, evolution, software development, developer, etc.}
IDE	IDE
eclipse	{eclipse, occultation, computing platform}
de código aberto	{de código aberto, open source, computer software}

Table 4.7: Disambiguation of a Delicious resource example after the preprocessing step

- $\mathcal{CTC}_{\infty}$ = { $\{u_1, u_2, u_3, u_7\}$, $\{r_1, r_3\}$, $\{t_1, t_2, t_3\}$, software }
- $\mathcal{CTC}_{\in}$ = { $\{u_1, u_4, u_5, u_7\}$, $\{r_2, r_4\}$, $\{t_1, t_5\}$, island }
- $\mathcal{CTC}_{\ni}$ = { $\{u_1, u_7, u_8\}$, $\{r_1, r_3\}$, $\{t_1, t_2, t_3, t_4\}$, software }

Our users' clusters or interest's groups are now computed to create the new social network, as illustrated in Table 4.9. As we see, users u_9 and u_{10} did not appear in any group since u_9 did not share resources assigned to a same context as other users and u_{10} did not have the same behaviour of tagging the resources (i. e. did not use the same tags as other users).

$U/\mathcal{R}$-$\mathcal{T}$	$r_1\|software$					$r_2\|island$					$r_1\|software$					$r_2\|island$				
	t_1	t_2	t_3	t_4	t_5	t_1	t_2	t_3	t_4	t_5	t_1	t_2	t_3	t_4	t_5	t_1	t_2	t_3	t_4	t_5
u_1	×	×	×	×		×				×	×	×	×	×		×				×
u_2	×	×	×								×	×	×	×						
u_3	×	×	×								×	×	×	×						
u_4						×				×						×				×
u_5						×				×						×				×
u_6						×				×						×				×
u_7	×	×	×	×		×				×	×	×	×	×		×				
u_8	×	×	×	×							×	×	×	×		×				
u_9																				
u_{10}	×					×														

Table 4.8: The enriched folksonomy $\mathcal{E}F_1$

Interest's Group	**Users**
software	$\{u_1, u_2, u_3, u_7, u_8\}$
island	$\{u_1, u_4, u_5, u_7\}$
-	$\{u_9, u_{10}\}$

Table 4.9: Interest's groups for the generated social network

Dataset	users' number	resources' number	tags' number
Delicious	7154	14331	11118

Table 4.10: Dataset description

4.8 Experiments

4.8.1 Large-Scale Satisfaction

The very important number of users and the dynamism of the network prevent the designers of decentralized applications from estimating the performances of their applications before deploying them. Now, if the designed application did not satisfy the large-scale conditions, then it cannot be used on the scale of Internet which is generally the purpose that social networks are looking for.

In the aim to show that our approach satisfies the large-scale conditions and supports a large number of users, resources and tags, we simulated our approach using a dataset [9] collected from a real-life system, the social tagging site Delicious. Table 4.10 presents the characteristics of this dataset.

4.8.2 Approach Evaluation

The experiments presented, in the remainder, aim at distinguishing the most efficient resources from folksonomies allowing to enrich DBpedia. We carried out these experiments using the folksonomy of Delicious social network but they could be easily adapted to use other folksonomies and social tagging sites.

Subsection 4.6.1 described our approach to recognize relevant contexts for the

[9]The dataset is available in http://data.dai-labor.de/corpus/delicious/

folksonomy resources after the pre-processing step. To evaluate the relevance of this approach, we computed, in our first experiment, the precision values for the context recognition with and without the pre-processing step.

Precision is the fraction of the number of relevant tags associated to the dominant context by the total number of tags associated to this context. We had to specify first the relevant context in order to compare it with that given by our system (dominant context) and be able to compute these precision values.

To perform such a task, we chose 100 resources from Delicious site and we asked a group of 20 people, students and researchers from both the Faculty of Sciences of Tunis and Telecom SudParis Institute, to manually indicate relevant contexts from the DBpedia ontology. We asked these evaluators informing them about the n possible contexts associated to each resource. We made sure that every resource was rated by $n+1$ evaluators so that we can take into account the decisions made by the majority. In more than 90% of cases, evaluators chose the dominant context given by our system. We notice that, for the remaining cases, evaluators chose the second dominant context and we address that its dominance is very close to the most dominant one.

We define a metric PC to assess Popularity and Clearness of each resource. Indeed, Popularity is based on the number of times a resource is shared, i.e, the number of users sharing this resource. Clearness indicates the degree of clarity, comprehensibility and transparency a resource has. This means that users sharing the resource understand its content likewise, and thus they use the same tags to annotate it.

Eq. 4.2 indicates that the more the number of users sharing such a resource (N^r_{U}) increases and the number of tags (N^r_{T}) associated to this resource decreases (that is most of users used the same tags), the more popular and clear the resource is.

$$PC^r = \frac{N^r_{\mathrm{U}}}{N^r_{\mathrm{T}}} \tag{4.2}$$

As shown in Figure 4.7, the precision in finding the relevant tags' context increases with PC^r. This means that the more the resource is popular (shared by several users) and clear, the easier for the system to find the corresponding relevant context is. We notice, similarly, that it is clear that recognizing the context with the pre-process step produces more precise results than without it

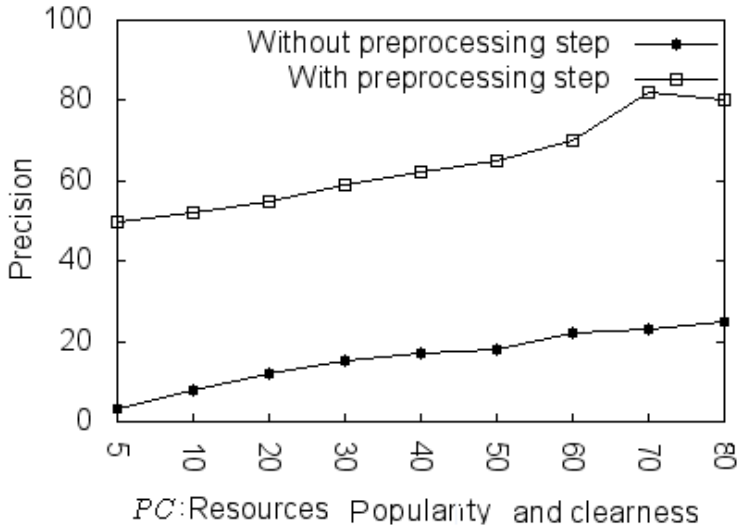


Figure 4.7: Average precision for different types of resources from Delicious

whatever the PC^r values.

We conclude that there are two factors increasing the precision metric. The first one is based on the relevance, the clearness and the popularity of the resource. Indeed, the best resources to enrich DBpedia are the most shared ones by the folksonomy users. The second factor concerns the efficiency of the preprocessing step that precedes the context recognition step. Indeed, the tagging is free and users may make many grammatical, syntactic and other mistakes. Therefore without this important step, we are threatened to loose the precision of information.

Our second experiment investigated that unambiguity should be reached to guarantee better precision values. We vary the number of contexts associated to resources and we take care on including unambiguous and ambiguous resources. The degree of unambiguity is obtained by applying the formula defined in Equation 4.3 where $\mathcal{CO}_1$ and $\mathcal{CO}_2$ correspond, respectively, to the two most dominant contexts. The obtained results are shown in Figure 4.8.

$$\mathcal{X}(\mathcal{CO}) = |dom(\mathcal{CO}_1) - dom(\mathcal{CO}_2)| \tag{4.3}$$

As would highlight these statistics, better precision values are achieved whenever resources have only two different contexts. Indeed, we observe that the lower the number of contexts is, the higher the precision is. We can also observe that

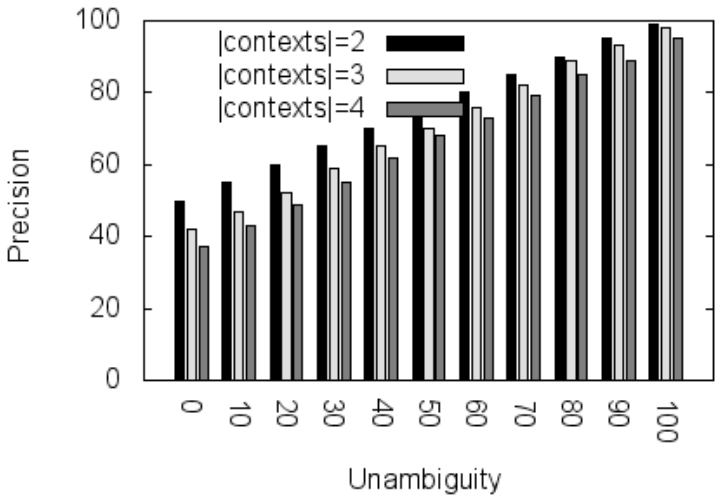


Figure 4.8: Average precision of the disambiguated resources (with pre-processing step) with different number of contexts

results ensured that unambiguous resources produce higher precision values. This resolves the problem of precising the most relevant tagged resources to the enrichment of DBpedia by instantiation. This means that, despite each resource has its dominant context, this latter may not be the most adequate one in reality due to the high ambiguity of the resource.

4.9 Conclusion

The most collaborative tagging systems enable users to create new communities or to join a community of interest. In this situation, finding a group of people who are working on the same topics is a challenge. We have proposed an approach to discover automatically semantic clusters of folksonomy users based on their domain of interests or context, consisting of new social networks. This clustering is dynamic and may be reviewed each time a new user joins the folksonomy. We use external tools like WordNet, Google and DBpedia to find meaning for the user tags in order to identify the resource context domain. We focused on the particularly challenging problem of ambiguity in tagged resources because of the issue of the free-tagging. We have validated that the pre-processing step of our method improves the precision values. The tags' context is defined as one or

several concepts of DBpedia ontology which is used as a referential domain. This ontology is then enriched with new instances referencing these resources as well as new semantic links corresponding to user tags. So, we addressed the problem of how to enrich DBpedia ontology with new concepts and instances from folksonomies. Our evaluation showed that not all resources shared by folksonomies' users could be relevant instances to DBpedia ontology.

The proposed social contextual information and the users' interests have significant influences on trust evaluation. In the next chapter, we describe how they are introduced to be included on the basis of our method of direct trust computing and, thus, trusted social networks generating.

Chapter 5

IRIS: Direct Trust Management in OSNs

5.1 Introduction

Trust based-systems have become widespread for OSNs. Such systems aim to enhance the level of trust among members, whether the goal is to help users to accurate evaluations to encourage benevolent behaviours.

In this Chapter we are interested in direct trust management in OSNs. In fact, establishing trust among the OSN users plays a vital role in improving the quality of services and enforcing security for the social activities. Our work is looking for introducing how direct trust, i.e. trust between two users directly connected, can be associated between users. Our key ideas and contributions are as follows:

1. We propose a novel approach for computing direct trust degrees between OSN users. It considers the **I**nteractions between users, their existing **R**elationship types and their **I**nterest **S**imilarity ($IRIS$).

2. We generate the trusted social network to distinguish between malicious, controversial and benevolent users.

3. To validate the effectiveness of $IRIS$, we conducted several experiments with a data set extracted from FOAF files.

5.2 The Trust Network Generating Issue

In OSNs, a trust network is critical and is the basis for the trust evaluation between two non-adjacent participants and the reputation evaluation of users, as it contains some important intermediate users, the direct trust relations between those users and social relations. Extracting the trust network between users becomes a fundamental and essential step before performing trust propagation or reputation and has important influences on their evaluation ([Golbeck et Hendler, 2006], [Hamdi *et al.*, 2013], [Hamdi *et al.*, 2012a], [Liu *et al.*, 2010]).

For example, in the sample social network depicted in Fig. 4.1, the user A is looking for a Tennis coach and D is a Tennis coach. In such a situation, as indicated in the theory of Social Psychology ([Christianson et Harbison, 1996], [Mansell et Collins, 2005]) and Computer Science ([Golbeck et Hendler, 2006], [Hamdi *et al.*, 20
A can evaluate the trustworthiness of D based on the trust network from A to D by using trust propagation and trust transitivity methods. Therefore, direct trust values between intermediate users are essential to make such a decision.

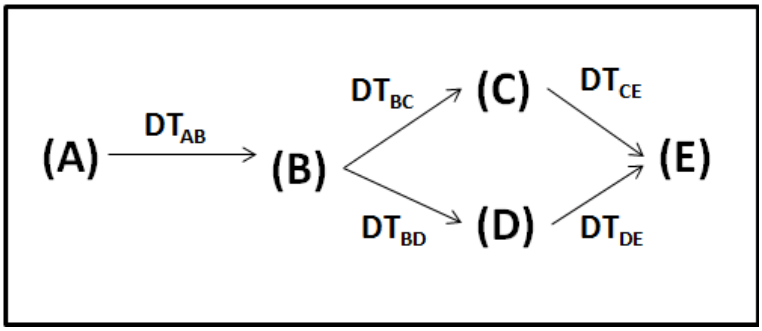


Figure 5.1: An example of trust network

For example, we consider the trust network depicted in Fig. 5.1 where the user A asks about the trustworthiness of the user E. We assume that the user B does not know D well. However, (s)he knows C very well and trusts him/her (i.e, $t_{BC} > t_{BD}$). Thus, C's recommendation of E may be considered more credible than that of D. Namely, it is not reasonable to adopt the trust value given by the user B to D and exclude that given by B to C. Therefore, it is relevant and essential to address the challenge of trust network extraction problem to provide the trust evaluation between two unknown users.

5.3 Social Contextual Impact Factors

As indicated in Social Psychology ([Adler, 2001], [Lichtenstein et Slovic,], [Miller, 2012], [Palchykov *et al.*, 2012]), social contexts have significant influence on trust evaluation. Then, based on the social contexts in social environments, several social context impact factors are proposed to be taken into account, as follows.

5.3.1 Social Relationships

To support social networking, it is helpful to represent various properties of, and relationships between, persons expressing a wide range of self-description and social connectivity. In Social Science [Miller, 2012], it is indicated that two persons can have more than one type of social relationships. In fact, persons whom participated in such a networked (sub)society may be friends, relatives, work collaborators, employees, and so on. A diagram that illustrates a schematic example of a social network with four kinds of links is depicted in Fig. 5.2.

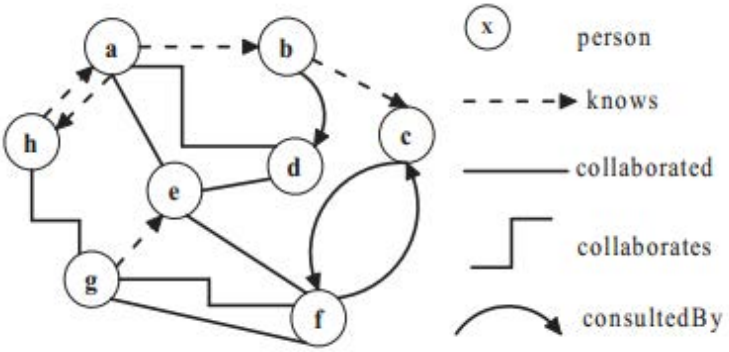


Figure 5.2: A social network with links

As illustrated in Social Psychology [Adler, 2001], [Ashri *et al.*, 2005], [D.J., 2009], a person can trust and have more social interactions with the persons with whom he/she has more intimate social relationships than those with whom (s)he has less intimate social relationships.

5.3.2 Preferences and interests

In Social Psychology, preferences could be conceived of as a person's attitude towards a set of objects, typically reflected in an explicit decision-making process [Lichtenstein et Slovic,]. A person can have different interests in different domains. For example, a researcher prefers doing collaboration with others whom have the same research interests with him/her, and the researcher may like playing Tennis as well.

The main objective of the survey presented in [Bhuiyan, 2010] was to survey and analyse the online users' opinion about the relationship between the trust and interest similarity of the users. They showed that the overall attitude of the online user about the relationship between trust and interest similarity is positive. In fact, most of the people think that there is a positive relationship that exists between trust and interest similarity among different users. They prefer to trust more those opinions which taste is similar to their ones in a particular matter. In Chapter 4, we have described our proposed method for discovering users' interests. We can use this factor to evaluate the direct trust between two users.

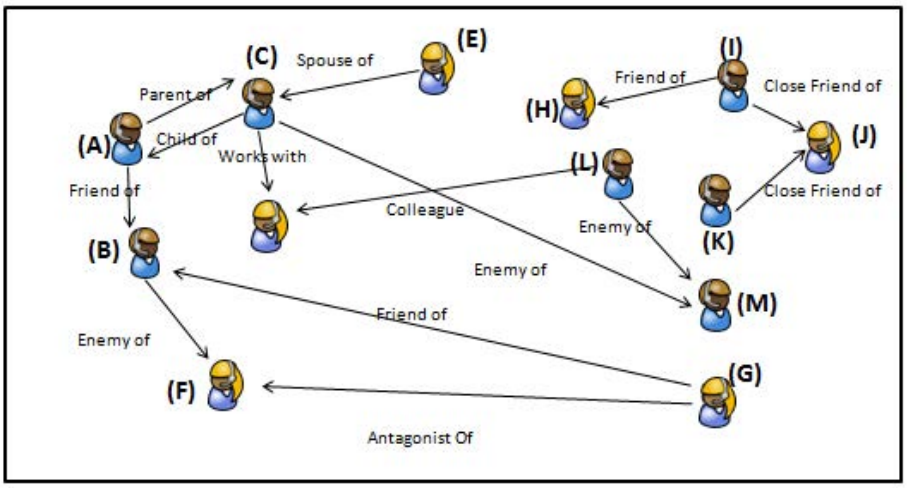


Figure 5.3: An example of an online social network

5.3.3 Interactions

The trust, a social network user A has in his/her friend B, may refer to the opinions of a user A about his/her interactions with B [Hamdi *et al.*, 2012a]. Examples of interactions include downloading files, posting informations, replying to another user, etc. The existing interactions between two users A and B strongly influence the trust between them.

5.4 Exploitation of FOAF vocabulary Information

These last years, semantic Web researchers have focused on social relationships. In addition, we have seen a dramatic increase in the amount of published RDF documents using the Friend of a Friend (FOAF) vocabulary [Brickley et Miller, 2010], providing a valuable resource for investigating how early semantic Web adopters use this technology as well as build social networks. In this section, we describe the FOAF vocabulary and we present its necessary information that we exploit to put on value our proposed method.

5.4.1 Defining Users' Relationships

The property "knows" from the FOAF vocabulary could be useful to create social links between users (i.e. one user knows another one). This property possesses several, more specific, sub-properties defined in RELATIONSHIP vocabulary [Davis et Jr, 2010]. This latter describes the different relationships that can exist between users. The list of the relationships is {*Acquaintance Of, Antagonist Of, Apprentice To, Child Of, Close Friend Of, Collaborates With, Colleague Of, Employed By, Employer Of, Enemy Of, Engaged To, Friend Of, Grandchild Of, Grandparent Of, Has Met, Influenced By, Knows In Passing, Life Partner of, Lives With, Lost Contact With, Mentor Of, Neighbor Of, Parent Of, Sibling Of, Spouse Of, Works With, Would Like To Know*}. More details about each property and its use can be found in [Davis et Jr, 2010].

Example 2: An example of an online social network is depicted in Figure 5.3, where users are connected through multiple direct relationship types. Alice (A), for instance, has a direct relationship of type *FriendOf* with Bob (B) and a direct relationship of type *ParentOf* with Carl (C).

After defining the relationships, we can consider a social network as a directed labeled graph. Each node of the graph denotes a user in the network, whereas edges represent the existing relationships between users. An edge is directed from the node specifying the relationship to the node for which the relationship has been specified, whereas the label associated with each edge denotes the type of the relationship.

We can formally redefine a social network as follows.

Definition 1 *(Social Network)*

A social network $\mathcal{SN}$ *is a graph* $\mathcal{SN} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ *where* $\mathcal{V}$ *is the set of users,* $\mathcal{R}$ *is the set of possible relationship types between them and* $\mathcal{E}$ *is the set of directed links between users labeled with relationship types.*

5.4.2 Presenting Users' Interests

Many properties ("interest", "Topic interest", "theme") from the FOAF vocabulary are used to define the users' interests. This information enriches the social network by assigning the users to different groups according to their interests. In fact, a group is composed of a set of users sharing interests in the same domain.

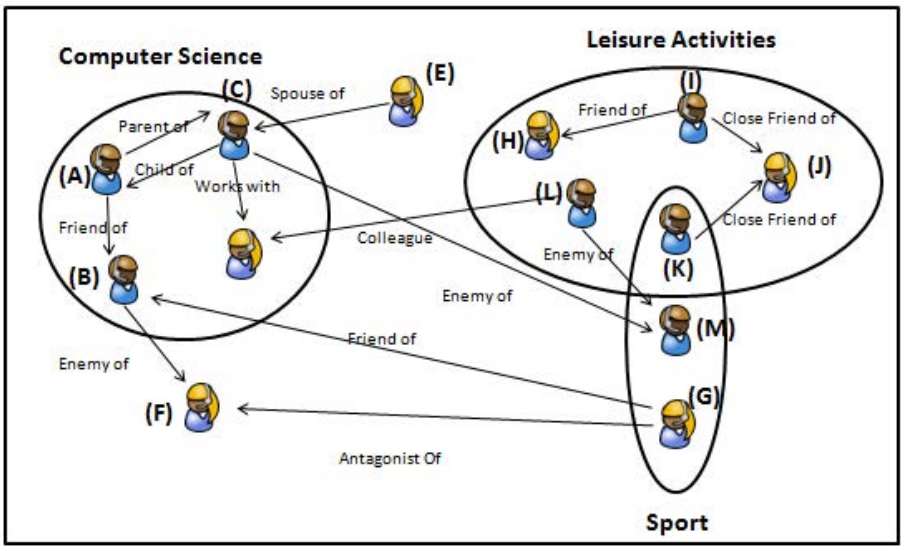


Figure 5.4: Classification of users according to their interests

Example 3: A classification example of users according to their domain of interests from the FOAF information is presented in Figure 5.4. For instance, Alice (A), Bob (B), Carl (C) and David (D) are all interested in the Computer Science domain.

5.5 Direct Trust Computation

We model the direct trust between a user v and a user v', $DT(v, v')$, as a real value ranging within the unit interval. On the one hand, if $DT(v, v') = 0$, the degree of trust v has in v' is the minimum, i.e, v totally distrusts v'. On the other hand, if $DT(v, v') = 1$, then this implies that v gives a total trust to v'.

Our approach for computing trust values between directly connected users, $IRIS$, considers the social contextual impact factors described in Section 5.3, namely the nature of Interactions between users, the types of Relationships connecting users, as well as their Interests Similarity.

Category	Relationships
Close relationships	{Child Of, Close Friend Of, Engaged To, Grandchild Of, Grandparent Of, Life Partner of, Lives With, Parent Of, Sibling Of, Spouse Of}
Friendships	{Colleague Of, Employed By, Employer Of, Friend Of, Mentor Of, Works With}
Acquaintance	{Collaborates With, Influenced By, Neighbor Of}
Superficial acquaintance	{Has Met, Knows In Passing, Lost Contact With}
Bad acquaintance	{Antagonist Of, Enemy Of}

Table 5.1: Relationship categories

5.5.1 Direct Trust Metrics

The Relationship Trust

The relationship type, denoted $r_{v \to v'}$, is a direct one since v and v' are directly connected through an edge $v \to v'$. The direct trust level assigned to a friend should be different from the one assigned to an enemy or a close friend.

We define a friendship function, $\mathcal{F} : \mathcal{V} \times \mathcal{R} \times \mathcal{V} \to [0, 1]$, that computes the trust degree ft in $[0, 1]$ between two vertices (users) according to their friendship relation. $F(v, r, v') = ft$ underlies that the user v assigns a friendship trust degree ft to his friend v'. As presented above, there are many relationship types connecting users in a social network. To be able to capture this potentially large amount of relationship information, we need to generalize it to other relationship types. In our method, we use relationship categories to represent which aspect

Algorithm 6: RelationshipTrustComputing

Data: 1: v, the user
2: V, the set of v's neighbors

Result: The relationship trust value ft

```
1  begin
2      foreach neighbor v' ∈ V do
3          r ← getRelationship(v, v');
4          switch r do
5              case r ∈ Close relationships
6                  ft_{v→v'} = 1;
7                  break;
8              case r ∈ Friendships
9                  ft_{v→v'} = 0.75;
10                 break;
11             case r ∈ Acquaintance
12                 ft_{v→v'} = 0.5;
13                 break;
14             case r ∈ Superficial acquaintance
15                 ft_{v→v'} = 0.25;
16                 break;
17             otherwise
18                 ft_{v→v'} = 0;
19     return (ft)
```

of closeness and proximity we are referring to, and trust values for the different levels of relationship within each category. The relationship categories are given in Table 5.1.

Algorithm 6 returns the trust values corresponding to the direct relationships. It takes as input a user v and the set of his neighbors V. In Line 2-3, the algorithm iteratively searches for the relationship type r between the user and each of his neighbours. From Line 4 to 18, it returns the corresponding relationship trust values ft according to the category of this relationship type.

The Interactions' Trust

Different interactions can take place between two users v and v' and influence the trust between them. To model this interactions' trust, we need first to identify the metrics that can be used. We propose metrics including the number of positive and negative feedbacks given from v to v', according to their interactions.

We define a satisfaction mapping $\mathcal{S} : \mathcal{V} \times \mathcal{I} \times \mathcal{V} \rightarrow \{0, 1\}$ where $S(v, i, v')$ implies the satisfaction value a user v gives to his neighbor v', based on their mutual interaction i. If $S(v, i, v') = 0$, then v is not satisfied by the interaction i, otherwise, $S(v, i, v') = 1$ means that v is satisfied by the interaction i.

We define, in Eq 5.1, a trust value assigned by user v to user v' after n interactions.

$$it^n_{v \rightarrow v'} = \begin{cases} 1 - \frac{Neg_{v \rightarrow v'}}{Pos_{v \rightarrow v'}} \; if \; Pos_{v \rightarrow v'} > Neg_{v \rightarrow v'} \\ 0 \; otherwise \end{cases} \tag{5.1}$$

For each user v, the local value $Pos_{v \rightarrow v'}$ presents the sum of p interactions between v and v' considered as positive by v and $Neg_{v \rightarrow v'}$ the sum of $n - p$ interactions between v and v' considered as negative by v. These values are computed recursively as shown respectively in Eq 5.2 and Eq 5.3.

$$Pos_{v \rightarrow v'} = \sum_{j=1}^{n} S(v, i_j, v') \; if \; S(v, i_j, v') = 1 \tag{5.2}$$

$$Neg_{v \rightarrow v'} = n - Pos_{v \rightarrow v'} \tag{5.3}$$

The Interests' Similarity Trust

Sharing the same interests would increase trust between two users. Indeed, generally, recommendation systems based on users' collaboration compute the recommendations by measuring resemblance between users. Golbeck also highlighted, through the analysis of data in the FilmTrust website [Golbeck, 2006b], that there is a correlation between similarity of users and trust between them.

Each user v in a social network is interested in N different domains, where $N = |domains_v|$. Thus, we propose a similarity trust degree, as shown in Eq 5.4, that allows to assess to which extent a user v' is similar to a user v. Note that neither trust nor this definition of similarity are symmetric.

$$st_{v \rightarrow v'} = \frac{|domains_v \cap domains_{v'}|}{|domains_v|} \tag{5.4}$$

5.5.2 Generating the Trusted Social Network

Direct Trust

Considering the direct trust metrics as described above (Subsection 5.5.1), we can compute the direct trust assigned to v' by v as presented in Eq 5.5, where α, β and γ are the normalized factors of weights assigned respectively for the friendship trust, the interactions' trust and the similarity trust, with $\alpha + \beta + \gamma = 1$ and $\{\alpha, \beta, \gamma\} \in [0, 1]$. It is worth of mention here that we assign the three parameters equal weights to equate the different direct trust metrics, so we consider $\alpha = \beta = \gamma = 1/3$.

$$DT_{v \to v'} = \alpha ft_{v \to v'} + \beta it^{n}_{v \to v'} + \gamma st_{v \to v'} \tag{5.5}$$

By associating a trust value with each directed edge linking two users, we obtain a trusted social network as defined below.

Definition 2 *(*Trusted Social Network*)*

A trusted social ntework is the trusted graph $\mathcal{TSN} = (\mathcal{SN}, \mathcal{DT})$, where $\mathcal{SN}$ is the social network and $\mathcal{DT}$ is the value function that describes the degrees of the direct trust relationships between two participants in the social network.

Example 4: Figure 5.5 depictes the generated social network after the computation of the direct trust. Each relationship type is associated with a trust level denoting the direct trust between the two users participating in the given relationship. We consider the trust level existing between Alice (A) and Bob (B) is $DT(A, B) = 0.7$. The relationship type assigned to Bob is *FriendOf*, so the friendship level is $ft_{A \to B} = 0.75$. We also consider their interactions' trust is $it_{A \to B} = 0.35$ and, since they share the same interests, their similarity trust $st_{A \to B} = 1$. Thus, the obtained direct trust value is $DT(A, B) = 1/3 \times (0.75 + 0.35 + 1) = 0.7$.

Enriching the Trust Network

The trust computation of DT allows to: (i) know and consequently isolate *malicious* users from the network; (ii) know and encourage *benevolent* users by rewarding them with good reputation. Therefore, the trust values will be of less use in preventing *malicious* users from giving negative interactions.

We now redefine, in Eq 5.6, the $controversialityPercentage$ quantity introduced in [Guth *et al.*, 2006]. A user with 1 as $controversialityPercentage$ is

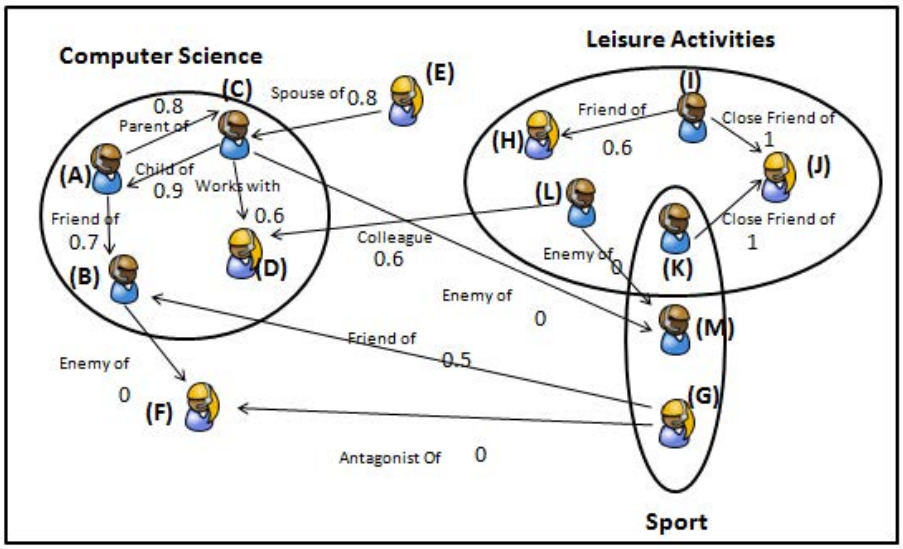


Figure 5.5: Trusted Social Network

totally trusted by all his judgers, i.e. $|receivedDistrust(v)| = 0$, and is judged as *benevolent*. On the other side, a user with -1 is totally distrusted by all his judgers, i.e. $|receivedTrust(v)| = 0$ and is judged as *malicious*. Users at these two extremes are non-controversial since all the remaining users agree on their opinions about them. Conversely, a user v with 0 as controversiality percentage, i.e. $|receivedTrust(v)| = |receivedDistrust(v)|$, is highly controversial since other users split into two same-sized opinions group about this user, i.e. one half of users trusts him and the other half distrusts him.

$$controversialityPercentage(v) = \frac{|receivedTrust(v)| - |receivedDistrust(v)|}{|receivedTrust(v)| + |receivedDistrust(v)|} \tag{5.6}$$

Example 5: Consider the generated trusted social network depicted in Figure 5.5. The network can thus be enriched by defining *benevolant*, *controversial* and *malicious* peers. For instance in Figure 5.5, John (J) is totally trusted by all his judjers, he is then considered as *benevolent* user. Whereas Mark (M) and Frederic (F) are totally distrusted by all their judjers, so they are considered as *malicious*. Others like Alice (A), Bob (B) and Carl (C) are *controversial* users, with different levels of controversiality.

We modify the $controversialityLevel$ quantity, introduced in [Guth *et al.*, 2006], as shown in Eq 5.7. The $controversialityLevel$ of a user is the number of users

who disagree with the majority in issuing a statement about that user. For example, a user who received 21 distrust statements and 14 trust statements has a *controversialityLevel* of 2/5.

$$controversialityLevel(v) = \frac{min(|receivedTrust(v)|, |receivedDistrust(v)|)}{|receivedTrust(v)| + |receivedDistrust(v)|} \tag{5.7}$$

A user who has a *controversialityLevel* of α is called $\alpha - controversial$. $0 - controversial$ users received only trust or distrust statements and they are non controversial. A user who has a *controversialityLevel* not less than α is defined as at least $\alpha - controversial$ user.

It is still unpractical to run trust management algorithms each time a trust requester asks about the trustworthiness of an other user. Thus, in the remainder, we propose a new method to satisfy the needs of OSNs' users by storing important trust data in a well organized and easy to understand structure using the FOAF vocabulary.

5.6 Linking Semantic Web with trusted social networks

Our goal in this section is to enrich and publish the FOAF ontology with trust information helping the OSNs' users to constitute a common knowledge base that is not only helpful for each user to know those whom (s)he can trust, but also it encourages users to have benevolent behaviours.

5.6.1 The proposed Approach

The lack of trust details in the FOAF data could be addressed by enriching it using the information available on trust social networks given by methods like IRIS. We therefore, decided to map the instances from the trust values set to FOAF. For that purpose, we introduce an RDF schema (cf. Figure 5.7), designed to extend the foaf:Person and foaf:knows classes. The FOAF enrichment using the proposed schema is to allow users to indicate a level of trust for people they know. As we propose to extend FOAF, users are still identified by their email address. Our trust schema adds to FOAF the **foaf:directTrust** property, having

foaf:directTrust Property value	**Direct Trust levels' range**	**Description**
Very Low	[0, 0.2]	The person has a **very low** trust level from the person knowing him
Low	]0.2, 0.4]	The person has a **low** trust level from the person knowing him
Medium	]0.4, 0.6]	The person has a **medium** trust level from the person knowing him
High	]0.6, 0.8]	The person has a **high** trust level from the person knowing him
Very High	]0.8, 1]	The person has a **very high** trust level from the person knowing him

Table 5.2: The proposed vocabulary for describing direct trust between people

as domain value foaf:Person.

The **foaf:directTrust** property is functional (i.e, a user can assign at most one direct trust value to another one) and asymmetrical (i.e, for two users involved in a relationship, direct trust is not necessarily the same in both directions), and presents the trust level given by a person to another one he knows. Therefore, we propose this property to be linked to the foaf:knows property.

- Domain: foaf:Person; having this property implies being a Person.

- Range: every value of this property is a String.

- Used with: foaf:knows property.

In the FOAF description of a user, the **foaf:directTrust** property relates a Person to a string representing the direct trust level that he has from the given user. The value of this property belongs to the predefined set "very low", "low", "medium", "high", "very high". Table 5.2 presents the Trust vocabulary for describing the direct trust between people.

5.6.2 Merging and Querying Current OSNs

Since an RDF graph is wealthy of stored information, it is natural to try to retrieve this information from the graph. SPARQL (SPARQL Protocol and Query Language) is a query language designed to do this [Prud'hommeaux et Seaborne, 2006]. SPARQL is based on matching patterns in the RDF graph, by specifying parts of the relevant triples, while giving unknown values a variable name. When the query is processed, the value of these variables can be returned.

For example, to retrieve the most benevolent users from FOAF files, we use the SPARQL query, presented below, returning the set of users having the highest trust values ("very high"), without having to crawl the Web for data:

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE { ?user foaf:name ?name .
        ?user foaf:directTrust "very high".  }
```

5.6.3 Illustrative example

We present, in this section, an illustrative example to concretize our contribution. We make use of the example presented in Wikipedia [1]. As shown in Figure 5.6, there is a person named "Jimmy Wales", identified by his email "jwales@bomis.com". His nickname is "Jimbo" and he is interested in wikimedia. This person knows two persons: "Angela Bisley" and "Jimmy Criket". However, there is no indication about trust or reputation of concerned persons.

To emphasize our contribution, we assign two direct trust values from "Jimmy Wales" to "Angela Bisley" and "Jimmy Criket" that are equal to respectively 0.65 and 0.85. In Figure 5.7, we can guess the difference after the enrichment (green lines) of the RDF document describing Jimmy Wales, with trust and reputation information. On the one hand, the "trustworthy" reputation value is assigned to "Jimmy Wales". On the other hand, a sibling trust relation with the property foaf:directTrust to indicate that "Jimmy Wales" gives a "high trust" level to "Angela Bisley" and a "very high trust" to "Jimmy Criket".

We took a screenshot, shown in Figure 5.8, when displaying the Foaf ontology

[1] http://fr.wikipedia.org/wiki/Friend_of_a_friend

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <foaf:Person>
    <foaf:name>Jimmy Wales</foaf:name>
    <foaf:title>Mr.</foaf:title>
    <foaf:givenName>Jimmy</foaf:givenName>
    <foaf:familyName>Wales</foaf:familyName>
    <foaf:mbox rdf:resource="mailto:jwales@bomis.com"/>
    <foaf:homepage rdf:resource="http://www.jimmywales.com/"/>
    <foaf:nick>Jimbo</foaf:nick>
    <foaf:depiction
rdf:resource="http://www.jimmywales.com/aus_img_small.jpg"/>
    <foaf:interest>
      <rdf:Description rdf:about="http://www.wikimedia.org"
rdfs:label="Wikipedia"/>
    </foaf:interest>
    <foaf:publications
rdf:resource="http://www.jimmywales.com/pubs/publications.rdf"/>
    <foaf:account>
     <foaf:OnlineAccount>
      <rdf:type
rdf:resource="http://xmlns.com/foaf/0.1/OnlineChatAccount"/>
      <foaf:accountServiceHomepage
rdf:resource="http://www.freenode.net/"/>
      <foaf:accountName>jwales</foaf:accountName>
    </foaf:OnlineAccount>
    </foaf:account>
    <foaf:knows>
      <foaf:Person>
        <foaf:name>Angela Beesley</foaf:name> <!-- Wikimedia Board
of Trustees -->
      </foaf:Person>
    </foaf:knows>
    <foaf:knows>
      <foaf:Person rdf:about="http://jimmycricket.com/me">
        <foaf:name>Jimmy Criket</foaf:name>
      </foaf:Person>
    </foaf:knows>
  </foaf:Person>
</rdf:RDF>
```

Figure 5.6: An example of FOAF document before enrichment process

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <foaf:Person>
    <foaf:name>Jimmy Wales</foaf:name>
    <foaf:title>Mr.</foaf:title>
    <foaf:givenName>Jimmy</foaf:givenName>
    <foaf:familyName>Wales</foaf:familyName>
    <foaf:mbox rdf:resource="mailto:jwales@bomis.com"/>
    <foaf:homepage rdf:resource="http://www.jimmywales.com/"/>
    <foaf:nick>Jimbo</foaf:nick>
    <foaf:reputation>"trustworthy"</foaf:reputation>
    <foaf:depiction
rdf:resource="http://www.jimmywales.com/aus_img_small.jpg"/>
    <foaf:interest>
      <rdf:Description rdf:about="http://www.wikimedia.org"
rdfs:label="Wikipedia"/>
    </foaf:interest>
    <foaf:publications
rdf:resource="http://www.jimmywales.com/pubs/publications.rdf"/>
    <foaf:account>
     <foaf:OnlineAccount>
      <rdf:type
rdf:resource="http://xmlns.com/foaf/0.1/OnlineChatAccount"/>
      <foaf:accountServiceHomepage
rdf:resource="http://www.freenode.net/"/>
      <foaf:accountName>jwales</foaf:accountName>
    </foaf:OnlineAccount>
    </foaf:account>
    <foaf:knows>
      <foaf:Person>
      <foaf:name>Angela Beesley</foaf:name> <!-- Wikimedia Board
of Trustees -->
        <foaf:directTust>high</foaf:directTust>
      </foaf:Person>
    </foaf:knows>
    <foaf:knows>
      <foaf:Person rdf:about="http://jimmycricket.com/me">
        <foaf:name>Jimmy Criket</foaf:name>
        <foaf:directTust>very high</foaf:directTust>
      </foaf:Person>
    </foaf:knows>
  </foaf:Person>
</rdf:RDF>
```

Figure 5.7: An example of FOAF document after enrichment process (Green lines)

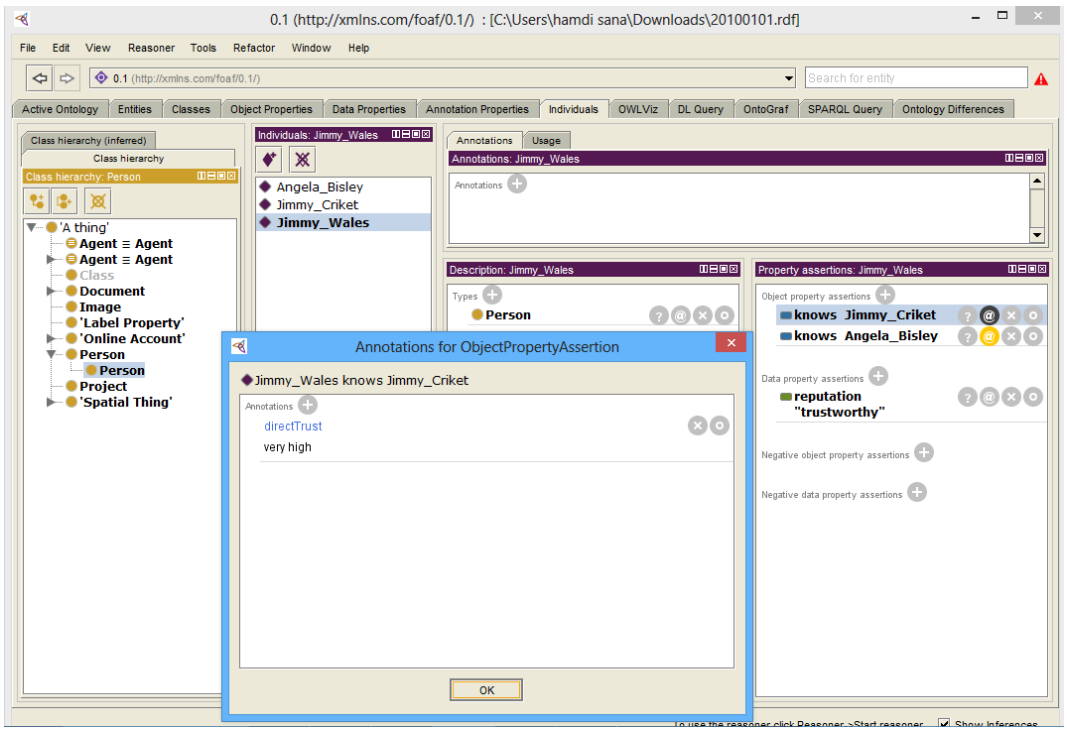


Figure 5.8: A screenshot spotlighting the foaf:reputation and foaf:directTrust properties

using the protégé's editor [2], after adding the three persons introduced on the example.

5.7 Experiments on IRIS

In this Section, we shed light on the performance of our proposed method. First, we test the accuracy metrics; then we present a comparison with some existing methods and we discuss the results.

5.7.1 Data collection

One of the most important aspects of trust management solutions is the process of data collection. In general, for the development of the trust management systems mentioned in this work, data related to the users' behaviour, interests and relationships is collected and then analysed.

[2] http://protege.stanford.edu/

We use the foafPub data set from http://ebiquity.umbc.edu/ which is a dataset of information extracted from FOAF files collected during the Fall of 2004. A total of 201,612 RDF triples with provenance information are included. The dataset is distributed as a zip file containing SQL commands. The SQL commands were generated from the original MySQL database using the export command.

5.7.2 Accuracy Metrics

To test the accuracy, we adapted the accuracy metrics in [Jiang et Wang, 2011] and [Shekarpour et Katebi, 2010] including absolute error, precision, recall, and Fscore.

Absolute Error

Absolute error is the difference between the actual value of trust and the computed value from the proposed method.

$$Absolute\ error = |calculatedTrust - actualTrust| \tag{5.8}$$

The actual value of trust is obtained by using Richardson's technique [Richardson *et al.*, 2003] which uses the concept of quality of users assigning a trust value to each node. Each user has a quality measure $q_v \in [0, 1]$. A user's quality determines the probability that a statement given by the user is true and complete. The higher the user's quality, the more likely to be trusted he is. Therefore, for any pair of users v and v' where v trusts v' :

$$actualTrust_{vv'} \in [max(q_{v'} - \delta_{vv'}, 0), min(q_{v'} + \delta_{vv'}, 1)] \tag{5.9}$$

In Eq 5.9, q_v is the quality of the user v and $\delta_{vv'} \in [0, 1]$ is a noise parameter that determines how accurate users were at estimating the quality of the user they were trusting. We suppose that a user with low quality is bad at estimating trust, so for these experiments we set $\delta_{vv'} = \frac{1-q_v}{2}$.

Precision, Recall and Fscore

The accuracy represents the ability of predicting a user to be trusted or not. Based on the defined criterion for accuracy, making a right decision is the ultimate metric for comparison of different existing methods. We set a threshold = 0.5 for

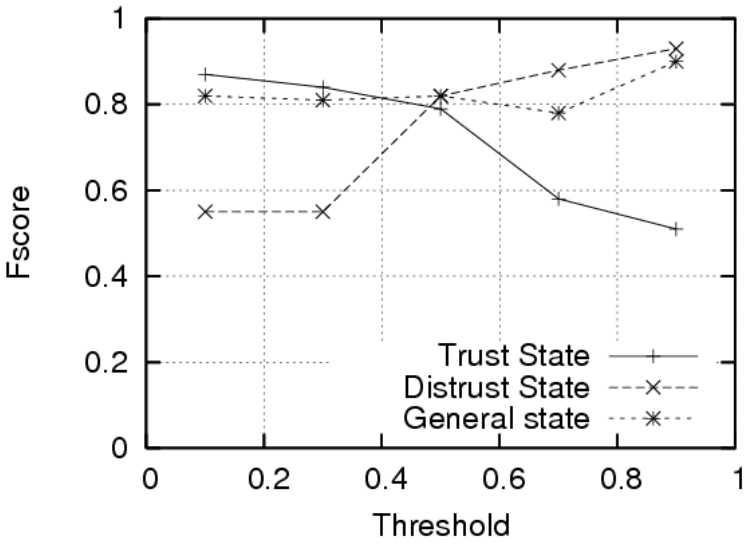


Figure 5.9: Comparison of Fscore applied to IRIS for different threshold

deciding to trust or not to trust. If the calculated DT value is equal to or greater than 0.5, we trust. Otherwise, we distrust the user. We use both precision and recall metrics to compare the accuracy of methods in making the trust decision. Precision and recall are defined for three states concerning trust, as follows.

Trust State: In this state, parameters that are used to compute the accuracy are:

X_t = the set of friends that a user actually trusts.

Y_t = the set of friends that our algorithm suggests to trust.

$$Precision_t = \frac{|X_t \cap Y_t|}{|Y_t|}, Recall_t = \frac{|X_t \cap Y_t|}{|X_t|} \tag{5.10}$$

Distrust State: In this state, parameters that are used to compute the accuracy are:

X_t = the set of friends that a user does not actually trust.

Y_t = the set of friends that our algorithm suggests to distrust.

$$Precision_d = \frac{|X_d \cap Y_d|}{|Y_d|}, Recall_d = \frac{|X_d \cap Y_d|}{|X_d|} \tag{5.11}$$

General State: In this state, parameters that are used to compute the accuracy are:

$$Precision_{total} = \frac{|X_t \cap Y_t| + |X_d \cap Y_d|}{|Y_t + Y_d|} \tag{5.12}$$

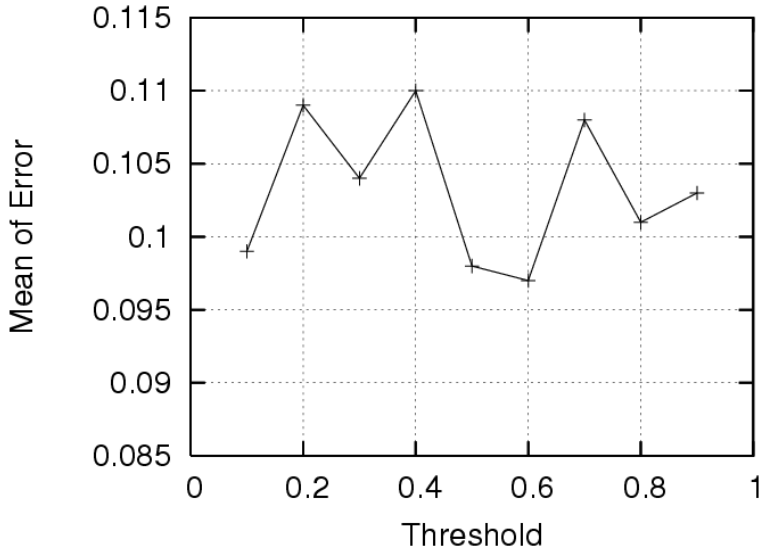


Figure 5.10: Mean of error for $IRIS$

$$Recall_{total} = \frac{|X_t \cap Y_t| + |X_d \cap Y_d|}{|X_t + X_d|} \tag{5.13}$$

We use the Fscore metric (Eq. 5.14) to compute the accuracy using recall and precision jointly.

$$Fscore = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{5.14}$$

5.7.3 Results for the IRIS Method

The programs computing the different metrics for the IRIS method were run for a range of threshold values $th \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Figures 5.9 and 5.10 respectively show the behaviour of this parameter with respect to accuracy metrics.

In Figure 5.9, as far as th varies, the accuracy decreases significantly in trust state while it increases in distrust state, and steadily increases in general state. This finding indicates that whenever we are interested in just predicting to trust or to distrust, then $th = 0.5$ would be the best choice. Nevertheless, Figure 5.10 shows the behaviour of the mean of error with th. It has the best values when th ranges between 0.5 and 0.6. From these two figures, the optimum value of th could be 0.5.

Method	Fscore
Max-Mean	0.49
Tidal Trust	0.67
Max-Min	0.72
Max-*	0.73
Max-weight	0.81
IRIS	**0.82**

Table 5.3: Fscore for different methods

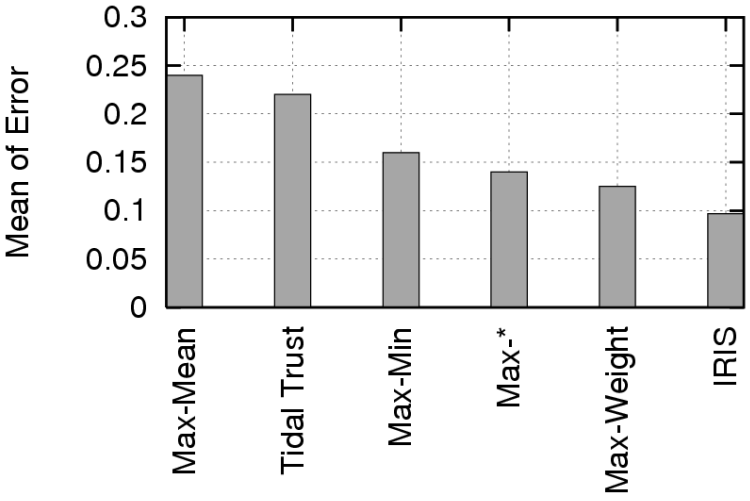


Figure 5.11: Mean of error versus the threshold

5.7.4 Comparison with Other Methods

Table 5.3 compares Fscore for the proposed method with TidalTrust [Golbeck, 2006b] and fuzzy based composition methods such as Max-min, Max-*, Max-mean and Max-weight. For a threshold value of 0,5, it is observed that the accuracy is the highest and reaches 82% for $IRIS$. Figure 5.11 shows that $IRIS$ has the best mean of error (for a threshold of 0,5): it is lower than other methods and is around 0.098.

5.8 Conclusion

In this Chapter, the new IRIS method for direct trust computation and evaluation is proposed. It aims at generating trusted social networks, what is useful to develop different trust-based methods for computing indirect trust or for providing the access control policies. This new method is tested experimentally using a data set extracted from FOAF files. The tests showed that our work presents high accuracy. The obtained results are compared and contrasted with those obtained from other methods. Experimental results demonstrate that the proposed approach computes more accurate trust results than existing methods. Moreover, we introduced a method for linking trust in OSNs with the Semantic Web. Indeed, it presents an enrichment of the FOAF ontology schema with new properties related to trust.

Chapter 6

TISON: Trust Inference for Social Networks

6.1 Introduction

In Chapter 5, we have introduced the direct trust computation and the trust network generating methods in OSNs. After extracting the trust network, we can evaluate the trustworthiness of a target user by using trust propagation methods to propagate the trust via the social trust paths in the trust network. However, in large-scale trust-oriented social networks, there could be tens of thousands of social trust paths between a source participant and a target one [Kunegis *et al.*, 2009]. Evaluating the trustworthiness of the target participant based on all these social trust paths is very time consuming, and thus cannot be applied into real applications [Baase et Van Gelder, 2000], [Liu, 2013]. Alternatively, we can search the optimal path yielding the most trustworthy trust propagation result from multiple paths. We call this the trust paths' selection problem that is known to be a challenging research problem [Hamdi *et al.*, 2013]. If there is a trust path linking two non-adjacent users, the source participant can evaluate the trustworthiness of the target one along an existing path based on the trust transitivity property (i.e., if A trusts B and B trusts C, then A trusts C to some extent) under some semantic constraints [Jøsang et Pope, 2005]. The computation of the value of trust for the target participant requires an understanding of how trust is transitive along the trust path, which is a critical and challenging problem in OSNs [Golbeck, 2005], [Hamdi *et al.*, 2013].

As mentioned in Chapter 1, the trust network is modelled as a directed graph **G**. Nodes in **G** represent people and an edge between nodes i and j represents a trust relationship between them. A label on edge (i, j) represents the trust value from i to j. Notice that we assume trust as an asymmetric relation and the trust value from i to j is not the same as that from j to i.

In this chapter, we propose a trust inference model called **TISoN**. Using the example of partial trusted OSN depicted in Fig. 6.1, we further demonstrate how **TISoN** carries out trust inference from a trusted graph. Node A, or the source node, is directly connected to B, C and D, but is not directly connected to E, F, G, H, I and J. Furthermore, we see that A is indirectly connected to J via six paths: $A \to B \to E \to J; A \to B \to E \to H \to J; A \to B \to F \to H \to J; A \to C \to F \to H \to J; A \to C \to G \to I \to J$ and $A \to D \to G \to I \to J$. Suppose that we consider all paths for determining the trust inference value from A to J without having any particular preference for a path. While this may

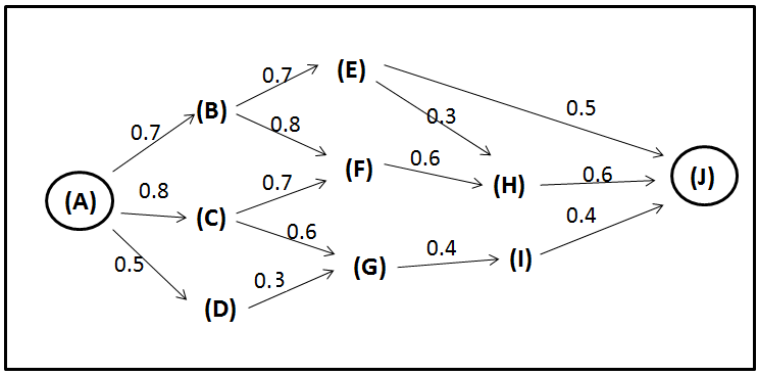


Figure 6.1: A partial trusted OSN

seem reasonable and uses all available information in the network, it has a major drawback: it is very computationally expensive to consider all possible paths between two users, even in moderately sized networks. It would represent an unreasonable effort on part of the agents to use all this information. To palliate this drawback, instead of considering all possible paths, we compute the set of trust paths that will be considered for inferring the trust value.

In the remainder, we describe our methodology for measuring indirect trust between users. On the one hand, we define the trust path notion and then we propose *TPS*, an algorithm for trust paths' searching. On the other hand, we describe our approach for computing trust. Thus, we propose a set of algorithms measuring the trust inference (*TIM*). First, we present a propagation function that computes the indirect trust specific for each trust path. Then, we discuss different aggregation methods to aggregate the trust values coming from the considered paths.

6.2 TPS : Trust Paths' Searching

A trust path is a basic path (a path not passing twice by the same vertice) connecting, with transitivity, two different nodes not having a direct connection. When defining a trust path, we have consider two features to produce the most accurate results for trust inference computation: Time To Live parameter (TTL) and the Minimum Trust Threshold (MTT). In the following, we formally define a trust path and we describe how path length and trust values on paths affect the computations, and how these features are considered into our TPS algorithm.

Definition 3 *(*TRUST PATH*) A trust path $\mathcal{P}(o, s)$ from the origin o leading to the sink s is a triplet $\mathcal{P}(o, s) = (V, E, T)$. V is the set of users $(v_0, v_1, \ldots, v_n)$ where $v_0 = o$, v_n is the direct predecessor of s and $n \leq$ TTL. $E \subseteq V \times V$ is the set of directed links between users labelled with the set of information T. The latter indicates whether, and possibly how much $v_i \in V$ considers, $v_{i+1} \in V$ trustworthy, where $i \in [0, n-1]$, such that each $t \in T \geq$ MTT.*

To further illustrate this definition, let us take a TTL = 4 and a MTT = 0.4. The path $A \rightarrow D \rightarrow G$ between nodes A and G is considered as a trust path since each direct trust t_i within this path is greater than 0.4 and its length is less than 4.

In principle, a limit on the depth of the search should lead to more accurate results. In fact, in [Golbeck, 2005], the authors showed that the average error increases as far as the depth does. This intuitively makes sense: getting information from only one intermediate person should usually be more reliable than information passed down a long chain of users. However, the trade-off is that imposing a fixed limit might not satisfy the source. To balance these factors, we propose that the source node, according to its behaviour, sets the Time To Live parameter (TTL).

The authors, in [Ziegler et Golbeck, 2007], also highlight that the most accurate information will come from the most highly trust neighbours. Thus, the source sets a Minimum Trust Threshold (MTT) that represents the largest trust value that can be used as a minimum threshold such that a path can be found from source to sink neighbours. In this manner, we only require to consider paths where all edges have trust values at least equal to the MTT and ignore paths having lower values.

Both TTL and MTT parameters depend on the level of trust of the source; if this latter is interested in being very confident, then it would increase the MTT and decreases the TTL, and vice versa.

6.2.1 The TPS Algorithm

In what follows, we introduce a new algorithm, called TPS, for trust paths' searching, that looks for all possible trust paths, as defined above. The pseudo-code of TPS is sketched by Algorithm 7. This algorithm takes as input the truster (source node o), the maximum length requested (TTL), the minimum threshold

Algorithm 7: TPS

Data: o: Source or the origin node
s: Sink node
MTT: Minimum trust threshold
TTL: Time to leave
t: Direct trust matrix

Result: $PATHS$: Set of Trust Paths from o to neighbors of s.

```
1  begin
2      path ← o;
3      Q ← [s, TTL, MTT, path];
4      P ← ∅;
5      TrustPath (t, Q, P);
6      PATHS ⟵ P;
```

Algorithm 8: TrustPath (t, Q, P)

```
1  begin
2      ADJ ← adjacent(t, getLast(path), MTT);
3      foreach adj ∈ ADJ do
4          if adj ∉ adjacent(t, s) then
5              if TTL != 0 then
6                  TTL − −;
7                  add(adj, path);
8                  Q ← [path, TTL, MTT, s];
9                  TrustPath (t, Q, P);
10         else
11             add(adj, path);
12             P = P ∪ path;
```

requested (MTT), the trustee node (sink s) and the direct trust matrix t that contains direct trust values between the social network nodes.

The source starts a search for the sink. In line 3, the source, in its initial query Q, adds itself to be the first node of the trust path and specifies the TTL, the MTT and the sink s. In line 5, the algorithm invokes the TrustPath function

Trust path	$t_{n \to s}$	ttl	mtt	Arbitrator n
$A \to B \to E$	0.5	5	0.35	E
$A \to B \to F \to H$	0.6	4	0.35	H
$A \to C \to F \to H$	0.6	4	0.35	H
$A \to C \to G \to I$	0.4	4	0.35	I

Table 6.1: The different possible responses for A's query $Q = [A, 7, 0.35, J]$ fulfilling the MTT and TTL criteria

that is sketched by Algorithm 8. This routine invokes all trust paths between the origin and the sink. In line 2, it selects the neighbours of the current node (the last node of the path under construction), taking into account MTT. From lines 3 to 12, the function iteratively searches for nodes fulfilling both MTT and TTL criteria.

The current node sends the query towards each of its neighbours fulfilling the MTT criterion to obtain their ratings about the sink. If the neighbour node *adj*, receiving the query, does not have a direct rating for the sink, i.e., it is not a sink's neighbour (line 4), then it updates the query while decreasing the TTL and adding itself as the last node of the path. Each neighbour repeats this process until upsetting the TTL value or being an adjacent to the sink.

If the neighbour *adj* is a final arbitrator (i.e., the last trust path node) and has a direct rating of the sink, then the respective constructed trust path is added to the set P as mentioned in lines 11 and 12.

Let us again consider the OSN illustrated in Fig. 6.1. Suppose that Alice (A for short) searches an answer to her request "Can I trust John (J) or not?". First of all, A has to initialize her query Q, we take the example of $Q = [A, 7, 0.35, J]$; then she executes the TPS algorithm to receive the query responses. Table 6.1 details the different responses for the A's request. In fact, we notice that there are four responses and, consequently, four different trust paths among six possible ones. For example, since the direct trust between nodes D and G does not meet the MTT criterion ($t_{D \to G} < 0.35$), the algorithm does not retain the path $A \to D \to G \to I$.

Once the set $PATHS$ of constructed trust paths is returned by TPS to the source node, the latter computes the inferred trust to the sink. In the following,

we explain how to compute this value.

6.3 TIM: Trust Inference Measure

We propose two complementary operations for computing inferred trust value: (*i*) trust propagation from *source* to the direct neighbour of *sink* through a trust path; (*ii*) trust aggregation from the *source* to the *sink*. Worth of mention, if there is no path between two nodes, then the inferred trust value is equal to 0.

6.3.1 Trust propagation

We propose a propagation function that computes the strength s_p of each path p. As shown in Eq. 6.1, we consider three features to produce the most accurate results.

$$s_p = \alpha \bar{t}_p + \beta(1 - v_p) + \gamma w_p \tag{6.1}$$

- **PathAverage** : $\bar{t}_p$ denotes the average of direct trust values, t_i, between the n path nodes as shown in Eq. 6.2. This feature is defined as an assessment of the trust ratings between the intermediary nodes in a chain of acquaintances, standing on the assumption that intermediary friends with high value of trust rating increase the accuracy of trust computation. If we have trusted friends in a path, then we have more confidence in the evaluated trust rating in that path. Hence, the strength of a path directly depends on trust rating among friends.

$$\bar{t}_p = \frac{1}{n} \sum_{i=1}^{n} t_i \tag{6.2}$$

- **PathVariance** : v_p denotes the path variance to measure the amount of deviation between the path average trust $\bar{t}_p$ and the n trust values t_i of that path (see Eq. 6.3). The assumption is that intermediary friends with close values of trust rating guarantee a balance between trust values and hence better assess the confidence degree on that path. A high value of PathVariance decreases the confidence on that path, whereas a low value increases that confidence.

$$v_p = \frac{1}{|n|} \sum_{i=1}^{|n|} (t_i - \bar{t}_p)^2 \tag{6.3}$$

- **PathWeight** : As shown in Eq. 6.4, w_p denotes the path weight computed as the fraction between the shortest trust path length ($|n'|$) and that of the current path ($|n|$). The rationality behind the PathWeight is that the limit on the depth from a source to a sink should lead to more accurate results. So, the shorter the path is, the more accurate the evaluated trust rating value from a source to a sink is.

$$w_p = \frac{|n'|}{|n|} \tag{6.4}$$

The path considered "perfect", having a strength $s_p = 1$, is obtained under these conditions: *(i)* its trust values are worth ($\bar{t}_p = 1$); *(ii)* its accurate balance is reached ($v_p = 0$); and *(iii)* it is the shortest trust path ($w_p = 1$).

In every path, only the last intermediary node knows the sink directly, i.e., it has a direct connection with the sink. In fact, TPS does not consider the direct opinion between the last trust path node a_{n_p} (the final arbitrator) and the sink to discover a trust path p. Thus, this opinion is not considered when measuring the path's strength s_p. Being so, it does not only avoid a bias in the final direct opinion $t_{a_{n_p} \to s}$, but also s_p has a simple interpretation as being the value of trust of the final recommendation, i.e., the amount of confidence that the source has in the suggested trust rating ($t_{a_{n_p} \to s}$) depends on the trust ratings of intermediary nodes (s_p). The stronger the path p is, the more acceptable by the source the last direct opinion $t_{a_{n_p} \to s}$ with a high confidence is. So, the inferred trust value associated to each path p of length n is computed by multiplying the direct trust value given to the sink ($t_{a_{n_p} \to s}$) by the strength of p (s_p).

$$t_{o \to s} = s_p \times t_{a_{n_p} \to s}. \tag{6.5}$$

Parameters' Estimation

We use parameters α, β and γ to assess effects of respectively PathAverage, PathVariance and PathWeight on the accuracy value (cf., Eq6.1). We show now

how to compute the optimal values of α, β and γ, in order to have more accurate results and to minimize the impact of experimental errors, i.e., to minimize the error of the difference between the computed trust values through our algorithm and the real trust values. These real trust values are obtained from a real dataset for fair comparison. If there is a direct trust value between two nodes (say a source o and a sink s), then this value is masked and we compute the indirect trust value between them. So, we aim to satisfy Eq. 6.6, and consecutively Eq. 6.7 and Eq. 6.8, where $i \in [1, m]$ and m is the number of direct trust values $r_{i_{o\to s}}$ in the dataset, $t_{i_{o\to s}}$ represents the computed trust values, such that there exists at least one path p_i of size > 1 between each couple (o, s).

$$t_{i_{o\to s}} = r_{i_{o\to s}} \tag{6.6}$$

$$s_{p_i} \times t_{a_{n_{p_i}}\to s} = r_{i_{o\to s}} \tag{6.7}$$

$$(\alpha\bar{t}_i + \beta(1 - v_i) + \gamma w_i) \times t_{a_{n_i}\to s} = r_{i_{o\to s}} \tag{6.8}$$

Eq. 6.8 is solved for $i = [1, m]$ using a linear system of m linear equations with 3 unknown parameters. Then, we can write it using a matrix notation. We define $C_{(m,3)}$ to be the matrix where $C[i, 1]$, $C[i, 2]$ and $C[i, 3]$ contain, respectively, the values $\bar{t}_i \times t_{a_{n_{p_i}}\to s}$; $(1 - v_i) \times t_{a_{n_{p_i}}\to s}$ and $w_i \times t_{a_{n_{p_i}}\to s}$. We define $\vec{b_3}$ as the vector containing the parameters α, β and γ. Finally, we define $\vec{r_m}$ as the vector containing the actual trust values $r_{i_{o\to s}}$.

To solve this system of m equations, we use the method of least squares [1] since $m > 3 : \vec{b_3} = (C^T_{(3,m)}C_{(m,3)})^{-1}C^T_{(3,m)}\vec{r_m}$. In Section 6.5, we show the impact of the optimised values of parameters α, β and γ on our experiments.

"Method's Properties"

Our trust propagation method should fulfil some key properties in order to infer trust values which are coherent with the direct trust values in the trust graph. These properties and the ways that our method fulfils them are described in the remainder.

[1] http://en.wikipedia.org/wiki/Least_squares

Property 1 *The evaluated inferred trust rating between a source o and a sink s in a path p cannot be greater than the direct trust given to s by its last arbitrator a_n from the same path.*

It is clear that, in our approach, we have always the inferred trust value $t_{o\to s}$ less than or equal to $t_{a_n\to s}$. Indeed, in the perfect case from Eq. 6.5, for an optimal path's strength $s_p = 1$, we have $t_{o\to s} = t_{a_n\to s}$.

Property 2 *If there is a unique path between the origin and the sink, or there are several paths between them having all the same length, then the path weight feature, w_p, which is always in that case equal to 1, must not be considered for the trust value computation. In fact, in such cases, the path length is not a discriminatory factor, i.e., it does not give an effective information about the strength of a path.*

Property 3 *If there is a unique direct value in the trust path, that is its length is 1, then the path variance feature, v_p, which is always in that case equal to 0, must not be considered in the trust value computation. Indeed, considering it would uselessly increase the inferred trust value.*

Property 4 *The inferred trust between a source and a sink cannot be greater than the direct trust, $t_{o\to a_1}$, given by o to its first arbitrator a_1 as showing by Eq. 6.9.*

$$t_{o\to s} = Min\left\{t_{o\to a_1}, s_p \times t_{a_{n_p}\to s}\right\}. \tag{6.9}$$

6.3.2 Trust aggregation

Different versions of aggregation have been suggested which differ in their final results and mathematical properties. To answer how can we aggregate trust rating from each path and decide which path is the most trusted one, we use some of the well known aggregation methods [Lesani et Montazeri, 2009].

- **The Mean Aggregation Method**: Assume that a person P has n different information sources for the value of an unknown continuous quantity X. P should aggregate the values that are obtained from these sources. If

each source i reports value B_i as his believed value for X and P trusts each source i with a crisp trust value T_i, then the result of aggregation B_P, i.e., the value that P believes for X, is given by Eq. 6.10.

$$B_P = \frac{\sum_{i=1}^{n} T_i \times B_i}{n} \tag{6.10}$$

The preceding aggregation method can be applied to trust aggregation. Assume that a source o has to aggregate different trust values to a sink that are obtained from its neighbours in paths contained in the set $PATHS$ returned by the TPS algorithm. If the respective strengths of the different trust paths between the origin node o and the sink node are s_{p_1}, s_{p_2}, ..., s_{p_n}, and trust values from different sink's neighbours to the sink are $t_{a_{p_1} \to sink}$, $t_{a_{p_2} \to sink}$, ..., $t_{a_{p_n} \to sink}$, respectively, then the trust of node o to the sink, as the result of the Mean aggregation, is given by Eq. 6.11.

$$t_{o \to sink} = \frac{\sum_{i=1}^{|PATHS|} s_{p_i} \times t_{a_{p_i} \to sink}}{|PATHS|} \tag{6.11}$$

- **The Mult Aggregation Method**: The Mult aggregation method is similar to the mean aggregation method, computing B_P and $t_{o \to sink}$ as given, respectively, by Eq. 6.12 and Eq. 6.13.

$$B_P = \prod_{i=1}^{n} T_i \times B_i \tag{6.12}$$

$$t_{o \to sink} = \prod_{i=1}^{|PATHS|} s_{p_i \times t_{a_{p_i} \to sink}} \tag{6.13}$$

- **The Min Aggregation Method**: Consider two statements S_1 and S_2, for which the truth values are μ_{S_1} and μ_{S_2}, respectively (μ_{S_1} and $\mu_{S_2} \in [0, 1]$). It is shown in [Lesani et Montazeri, 2009] that the truth value of the logical "AND" connector is given by the *min* function (Eq. 6.14).

$$\mu(S_1 \wedge S_2) = min(\mu_{S_1},\ \mu_{S_2}) \tag{6.14}$$

The *min* aggregation method can be applied for trust aggregation. Assume that a source o has to aggregate different trust values to a sink that are obtained from its neighbours. If the respective strength values of the different trust paths between the origin node o and the sink node are s_{p_1}, s_{p_2}, ..., s_{p_n}, and trust values from different sink's neighbours to the sink are $t_{a_{p_1}\rightarrow sink}$, $t_{a_{p_2}\rightarrow sink}$, ..., $t_{a_{p_n}\rightarrow sink}$, respectively, then the trust of node o to the sink as the result of the aggregation is given by Eq. 6.15.

$$t_{o\rightarrow sink} = s_{p_k} \times t_{a_{p_k}\rightarrow sink} with\ s_{p_k} = min\{s_{p_i}\}|i \in [1, |PATHS|] \tag{6.15}$$

- **The Max Aggregation Method**: The Max aggregation method using the truth value of the logical "OR" connector (Eq. 6.16) is similar to the Min aggregation method, and computes $t_{o\rightarrow sink}$ using Eq. 6.17.

$$\mu(S_1 \vee S_2) = max(\mu_{S_1},\ \mu_{S_2}) \tag{6.16}$$

$$t_{o\rightarrow sink} = s_k \times t_{a_{p_k}\rightarrow sink} with\ s_{p_k} = max\{s_{p_i}\}|i \in [1, |PATHS|] \tag{6.17}$$

During our experiments, we considered these four methods to compute trust and we show which one gives the best accuracy results.

6.4 Comparison with other algorithms

We applied the developed **TIM** algorithms on different sample trust networks. The best reasonable results are obtained with the Max aggregation method, as explained in Section 6.5. Thus, we consider now that TISoN relies on this method to infer trust values.

	A	B	C	D	E	F	G	H	I	J	K	L
A	-	5.00	7.50	10.00	**3.75**	3.75	**2.50**	2.50	4.34	**7.50**	2.32	7.25
B	6.09	-	6.25	7.50	7.50	0.00	7.50	2.18	3.75	3.75	2.27	7.50
C	6.25	4.37	-	8.75	6.37	5.00	**10.00**	2.26	4.25	10.00	2.50	7.50
D	7.50	3.95	5.93	-	6.01	4.09	2.50	2.50	4.22	2.50	1.56	6.56
E	3.75	4.06	6.09	5.00	-	4.16	5.00	2.18	5.00	5.00	2.26	5.24
F	3.75	4.06	6.09	5.00	5.93	-	5.00	2.18	5.00	7.50	2.26	5.62
G	2.50	4.06	6.09	**2.50**	5.93	4.16	-	2.18	4.24	10.00	2.50	7.50
H	5.00	4.37	6.56	5.00	6.25	4.37	2.29	-	4.40	0.00	5.00	7.02
I	7.50	3.75	5.62	7.50	6.09	4.37	2.34	2.34	-	7.50	2.34	10.00
J	-	-	-	-	-	-	-	-	-	-	-	7.50
K	10.00	5.00	7.50	10.00	6.56	4.68	2.50	2.50	4.51	9.67	-	7.31
L	-	-	-	-	-	-	-	-	-	-	-	-

Table 6.2: Results of TISoN for the trust network in Fig. 3.4.

We compare TISoN *vs* respectively TidalTrust [Golbeck, 2005], RN-Trust [Taherian *et al.*, 2008] and SW-Trust [Jiang *et al.*, 2014] algorithms. In the sake of a fair comparison, TISoN is applied in the same illustrative OSN given in Fig. 3.4, and we discuss which results are more reasonable with regard to properties of trust in real life OSNs.

Table 6.2 shows the results of applying TISoN on the trust network depicted in Fig. 3.4. Notice that we consider a TTL = 10 and a MTT = 0 to estimate TISoN performances in the worst cases.

Case 1: Beginning by the trust value from A to E and G to D given by TISoN are 3.75 and 2.5 respectively; these values are almost equal to values returned by both RN-Trust and SW-Trust algorithms (3.7 and 2.5 resp.), and more reasonable than the values computed by TidalTrust (7.5 and 10 resp.).

Case 2: Our outputted trust value from nodes A to J is 7.5. With respect to values provided by both TidalTrust and RN-Trust (10 and 8.1 resp.), 7.5 appears as more reasonable value. In fact, 10 and 8.1 are both greater than all the trust values existing in the path between A and J.

Case 3: TISoN and TidalTrust give back a trust value from nodes A to G equal to 2.5 which we consider reasonable, while RN-Trust returns 7.9.

Case 4: All of TISoN, RN-Trust and TidalTrust output 10 as indirect trust value from nodes C to G; except for SW-Trust that outputs 0.45. In fact, with TISoN, the strength of the paths is computed; moreover, considering the strongest path and applying the Max method makes the final trust value become higher.

An OSN's user needs to receive the most reasonable values. This criterion (to be reasonable) depends on the behavior of the user himself. Thus, an algorithm for trust inference has, not only, to fulfil the properties of trust in social networks, but also to comply the behaviour of the truster that we translate by TTL and MTT parameters.

6.5 Experimental evaluation

In this Section, the experimentation of TISoN algorithms is reported and a comparison with other existing algorithms is carried out. In the first serie of our experimentation, we analyse the performance and scalability of TISoN. Due to the lack of a real trust OSN dataset with a large number of users (existing datasets have at most 1 or 2 hundreds of users), we use a random data set generator producing until one million of users.

In the second serie of the experiments, we aim to validate the effectiveness of TISON and to test the inferred trust quality. Since trust is not randomly distributed, we conduct extensive experiments in the real OSN dataset Advogato.

6.5.1 Scaling up satisfaction

Each time a new user consults the social network, leaves it, or makes new relationships, we need to update our trust matrix. Besides, in an OSN, the number of users is rather high and the more it increases, the more the direct trust matrix size important is. Thus, managing and updating this index will be more and more costly, and will lead to scalability problems. Yet, if the designed application does not fulfill the scaling up requirements, then it could not be of use on the internet scale which is the purpose of OSN, and its life time will be limited.

In the following, we compare TISoN to TidalTrust, RN-Trust and SWTrust algorithms according to their running time. The aim of this comparison between the different algorithms is to show which algorithm performs reduced execution

time and is more scalable.

The performance evaluation can be performed through reality (using real OSNs) or by simulation. In this work, we chose to use simulation because of the lack of a real OSN dataset with a large number of users. Thus, to analyse the performance of the algorithms with respect to the number of users, we create a random dataset generated by a program written in java language, that respects the following format:

```
userA  userB  directTrustValue
```

For instance, a line <12 9230 0.7> indicates that the user 12 trusts the user 9230 with a trust degree equal to 0.7. Moreover, the dataset uses two parameters which highlight how the algorithms can scale:

- Number of users: this is the most obvious choice as a test parameter, since the number of users is at the core of the problems that arise as far as one is interested in developing and simulating an algorithm needing to be scalable.

- Out-degree: this is an important issue since it directly affects the size of the network.

The program creating the dataset, produces until one million users overlay network with the parameters listed above.

Fig. 6.2 shows the produced topology and highlights the correlation between the number of users and the degree of the network.

Moreover, the program offers the possibility to configure the TTL and MTT simulation parameters. For our case and that of SWTrust, different scenarii may exist. For this reason, we use specialized scenarii presented in Table 6.3. These scenarii depend on the level of trust of the requester; if this latter is interested in being very confident, (s)he increases the MTT and decreases the TTL, and vice versa. In addition, since the SWTrust algorithm is interested only on the length of the path, we vary in particular the TTL value. In the worst cases (described with the scenarii TS4 and SS4), we consider that the MTT is equal to 0 and the TTL is equal to L (length of the longest path). Running the programs with TS4 and SS4 scenarii, implies that TISoN and SWTrust do not consider trust paths

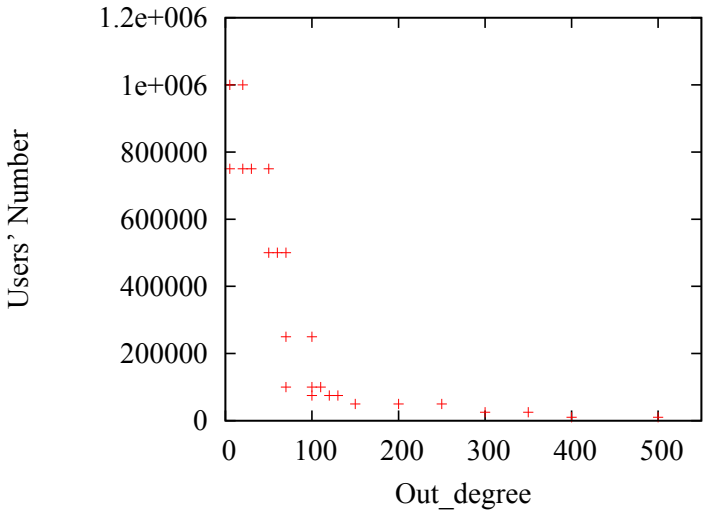


Figure 6.2: Topology and out-degree distribution

but all paths between the source and the sink.

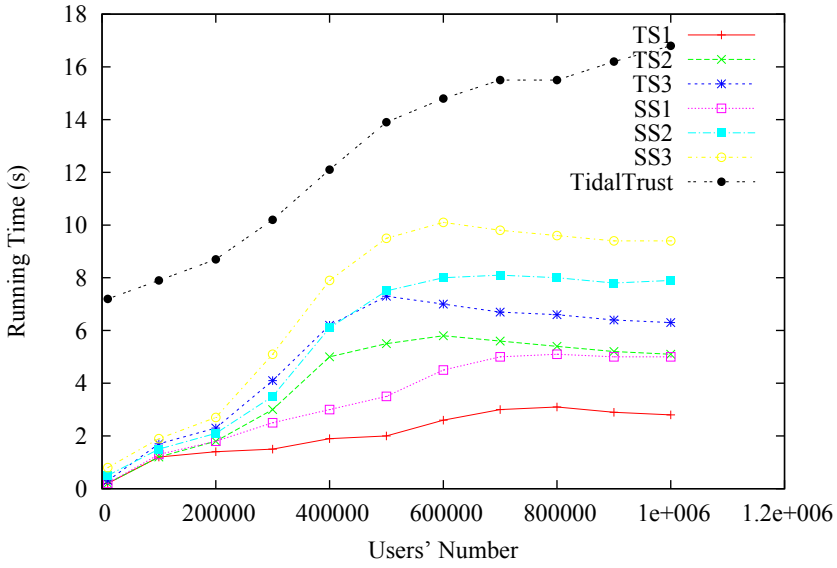


Figure 6.3: Run Time varying the users' number

We process the dataset and test the algorithms by varying the number of users. The obtained results of TISoN and SWTrust algorithms with the respective scenarii: TS1, TS2, TS3, SS1, SS2 and SS3, as well as these of TidalTrust are shown in Fig. 6.3. However, the average running times of TISoN and SWTrust with the respective scenarii: TS4, SS4, as well as that of the RN-

	Confidence degree	ttl	mtt
TISoN-Scenario1 (TS1)	High	3	0.8
TISoN-Scenario2 (TS2)	Medium	5	0.5
TISoN-Scenario3 (TS3)	Low	8	0.3
TISoN-Scenario4 (TS4)	-	L	0.0
SWTrust-Scenario1 (SS1)	High	3	0.0
SWTrust-Scenario2 (SS2)	Medium	5	0.0
SWTrust-Scenario3 (SS3)	Low	8	0.0
SWTrust-Scenario4 (SS4)	-	L	0.0

Table 6.3: The different scenarii

Trust algorithm, are shown in Table 6.4 for better vision.

Indeed, it is clear that TISoN and SWTrust give best execution times when fixing ttl and mtt values. We deduce that the parameters ttl and mtt impact on the runtime and provide better performance results. In fact, we notice that the value of the running time increases insignificantly with the variation of the ttl and mtt even if the number of users reaches one million. Moreover, by fixing the ttl and mtt values, the number of trust paths between the source and the sink decreases and the search ends quickly. More specifically, in the case of high mtt and low ttl values (TS1), the runtime reaches its better values. We can also see that TidalTrust is quicker than RN-Trust, TISoN and SWTrust with the scenarii TS4 and SS4. In fact, the search of the shortest trust path by TidalTrust increases its running time, but once it is found, the search considers only shortest paths. However, for the case of the RN-Trust algorithm as well as TISoN and SWTrust with the scenarii TS4 and SS4, the runtime increases significantly since they search for all the paths between the source and the sink. Indeed, the number of paths increases as far as the number of users increases.

We mention that we test the scenario TS4 to evaluate TISoN in its worst case. In fact, a TTL < 8 is sufficient to guarantee an inferred trust value with high accuracies as we will show in Subsection 6.5.3. Moreover, the solution that we propose, to support the scaling up factor with the scenario TS4, is to search for all the possible paths between all possible nodes in an offline way. Thus, once online, we only compute the indirect trust value between the truster and the trustee to

avoid a long time the truster spends on waiting the response.

Number of users	10000	100000	300000	500000	700000	900000	1000000
RN-Trust	60	300	720	1500	3000	5520	7680
TS4	30	126	356	722	1220	2100	3260
SS4	35	140	350	728	1340	2104	3270

Table 6.4: Average running times (s) of RN-Trust as well as worst scenarii of TISoN and SWTrust

6.5.2 The Advogato Dataset

Most of works interested in trust management in OSNs (e.g. [Jiang *et al.*, 2014], [Massa et Avesani, 2005], [Shekarpour et Katebi, 2010]), conduct their experiments using the Epinions dataset [2]. On the available data of Epinions.com, values are just 1 and 0 and not real value standing within the interval $[0, 1]$. Intermediate values such as 0.7 are not expressible on Epinions.com. Due to this constraint, we looked for another dataset to perform our experiments.

We use the trust metric from the real dataset of the online community and social networking site Advogato [3]. This dataset contains more than 14000 users and 55000 relations. Advogato utilizes a social graph representing Advogato's members and their relationships. Each node in the graph represents a user's account, and a directed edge indicates a certificate [Al-Oufi *et al.*, 2012]. An Advogato's user can certify other users on 4 different levels: Observer, Apprentice, Journeyer, and Master. The Advogato trust metric uses this information in order to assign to every user a certification level. An almost original characteristic of the Advogato dataset stands in the fact that weights of directed edges between nodes specified on these 4 levels (Observer, Apprentice, Journeyer, and Master) can be mapped. Many mapping solutions of these levels are proposed in the literature [Massa et Souren, 2008]. We use the method presented in [Massa et Souren, 2008] which assigns a trust value to each level within the unit interval (Observer=0.4, Apprentice=0.6, Journeyer=0.8, and Master = 1.0), while making a little modification as shown in Fig. 6.4.

[2]This dataset is available at this address: http://www.trustlet.org/wiki/Epinions_datasets
[3]This dataset is available at this address: http://www.advogato.org

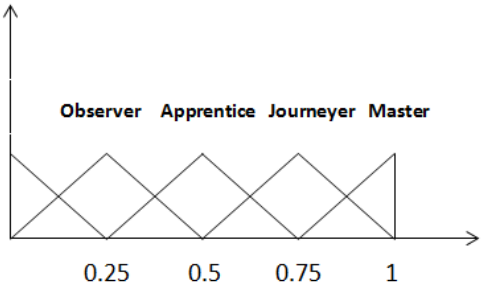


Figure 6.4: Mapping of trust linguistic terms.

6.5.3 Accuracy Metrics

In order to assess the performance of our algorithm, we use a standard evaluation technique from the machine learning field: leave-one-out [Jiang et Wang, 2011]. If there is an edge between two nodes (say *source* and *sink*), that edge is masked and trust is computed through the trusted graph from *source* to *sink*; then we compare the computed value with the masked one.

To test the accuracy, we adapt the accuracy metrics in [Jiang et Wang, 2011] and [Shekarpour et Katebi, 2010] including absolute error, precision, recall, and Fscore, which are recalled in the following.

Absolute Error

The absolute error is the difference between the real (actual) value of trust and that calculated by the proposed method.

$$Absolute\ error = |calculatedTrust - realTrust| \tag{6.18}$$

Precision, Recall and Fscore

The accuracy represents the ability of predicting a user to be trusted or not. Based on the defined criterion for accuracy, making a right decision is the ultimate metric for comparison. We use precision and recall metrics to compare the accuracy of methods in making the trust decision. Parameters used to compute the accuracy are as follows:

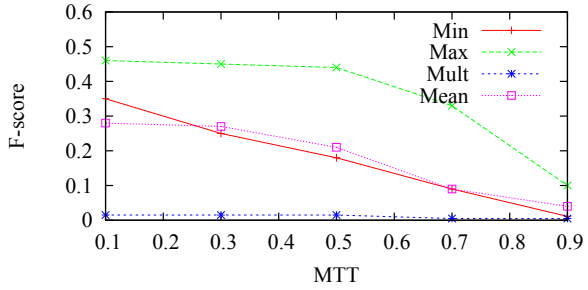
X = the set of users that a user actually trusts;

Y = the set of users that our algorithm suggests to trust.

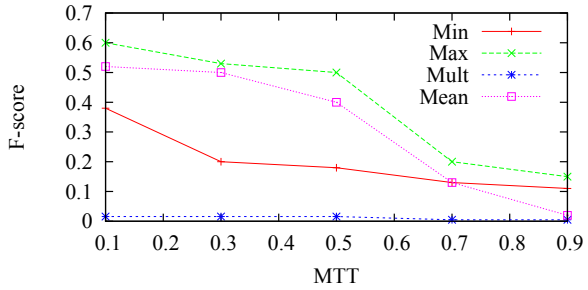
$$Precision = \frac{|X \cap Y|}{|Y|}; Recall = \frac{|X \cap Y|}{|X|} \tag{6.19}$$

We use the Fscore metric to measure the accuracy using recall and precision jointly.

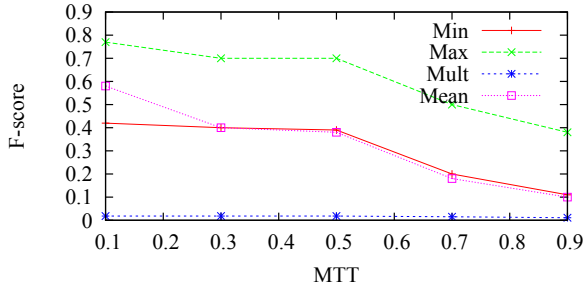
$$Fscore = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{6.20}$$



(a) Fscore for 1000 users

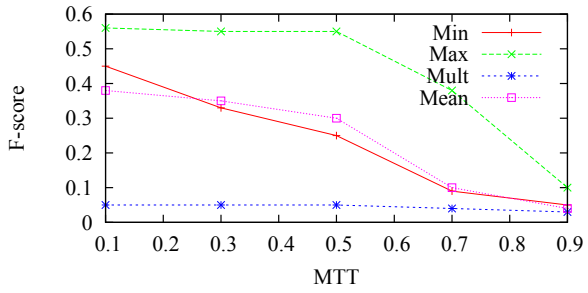


(b) Fscore for 8000 users

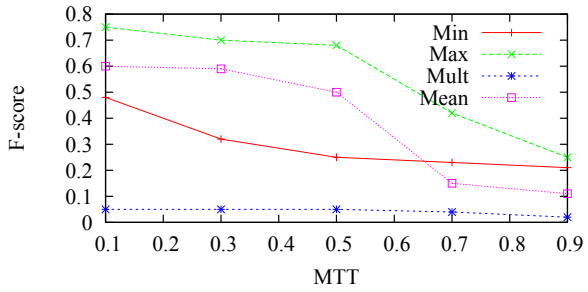


(c) Fscore for 14000 users

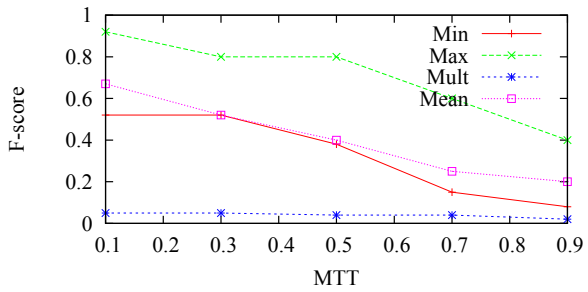
Figure 6.5: Comparison of Fscore for different thresholds before optimisation



(a) Fscore for 1000 users



(b) Fscore for 8000 users



(c) Fscore for 14000 users

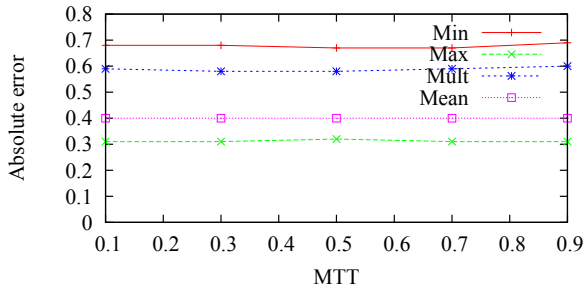
Figure 6.6: Comparison of Fscore for different thresholds after optimisation

6.5.4 Results for accuracy metrics

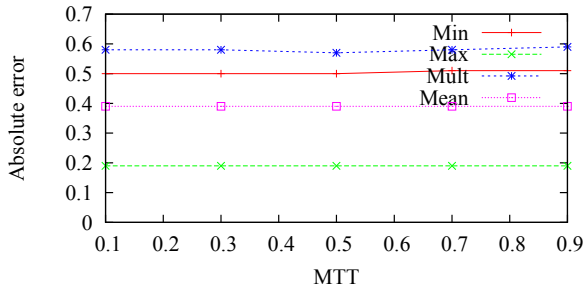
For the experiments about accuracy metrics, we process the Advogato dataset and run the programs for different MTT and TTL values. We present as well the behaviour of these parameters w. r. t. accuracy metrics while varying the users' number.

Effect of MTT variation

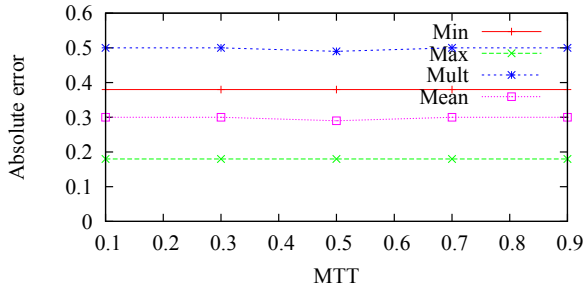
Figures 6.5, 6.6, 6.7 and 6.8 respectively show the variation of the Fscore and the Absolute error before and after optimising the values of α, β and γ *vs* the variation



(a) Absolute error for 1000 users



(b) Absolute error for 8000 users



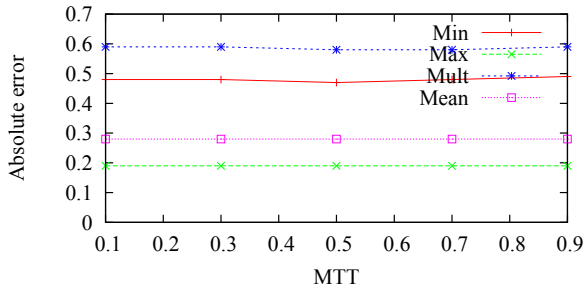
(c) Absolute error for 14000 users

Figure 6.7: Comparison of Absolute error for different thresholds before optimisation

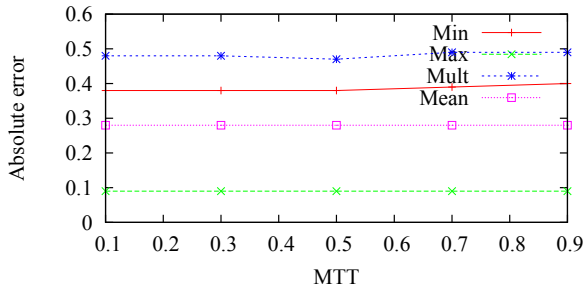
of the minimum trust threshold, MTT. In this first part of experiments, we set the TTL value to 10, in the sake of assessing the effect of the MTT parameter on our results.

On the one hand, Fig. 6.5 and Fig. 6.6 respectively show a comparison of the Fscore metric before and after the optimization for different users' number (1000, 8000 and 14000). It is clear that the obtained results in Fig. 6.6 are better than those in Fig. 6.5. In fact, optimising the α, β and γ values using the least squares' method has a positive impact and improves the Fscore values.

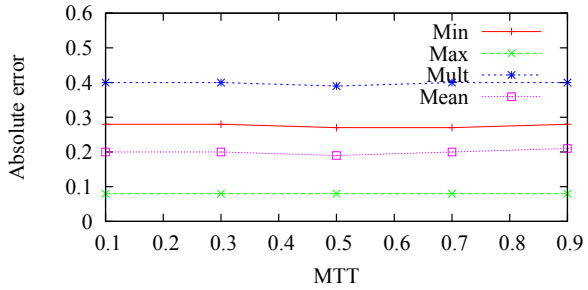
In addition, we remark an increase of the Fscore for all the methods as far as



(a) Absolute error for 1000 users



(b) Absolute error for 8000 users



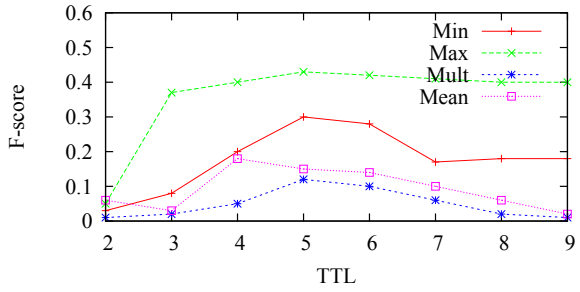
(c) Absolute error for 14000 users

Figure 6.8: Comparison of Absolute error for different thresholds after optimisation

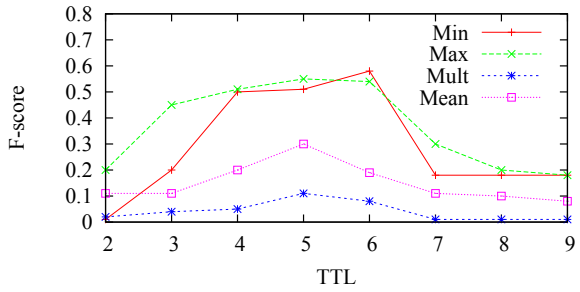
the users' number increases. Besides, when the MTT varies from 0.1 to 0.9, we remark that the FScore decreases for all the considered methods, especially when the MTT$>$ 0.5, it sharply decreases. This indicates that MTT should not be set to a too large value, since a smaller number of trust paths is obtained (a high number of direct trust values are discarded), yielding to a loss of information and thus to worsen the accuracy of the results. Moreover, to avoid taking into consideration the opinion of malicious users, small values for trustworthy decision making are not desired. This means that the MTT should not be set to a too small value.

On the other hand, Fig. 6.7 and Fig. 6.8 show a comparison of the Absolute

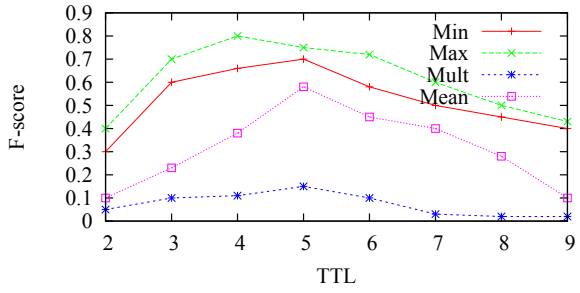
error before and after optimisation resp. for different users' number, 1000, 8000 and 14000. At a glance, best results are obtained after our optimisation method. Also, we have better values as far as the users' number increases becoming irrelevant with 14000 users. Nevertheless, as far as the MTT varies, the Absolute error does not vary for the Max aggregation method and varies weakly for the others.



(a) Fscore for 1000 users

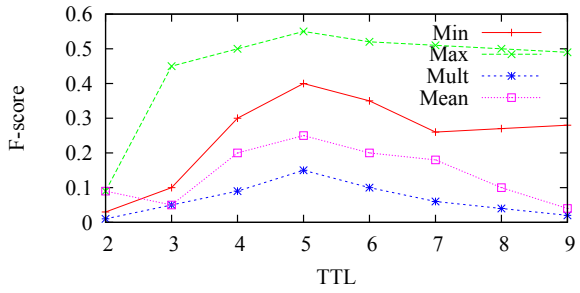


(b) Fscore for 8000 users

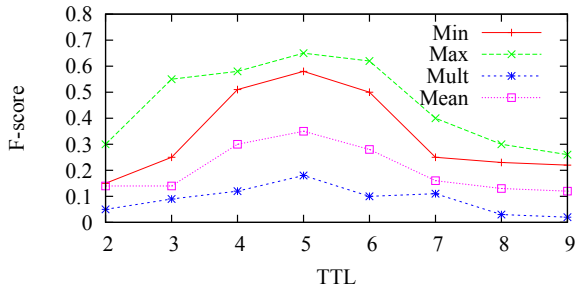


(c) Fscore for 14000 users

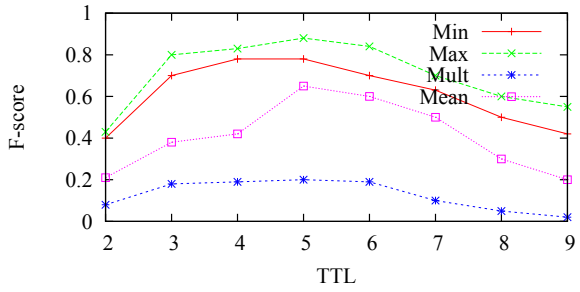
Figure 6.9: Comparison of Fscore for different transitivity hops before optimisation



(a) Fscore for 1000 users



(b) Fscore for 8000 users



(c) Fscore for 14000 users

Figure 6.10: Comparison of Fscore for different transitivity hops after optimisation

Effect of TTL variation

The second part of our experiments is carried out to show the variation of the FScore and the Absolute error vs the variation of the TTL. Notice that we set the MTT value to 0.

From the experimental results plotted in Figures 6.9, 6.10, 6.11 and 6.12, we remark the effect of the optimisation method on the improvement of the accuracy results. We also remark an increase of accuracy for all the methods as far as the users' number increases. Moreover, we can see the impact of the TTL on the

performance of our approach.

At first glance, in Figures 6.10 and 6.12, the anemic values of the accuracies, when the TTL is less than 4, could be surprising. Indeed, expectedly, the shorter the path, the greater the accuracy is. However, in fact, when the path length is limited to a small TTL value, then often there is no trust path between the source and the sink. For higher values of TTL, standing with the range $[4, 6]$, the accuracy reaches its best values, decreasing again when a TTL value exceeding 7. These results indicate that the TTL should not be set to a too small value, since we risk to overlook trust paths and thus the source will receive an empty answer for its request. Moreover, to avoid the information loss and consequently worsen the accuracy of the results, large values for the TTL are not desired.

Conclusion

From all these tests, we conclude that reaching the best accuracies is proportional to the increase of the users' number. This leverages the scalability need when dealing with real OSNs. Moreover, we can conclude that, whenever we are interested in just predicting to trust or to distrust, then a MTT equal to 0.5 with a TTL equal to 5 would be the best choice for this dataset.

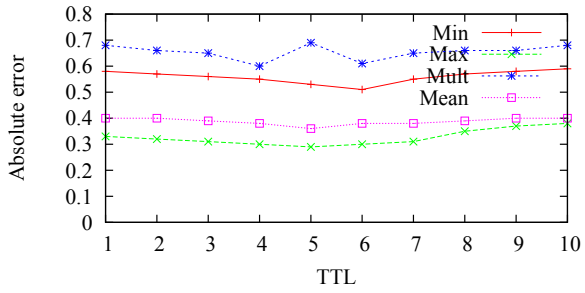
6.5.5 Results for Aggregation method

Regardless the users' number and the TTL and MTT values, the accuracies are relatively high whenever the Max method is used to perform aggregation. Choosing the Max method means choosing, among the set $PATHS$ of trust paths, the Most Trustable Path (MTP) which is the strongest path.

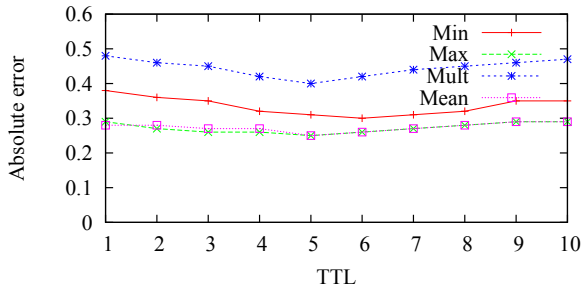
$$s_{\text{MTP}} = max\left\{s_{p_i}\right\}, \forall p_i \in PATHS. \tag{6.21}$$

To further illustrate, we conduct a third part of our experiments where we are only interested in the *Max* method. In fact, this method provides always the best results.

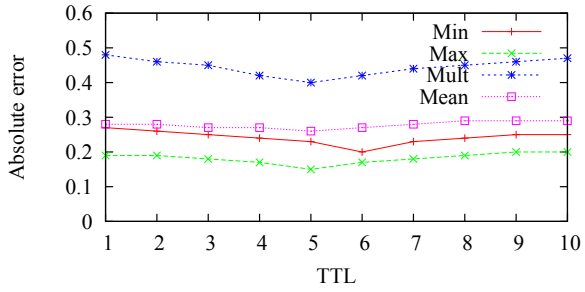
In the remainder, we aim to show the impact of combining MTT and TTL on the FScore results. For that, we define a user's behaviour parameter, λ, that can be specified by participants based on their own trust evaluation criteria (TTL and MTT). λ mixes both of the TTL and MTT which are critical factors to ensure the computed trust quality, with $\text{TTL} = \lambda$ and $\text{MTT} = 1 - \lambda/10$. We are



(a) Absolute error for 1000 users



(b) Absolute error for 8000 users



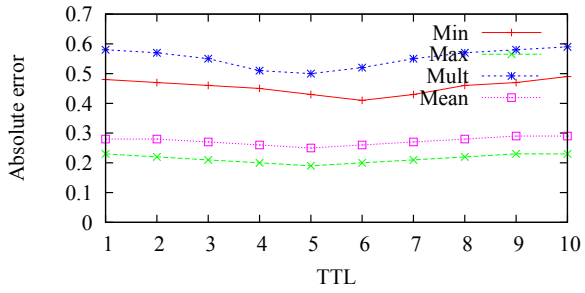
(c) Absolute error for 14000 users

Figure 6.11: Comparison of Absolute error for different transitivity hops before optimisation

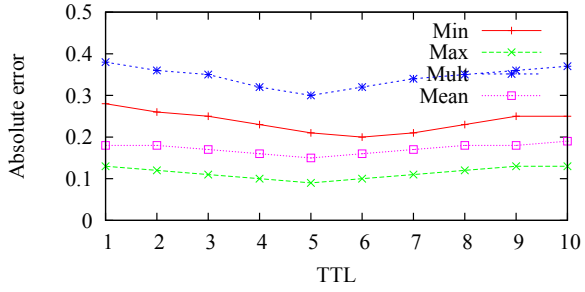
interested in determining what behaviour a user should have (the user should be very confident, averagely confident or somewhat confident), to receive the best answer. In general, behaviours of users can be split into three degrees as shown in Fig. 6.13.

Degree 1: (High confidence degree) Being very confident decreases the accuracy results. In fact, decreasing λ, (i.e decreasing TTL and increasing MTT) means limiting the number of intermediary users that risk to receive an answer.

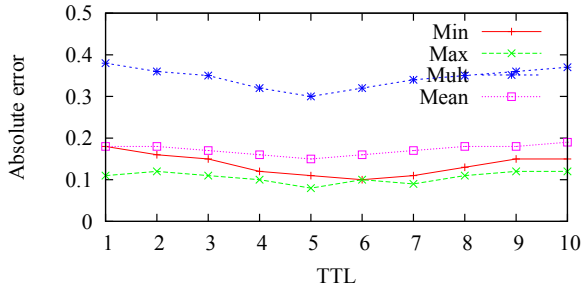
Degree 2: (Medium confidence degree) Best results are obtained with this behaviour. Even if the source increases the number of transitivity hops (TTL) and



(a) Absolute error for 1000 users



(b) Absolute error for 8000 users



(c) Absolute error for 14000 users

Figure 6.12: Comparison of Absolute error for different transitivity hops after optimisation

decreases the MTT, it can consider the familiarity with the trustee to extend not beyond some given value of TTL (e.g., from 3 to 7 hops) and no less than a certain value of MTT (e.g., between 0.3 and 0.7).

Degree 3: (Low confidence degree) By increasing λ, the sink becomes stranger to the source. Thus, the accuracy's results decrease until reaching its minimum values when MTT $= 0.1$ and TTL $= 9$.

Based on the above experimental results, we can conclude first that the MTT and TTL factors badly influence on the quality of the computed indirect trust. Second, to guarantee better results, these factors should not only be homogeneous

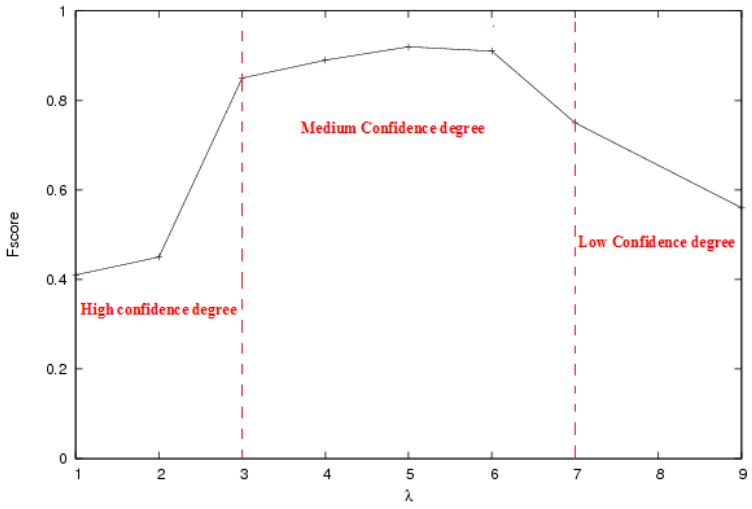


Figure 6.13: Impact of the user's behaviour

but also fulfil the medium confidence degree.

6.5.6 Comparison with other methods

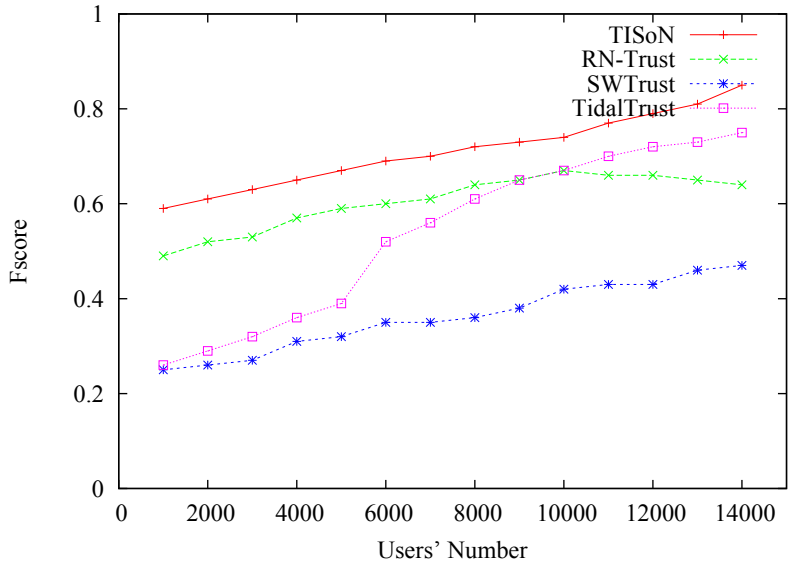


Figure 6.14: Fscore for the different methods

This part of experiments compares TISoN versus TidalTrust, RN-Trust and SWTrust algorithms. This comparison aims to show which algorithm is

more competitive and effective. According to Fig. 6.13, TISoN obtains better results as far as the MTT and TTL factors satisfy the medium confidence degree. Thus, we consider that MTT and TTL values are equal to resp. 0.5 and 5.

Fig. 6.14 shows that TISoN outperforms its competitors. The worst results are given by the SWTrust algorithm. In fact, in Section 3.3 in Chapter 3, we showed that SWTrust gives very low indirect trust values tending towards zero. This finding decreases the inferred trust quality.

Moreover, we see that Fscore values increase by increasing the number of users for all the methods except RNTrust. Indeed, by increasing the number of users, the number of trust paths increases. Then, we can conclude that the most trustworthy values comes from considering more available information. This finding explicates why RN-Trust (that considers all paths) exceeds TidalTrust until about 10000 users. Further, TidalTrust is interested only on shortest paths. Yet, with this restriction on the paths' length, TidalTrust risks suffering from its problem of the one path (mentioned in Chapter 3) which decreases the inferred trust quality, especially when the number of users (less than 7000 users) and then the number of paths is low.

However, when the number of users increases, the number of untrustworthy users increases. Consequently, RNTrust will consider the opinions of untrustworthy users. This explains its decreasing values of the accuracies starting from 10000 users.

To avoid worsen the accuracy of the results, considering all paths between the source and the sink is not desired. Certainly, our algorithm TISoN, considers only paths where all edges have trust values at least equal to the MTT and ignore paths having lower values to avoid using considering opinions of malicious users. It also, uses the TTL parameter, since getting information from few intermediate person should usually be more reliable than information passed down a long chain of users.

6.6 Conclusion

In this chapter, we introduced a new approach, TISoN, to infer trust relationships among OSN users. Accompanying this model, a new trust path searching algorithm TPS is proposed and a new trust inference method TIM is presented.

Furthermore, some required properties of the model are introduced and we investigate how our approach satisfies them. We discussed how trust inference with TISoN takes out all the major problems of previous existing algorithms. A trust network is modelled with the Advogato dataset to validate the effectiveness of TISoN. We conducted several experiments and the results show that our algorithm can generate high quality trust networks.

In this respect, the scalability issue is one of the most compelling challenges when facing real life OSNs. In fact, running TPS to compute a trusted path between two users in a network with several thousands nodes will exceed the complexity of the computations and will not satisfy the users' waiting time for the reply to their requests. We managed to run our algorithms in a way that can handle the size of the data and the complexity of the computations and to return high quality values in a good run time.

Next chapter includes proposing algorithms that compute the global trust of a user to present his reputation on the network. This computation is based on both local and inferred trust values to consider the opinions of all the network users. We think about using fuzzy linguistic expressions instead of the trust values since they are more natural for users than numerical values. Indeed, a user can simultaneously belong to several clusters of trust with different degrees of membership. So, we need to make use of a fuzzy clustering algorithm to determine which cluster reflects better a user reputation.

Chapter 7

Reputation management in OSNs

7.1 Introduction

A common challenge facing the OSNs as collaborative systems is how to effectively collaborate in accomplishing tasks while mitigating the malicious behaviours throughout collaboration. This is because users in OSNs have not knowledge about other users with whom there is no prior interaction or experiences. Reputation-based trust management has been used as an effective solution to evaluate how much one user can trust others, to help users to make the difference between trustworthy and untrustworthy users and encourage honest users by rewarding them with high trust values.

As we explained in Chapter 5, trust is often considered as a personal and subjective measure because it is computed primarily based on a set of personalized factors and can be derived from a combination of personal experience and relationships. However, reputation is often considered as a collective and objective measure of trustworthiness based on the transactional experiences and direct interactions of different users.

In this chapter, we propose two algorithms called *RepC* and FCR for reputation management in OSNs. The proposed algorithms are based on direct and indirect trust values computed respectively in Chapters 5 and 6. As illustrated in Fig. 7.1, the different OSN users participate to help the requester to have an idea about the reputation of an OSN user (cible). Some users, whom are indirectly connected to the cible, are observers and propagators, they observe direct interactions and, based on their experiences, they propagate information about trust with the different users. Other users (assessors), directly connected to the cible, are observers and evaluators since they evaluate directly the cible's trust. The requester can so scan the reputation of the cible based on direct and indirect trust.

RepC and FCR are truly unique since they are based on clustering algorithms. In fact, they divide OSNs users into clusters (groups) such that trustworthy users belong to the same cluster. *RepC* is an exclusive algorithm such that a user belongs to only one cluster. FCR is a fuzzy extension of the *RepC* algorithm which associates each user with every cluster to some degree using a membership function. The experimental results illustrate that FCR obtains a more reasonable reputation evaluation result than *RepC*.

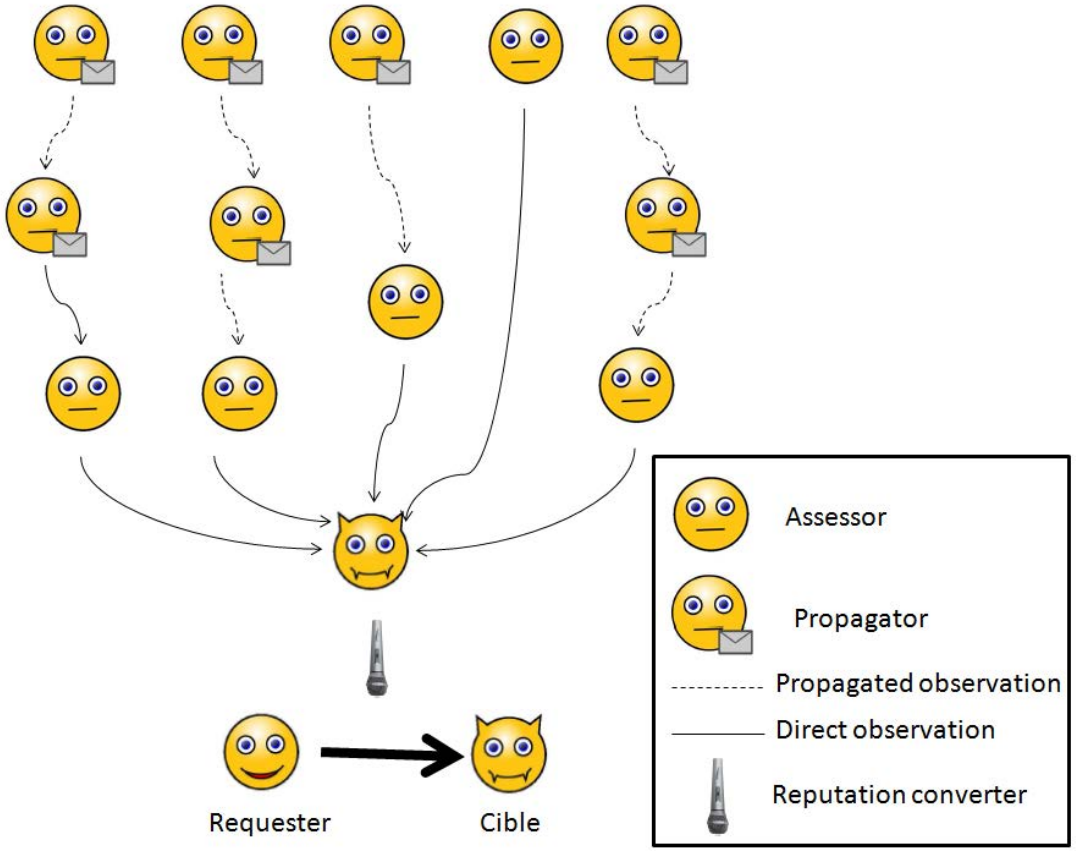


Figure 7.1: The propagated reputation

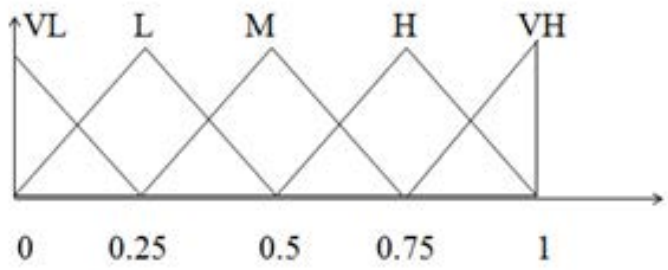


Figure 7.2: Stratification of reputation values

7.2 Choosing the initial centroids of the clusters

To counter the argument against subjectivity in reputation values, it is possible to use a stratification of reputation or trust as mentioned in Chapter 2. This lamination is easy to understand, flexible, tolerant of imprecise data and is used to capture the subjective and humanistic concept of a very good or good, and bad or very bad opinions. In this chapter, we give each strata a label. Thus, for example, a reputation value equal to 1 would be labelled "very high reputation". We suggest the stratification given in Figure 7.2. In fact, we associate a Triangular Fuzzy Number (TFN) that enables us to specify a range for a given reputation level instead of giving it a particular discrete value. The meaning of the different linguistic values (fuzzy set) are defined as: Very Low (VL), Low (L), Medium (M), High (H), and Very High (VH), to range users from very untrustworthy to very trustworthy as illustrated in Table 7.1. The advantage of this stratification is that a reputation value, denoted as "high" of one user is acknowledged by others as a high reputation value. Thus, we avoid the problem of "what does a reputation value of 0.2, or 20%, mean? Is it high or low?", for example.

In Chapter 3, we mentioned two main categories of clustering: hierarchical clustering and partitioning clustering. In this work, we choose to follow the second category. Indeed, the time and space complexities of the partitioning algorithms are typically lower than those of the hierarchical algorithms [Day, 1992]. In fact, partitioning methods have advantages in applications requiring large data sets as OSNs, which is not always the case for hierarchical clustering for which the construction of a tree is computationally expensive. In addition, the problem of the choice of the number of desired output clusters accompanying the use of a

Trust Level	Description	TFN
VL	very untrustworthy	-0.25, 0, 0.25
L	untrustworthy	0, 0.25, 0.5
M	medium trustworthy	0.25, 0.5, 0.75
H	trustworthy	0.5, 0.75, 1
VH	very trustworthy	0.75, 1, 1.25

Table 7.1: Linguistic values of reputation

partitioning algorithm is solved. In fact, each reputation strata shown in Section 7.2 presents a cluster, thus, the number of clusters (VL, L, M, H and VH) is 5.

7.3 RepC : Exclusive Clustering Algorithm for Reputation Management in OSNs

In this section, we introduce the *RepC* algorithm for the clustering of OSNs' users based on their reputation. In *RepC*, the global reputation of each user u is weighted by aggregating the direct and indirect trust values assigned to user u by other ones. In Subsection 7.3.1, we discuss how to aggregate these normalized trust values in a sensible manner in order to obtain the corresponding reputation values. Then, in Subsection 7.3.2, we tackle the problem of classifying users into different clusters based on their reputation similarity such that most trustworthy users belong to the same cluster.

7.3.1 Aggregating Direct and Inferred Trust Values

We have defined, on the basis of the previous Chapter 5, how to compute direct trust values in OSNs. Accordingly, our model titled $IRIS$, builds direct trust relations by aggregating different ties in a multiplex network (the direct interactions between users, their existing relationship types and their interest similarity). On the other hand, in Chapter 6, we have proposed the accurate model $TISoN$ to infer indirect trust in OSNs based on direct relationships between users. The direct and indirect computed values are normalized (all values are between 0 and 1) in the aim to lead to an elegant probabilistic interpretation.

A natural way to evaluate the reputation of an OSN user j is to aggregate the opinion of all users about that user, i.e., to consider all direct and indirect trust values assigned to him (See Eq. 7.1).

$$r_j = \frac{\sum_{i=1}^{n} \alpha.t_{ij}/i \neq j}{n + n'} \tag{7.1}$$

Where r_j denotes the reputation of the OSN user j based on all users' opinions; t_{ij} is the trust value assigned to j by the user i; n is the number of OSN users and n' is the number of direct relations in the OSN. The parameter is defined as follows:

$$\alpha = \begin{cases} 1 & \text{if } t_{ij} \text{ is an inferred trust value} \\ 2 & \text{if } t_{ij} \text{ is a direct trust value} \end{cases}$$

We can write this in matrix notation: if we define T to be the square matrix $[t_{ij}]$, then R is the column vector with r_j values such as $j \in [1..n]$. This is a useful way to have each user gain a view of the OSN that is wider than his own experience.

7.3.2 Algorithm Description of RepC

In *RepC*, we adopt the typical k-Means algorithm [Hartigan et Wong, 1979] which is the simplest and most used partitioning algorithm since it is easy to implement and its time complexity is about $O(n)$, where n is the number of objects. K-means starts with a random initial partition and keeps reassigning the object to clusters based on the similarity between the object and the cluster centroid until a convergence criterion is met. In our work, as mentioned in Section 7.2, the initial clusters are properly chosen and their number is equal to 5. Thus, we do not face the major problem of k-means which is sensitive to the random selection of the initial partition.

The process, used by *RepC*, is sketched by Algorithm 9. First, in lines 3 to 4, *RepC* creates the set G of 5 empty clusters (VL, L, M, H,VH) with the initialized centroids such that $c_1 = 0; c_2 = 0.25; c_3 = 0.5; c_4 = 0.75$; and $c_5 = 1$. Second, in line 7, the algorithm assigns each user j to one cluster g_p such that his reputation r_j is closer to this cluster centroid c_p. Then, in line 8, *RepC* recomputes the centroid of each cluster as the mean of reputations of users belonging to the cluster. The process of updating and recomputing centroids of clusters as well as

Algorithm 9: The RepC Algorithm

Data: R: reputation vector with r_j values, $j = 1 \dots 5$.
C: the set of initialized centroids c_p.
ϵ: error threshold, $\epsilon \approx 0$.

Result: G: the set of final clusters or groups g_p.

1 **begin**
2 $k \leftarrow 0$
3 $VL \leftarrow \emptyset$; $L \leftarrow \emptyset$; $M \leftarrow \emptyset$; $H \leftarrow \emptyset$; $VH \leftarrow \emptyset$;
4 $G \leftarrow \{(VL, c_1); (L, c_2); (M, c_3); (H, c_4); (VH, c_5)\}$;
5 **repeat**
6 $k \leftarrow k + 1$;
7 Update clusters $\in G$ by assigning each user j to one cluster g_p such that $Min(|r_j - c_p| / p \in [1 \dots 5])$;
8 Recompute the vectors of centroids $C^k = [c_p]^k$ by using Eq. 7.2;
9 **until** $\left|C^k - C^{k-1}\right| < \epsilon$;

assigning users to the adequate clusters is repeated until the stability condition is reached (line 9).

$$c_p = \frac{\sum r_j}{l}, p \in [1 \dots 5], l \in [1 \dots n] \tag{7.2}$$

In Eq. 7.2, n is the number of the OSN users, l is the number of users j belonging to cluster g_p with the centroid c_p.

7.3.3 An Illustrative Example

This section shows an illustrative example to epitomize our proposed method. Table 7.2 presents a simple square trust matrix $T = t_{ij}$, where $i \in [1 \dots 8]$ and $j \in [1 \dots 8]$, to show how much user i trusts user j. The set of reputation values $R = r_1, r_2, \dots, r_8$, corresponding to the different users, are computed in the vector matrix as described in Table 7.3. The process of the algorithm *RepC* when applied to these data is depicted in Figure 7.3.

As shown by Figure 7.3, the 5 clusters VL, L, M, H and VH with the respective centroids 0, 0.25, 0.5, 0.75 and 1, are initialized by the empty set. At step 1, *RepC* assigns users to clusters with the nearest centroids. For example, U_1 with

	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8
U_1	-	0.70	0.40	0.80	0.30	0.20	0.60	0.40
U_2	0.60	-	0.50	0.90	0.50	0.10	0.40	0.60
U_3	0.50	0.75	-	0.75	0.40	0.30	0.50	0.70
U_4	0.40	0.65	0.60	-	0.60	0.20	0.70	0.50
U_5	0.70	0.80	0.80	0.85	-	0.15	0.75	0.90
U_6	0.55	0.75	0.66	0.88	0.40	-	0.75	0.60
U_7	0.60	0.70	0.70	0.90	0.50	0.10	-	0.75
U_8	0.55	0.65	0.70	0.80	0.65	0.30	0.65	-

Table 7.2: A simple example of trusted OSN with 8 users

U	R
U_1	0.55
U_2	0.71
U_3	0.62
U_4	0.84
U_5	0.47
U_6	0.19
U_7	0.62
U_8	0.62

Table 7.3: The reputations' values

a reputation degree equal to 0.55 is assigned to the cluster M having the closest centroid. U_2 has a reputation degree equal to 0.71, for this, he is assigned to cluster H, and so on. Then, the different centroids are recomputed by using Eq. 7.2. For example, the cluster M, composed by 5 users U_1, U_3, U_5, U_7 and U_8, obtains the new centroid equal to 0.58. This value is computed, using those users' reputation values (Table 7.3), as follows: $c_3 = \frac{0.55+0.62+0.47+0.62+0.62}{5} = 0.58$. At step 2, users are reassigned to the clusters with nearest centroid and the process is repeated at step 3. *Repc* detects a stability condition. So, the process comes to an end and returns the obtained clusters.

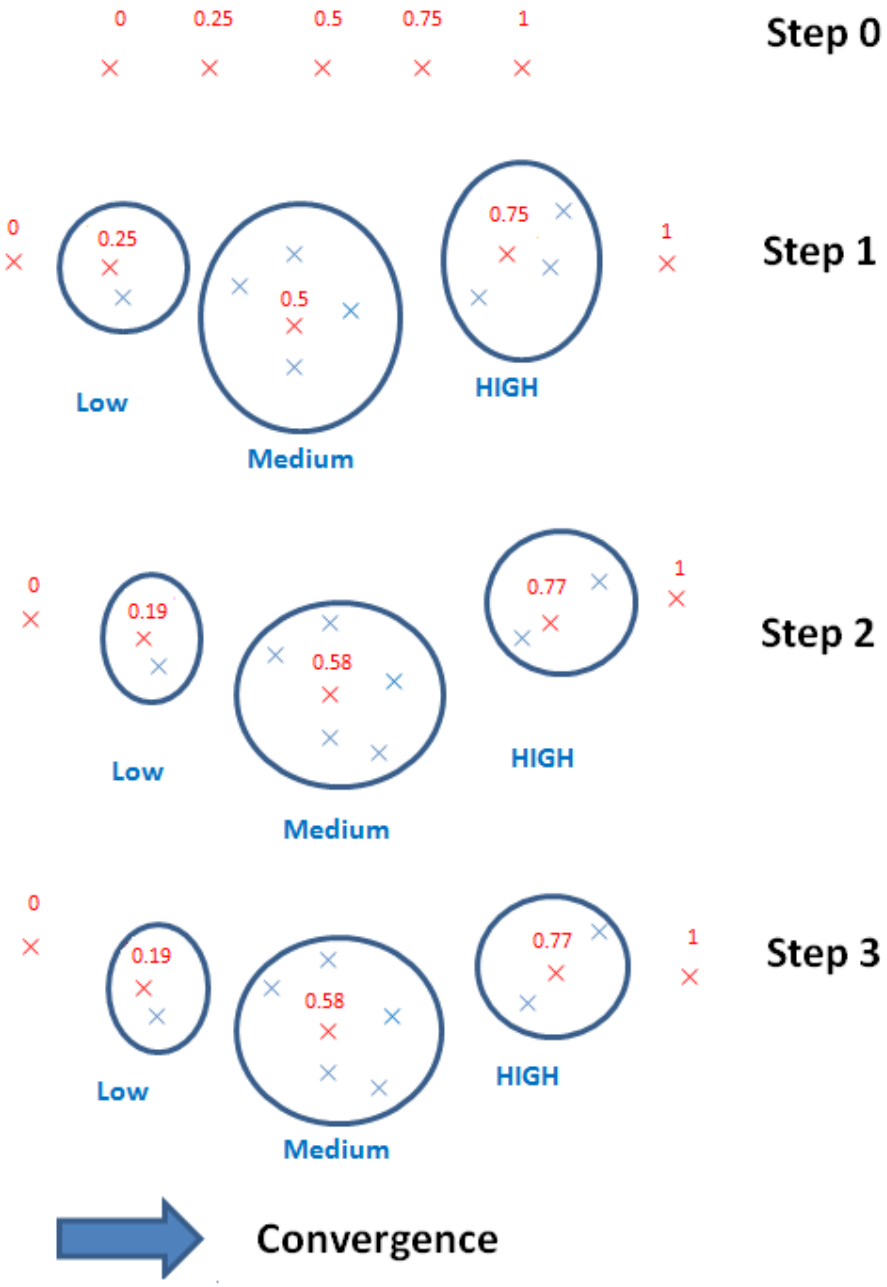


Figure 7.3: Clustering of users based on *RepC* algorithm

7.4 FCR : Fuzzy Clustering Algorithm for Reputation Management in OSNs

7.4.1 Drawbacks of the RepC algorithm

Since the opinions of users vary from an OSN user to another one, having several users could lead to different evaluations (benevolent users w. r. t. a user can be malicious w. r. t. another one). As described in Section 7.3, *RepC* is a hard (exclusive) algorithm that affects each user in the OSN to only one cluster. This classification ignores some users' opinions. Despite of the ignored values express opinions of a minority of users, they should be considered for two reasons. First, the objectivity property of reputation allowing to review all points of view must be quite respected. Second, this will offer a more clear vision to the OSN users by offering a much finer degree of detail of the reputation management model. For example, it is much clearer to a user requesting about a reputation of another user j, to answer with "45% of users think that j is untrustworthy, whereas 55% think that j is very trustworthy", than answering briefly with "(s)he is very trustworthy". That is, with the second answer, we do not know whether all the users agree that j is very reputable, or if there exist other users thinking that he is mediocre or not reputable.

To improve the process of users' classification and to address the above mentioned limits in *RepC*, we propose FCR, a novel Fuzzy Clustering algorithm for Reputation management in OSNs, that affects one user to more than one cluster. The use of FCR in grouping users better reflects the nature of human behaviours in that a user may have more than one behaviour and thus, he may be associated to more than one group.

7.4.2 Fuzzification of Trust Values

Certainty signifies that the characteristics and patterns of a model are known and that there are no disbeliefs about their metrics or their occurrence. However, fuzziness occurs more in models for human judgement, analysis and decision are important, since to decide and choose which data is more critical or indispensable than other data, or which data is needed more quickly, is a highly humanistic concept that fuzzy logic is able to model. This is the case of trust decisions.

Algorithm 10: The Membership Computing Algorithm

Data: T: trust matrix t_ij.

Result: Δ: the fuzzy partition of T.

```
1 begin
2     for k ← 1 to 5 do
3         for i ← 1 to N do
4             for J ← 1 to N do
5                 if i <> j then
6                     compute ω_k(t_ij) by using Eq. 7.3
7             compute δ_jk by using Eq. 7.4
```

Indeed, fuzzy logic can deal with uncertainty and imprecise information, as the trust and the reputation, with high efficiency based on the notion of membership function [Suna *et al.*, 2011]. We use fuzzy logic to define the membership that the object belongs to a set of data. It uses 1 to show the completely belonging to the set and 0 as completely not belonging to the set, and other values mean the degree of the membership.

In Section 7.2, we mentioned that we associate a Triangular Fuzzy Number (TFN) to specify a range for a given trust level instead of giving it a particular discrete value. With our proposed FCR algorithm, TFN is defined as a triplet $(p_{i-1}; p_i; p_{i+1})$ where $p_{i-1} \leq p_i \leq p_{i+1}$. The membership function of a TFN, as depicted in Figure 7.2, is defined in Eq. 7.3.

$$\omega_k(t_{ij}) = \begin{cases} 0, & \text{if } t_{ij} = p_{k-1} \text{ or } t_{ij} = p_{k+1} \\ 1, & \text{if } t_{ij} = p_k \\ \frac{t_{ij} - p_{k-1}}{p_k - p_{k-1}}, & \text{if } p_{k-1} < t_{ij} < p_k \\ \frac{t_{ij} - p_{k+1}}{p_k - p_{k+1}}, & \text{if } p_k < t_{ij} < p_{k+1} \end{cases} \tag{7.3}$$

We suppose each fuzzy set presents a cluster, the set of clusters being $G = [g_k]$, where $k \in [1 \ldots 5]$. Each cluster g_k has the centroid p_k.

The membership function $\Omega : T \times G \rightarrow [0, 1]$ computes the membership degree $\omega_k(t_{ij}) \in [0, 1]$ denoting to what extent a trust value t_{ij} belongs to the cluster g_k. For example, in Figure 7.2, a trust value $t = 0.8$ denotes that it belongs to the cluster **H** with 80% ($\frac{0.8-1}{0.75-1} = 0.8$) and it belongs to the cluster **VH** with 20%

Algorithm 11: The FCR Algorithm

Data: T: trust matrix t_ij.
C: the set of initialized centroids c_p.
ϵ: error threshold, $\epsilon \approx 0$.

Result: G: the set of final clusters or groups g_p.
Δ: the final fuzzy 5-partition of T.

begin
  $m \leftarrow 0$
  $VL \leftarrow \emptyset$; $L \leftarrow \emptyset$; $M \leftarrow \emptyset$; $H \leftarrow \emptyset$; $VH \leftarrow \emptyset$;
  $G \leftarrow \{(VL, c_1); (L, c_2); (M, c_3); (H, c_4); (VH, c_5)\}$;
  repeat
    Compute the membership matrix Δ by calling Algorithm 10;
    At step m, recompute the vectors of centroids $C^m = [c_k]^m$ as follows:
    $c_k = \frac{\sum_{j=1}^{N} \delta_{jk} \times \frac{\sum_{i=1}^{N} t_{jk}}{N}}{\sum_{j=1}^{N} \delta_{jk}}$ with N is the numbers of users and $k \in [1 \dots 5]$
    $m \leftarrow m + 1$
  until $|C^m - C^{m-1}| < \epsilon$;

$(\frac{0.8-0.75}{1-0.75} = 0.2)$.

7.4.3 Description of the FCR Algorithm

In this section we try to classify users considering the n trust values affected to them. We first initialize the trusted matrix T. Second, using the membership function Ω, we compute what clusters, g_k, a user j belongs to, with the correspondent membership degrees $\delta_k(j)$.

$$\delta_{jk} = \frac{\sum_{i=1}^{n} \omega_k(t_{ij})}{\sum_{i=1}^{n} \omega(t_{ij})} \tag{7.4}$$

Let Δ be a real $G \times N$ matrix, $\Delta = [\delta_{jk}]$. Δ is the matrix representation of the partition T obtained by running Algorithm 10. We refer to Δ as a fuzzy 5-partition of T when the elements of Δ are numbers in the unit interval $[0, 1]$ that continue to satisfy both equations 7.5 and 7.6:

$$\sum_{j=1}^{n} \delta_{jk} \geq 0 \tag{7.5}$$

	VL	L	M	H	VH
U_1	0%	5.7%	65.7%	28.6%	0%
U_2	0%	0%	17.2%	80%	2.8%
U_3	0%	5.7%	42.3%	49.2%	2.8%
U_4	0%	0%	0%	64%	36%
U_5	0%	22.8%	62.9%	14.3%	0%
U_6	28.6%	65.7%	5.7%	0%	0%
U_7	0%	5.7%	40%	54.3%	0%
U_8	0%	5.75%	42.8%	40%	11.45%

Table 7.4: Membership degrees of trusted OSN shown in Table 7.2

$$\sum_{k=1}^{c} \delta_{jk} = 1 \tag{7.6}$$

The main steps of the FCR algorithm are outlined below by Algorithm 11. First, in lines 3 and 4, FCR creates the set G of 5 empty clusters (VL, L, M, H,VH) with the initialized centroids such that $c_1 = 0; c_2 = 0.25; c_3 = 0.5; c_4 = 0.75$; and $c_5 = 1$. Second, in line 6, FCR calls Algorithm 10 to compute the membership matrix Δ. An element δ_{jk} of this matrix represents the membership degree of the user j to the cluster c_k. Then, FCR recomputes the centroids of the different clusters (cf. line 7). In fact, each new centroid is recomputed as the mean of the trust values of users assigned to its corresponding cluster, multiplied by their membership degrees. The process is repeated until centroids in C do not change significantly (cf. line 10).

7.4.4 Illustrative Example

Next we use an example to explain the process of FCR more thoroughly. Let us consider the trusted OSN shown in Table 7.2. The initial membership degrees to the different clusters VL, L, M, H and VH with the respective centroids 0, 0.25, 0.5, 0.75 and 1, are computed in the matrix described in Table 7.4 thanks to the Algorithm 10.

For example, U_5 belongs to clusters L, M and H with respectively membership degrees equal to 22.8%, 62.9% and 14.3%. In fact, the computation of the

User i	t_{i5}	VL	L	M	H	VH
U_1	0.3	0%	80%	20%	0%	0%
U_2	0.5	0%	0%	100%	0%	0%
U_3	0.4	0%	40%	60%	0%	0%
U_4	0.6	0%	0%	60%	40%	0%
U_6	0.4	0%	40%	60%	0%	0%
U_7	0.5	0%	0%	100%	0%	0%
U_8	0.65	0%	0%	40%	60%	0%

Table 7.5: Membership degrees of trust values affected to $U5$

membership degrees of U_5 to the different clusters respects Eq. 7.6 (0% + 22.8% + 62.9% + 14.3% + 0% = 100%). U_5 receives different trust values from the different users. For instance, (s)he receives a trust value equal to 0.3 from U_1. By applying Eq. 7.3, this value is assigned to the cluster L with 80% and to M with 20%. Table 7.5 details the membership degrees of all the different trust values affected to U_5 from the different users. As we see, no trust value belongs to each of clusters VL and VH what justifies the disassociation of U_5 to both of these clusters.

At each step m, FCR assigns each user to different clusters with regard to its membership degrees. Figure 7.4 depicts the clustering of users in step 1. The higher the membership degree corresponding to one cluster is, the nearer to the cluster centroid the user is. For instance, U_5 is nearer to the centroid of cluster M since the maximum membership degree of $U5$ is 62.9% corresponding to this cluster.

7.4.5 Controversiality of users' reputation

In this section, we explain our method to cluster users based on their controversiality degrees. To do this, we defuzzify the fuzzy partition of FCR while taking into account the fact that different users can have different opinions about a specific user.

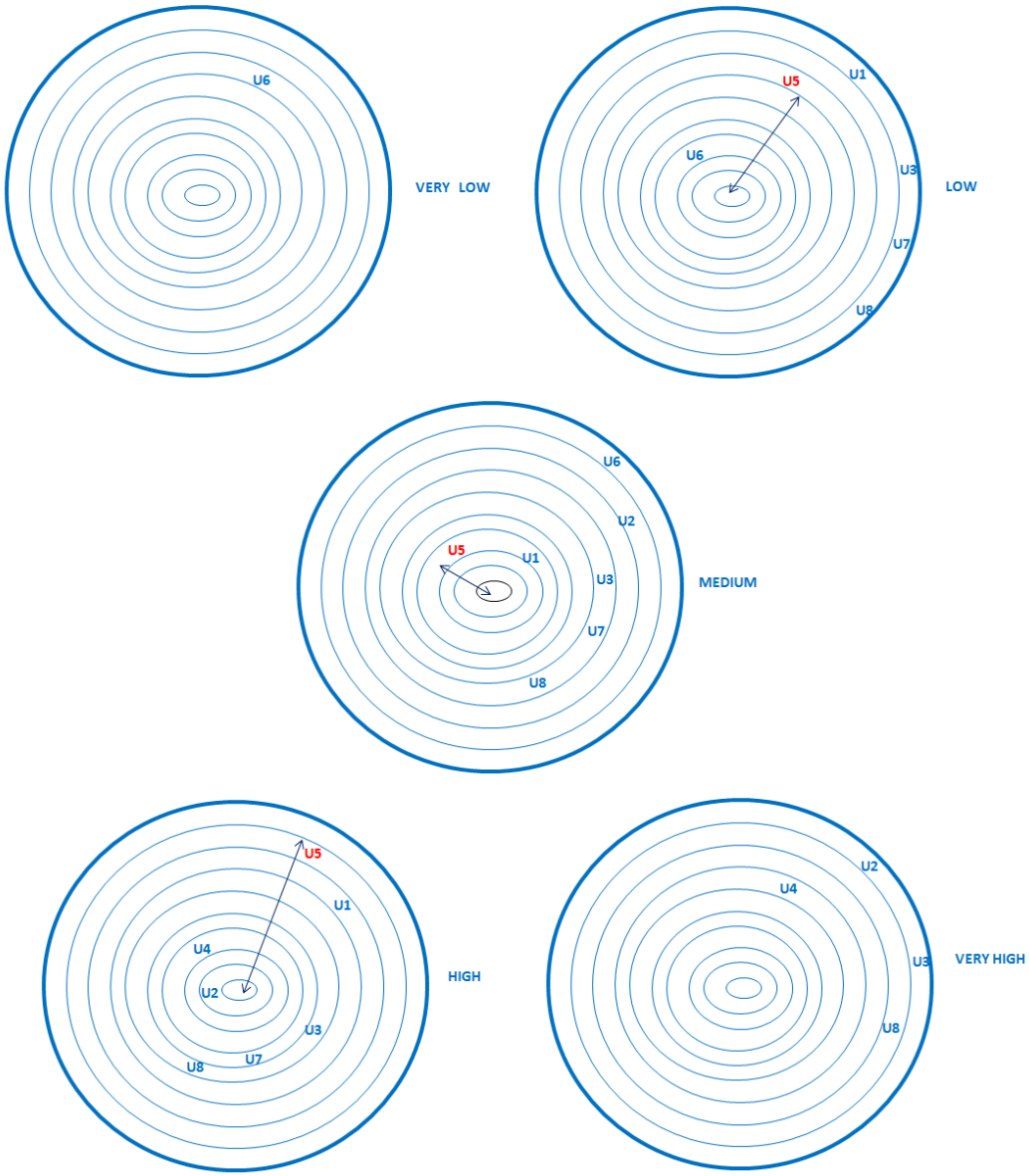


Figure 7.4: Clustering of users based on FCR algorithm (first iteration)

Defuzzification process of FCR

The input for the defuzzification process is the fuzzy membership matrix Δ and the output of defuzzification process is a crisp value obtained by using some defuzzification methods such as centroid and maximum.

In this work, we use the maximum membership degree for defuzzifying fuzzy output functions (membership functions). This method, also known as the *height method*, is given by the algebraic expression in Eq. 7.7 :

$$\delta_{jk^*} \geq \delta_{jk} \; for \; all \; cluster \; k \in [1,5] \tag{7.7}$$

where k^* is the defuzzified cluster.

Controversiality of users' reputation

Totally belonging to a cluster does not raise a problem and means a global agreement about the correspondent user. Thus, he is a non-controversial user since all the other users agree on their opinions about him. However, belonging to different clusters with different membership degrees means that the different users have different opinions about the specific user and thus a disagree between them in issuing a statement about that user.

We now define, in Eq. 7.8, the controversiality degree, cd_k. This quantity assigned to a user j, $cd_k(j)$, presents the number of users who disagree with the majority in assigning that user in the cluster g_k. Thus, it presents the user's membership degrees to the other clusters $g_{p \neq k}$. So, we can simplify the computation of $cd_k(j)$ using the matrix Δ as presented in Eq. 7.9.

$$cd_k(j) = \frac{|receivedTrust(j \in g_p)|/p \neq k}{|receivedTrust(j \in g_k)| + |receivedTrust(j \notin g_k)|} \tag{7.8}$$

$$cd_k(j) = \sum_{p=1}^{5} \delta_{jp} \; with \; p \neq k \tag{7.9}$$

The more this quantity increases, the more the user is controversial. A user who has a controversiality degree of x is called $x - controversial$. $0 - controversial$ users are belonging to a unique cluster (with a membership degree equal to 1) and they are non controversial.

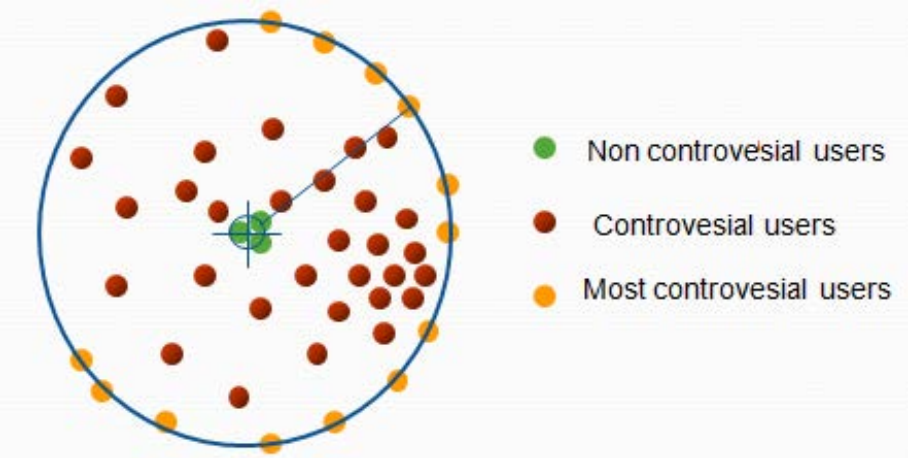


Figure 7.5: Dispersal of users in a cluster according to their controversiality degree

Clustering of users based on their controversiality degrees

The manner of presenting users in the cluster can not be carried out randomly and uniformly for all the users. We aim to find the best manner such that we can understand the difference between the different users of the same cluster.

Since we have 5 clusters (VH, H, M, L, VL), the maximum membership degree of a user j to a cluster k can vary between 0.2 and 1: $1/5 < \delta_{jk} \leq 1$. Thus, we can notice that the controversiality degree of this user also varies as: $0 \leq cd_k(j) < 0.8$. We translate this announcement by presenting each cluster with users belonging to it as depicted in Fig 7.5. On the one hand, the center consists of the non contreversial users that are mainly receiving the same statements. On the other hand, the perimeter is composed by the highest controversial users in the cluster. The rest of the users are scattered in the area of the cluster according to their controversiality degree. The less the users are controversial, the nearer to the cluster's center are.

In the case of the **VH** and **VL** clusters, users in the center are not contreversial since they are mainly trusted or distrusted respectively. Thus, they present respectively the most benevolent and malicious users in the social network.

7.5 Enriching FOAF with reputation

In Chapter 5, we have introduced the direct trust computation and proposed a trust schema adding to the FOAF ontology the **foaf:directTrust** property.

In the remainder, we propose an enrichment of FOAF by associating each user with a reputation value. Our trust schema adds to FOAF the **foaf:reputation** property, having as domain value foaf:Person.

This property is functional (this means that, for any particular person, we can expect at most one value for that property) and presents the reputation of the concerned person.

- Domain: foaf:Person; having this property implies being a Person.

- Range: every value of this property is a String.

This property connects a Person to a string representing its reputation. This string corresponds to the cluster a user belongs to, when applying one of our algorithms. It takes a value from the predefined set "very untrustworthy", "untrustworthy", "medium trustworthy", "trustworthy", "very trustworthy". Table 7.6 sketches the Reputation vocabulary for describing the reputation of people.

We use the SPARQL query language to retrieve reputation information from the RDF graph. For example, to retrieve the most benevolent users from FOAF files, we use the SPARQL query, presented below returning the set of users having the best reputation values: ("very trustworthy"), without having to crawl the Web for data:

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE { ?user foaf:name ?name .
        ?user foaf:reputation "very trustworthy".  }
```

7.6 Experiments

In this section, we describe different experiments we lead on the proposed algorithms *RepC* and *FCR*. We use different types of criteria for clustering evaluation. In addition, we discuss our experimental results, of the proposed algorithms *RepC* and *FCR* based on two real datasets of social networks as well as on a random data.

foaf:reputation Property value	**Corresponding Cluster**	**Description**
very untrustworthy	**VB**	Classifies a person having a very bad reputation and considered as a malicious person by almost of his judgers.
untrustworthy	**VB**	Classifies a person lowly trusted by his judgers.
medium trustworthy	**M**	Classifies a controversial person, which means that his judgers split into people that mediumly trust him, and other users that split into two same-sized opinions group about this user, i.e. one half of users considers him as benevolent (with good reputation) and the other half as malicious (bad reputation).
trustworthy	**H**	Classifies a person highly trusted by his judgers
very trustworthy	**VH**	Classifies a person having a very good reputation and considered as a benevolent person by almost of his judgers.

Table 7.6: The proposed vocabulary for describing reputation of people

7.6.1 Validation of the clusters

The clusters' evaluation or the assessment of the quality of the obtained clusters presents an important topic related to clustering. Most of cluster validity measures evaluate the trade-off between cluster compactness and separability [Portmann, 2012]. Other measures are used to evaluate how well a clustering approach performs on a dataset. These measures are usually associated to the type of the considered criterion in evaluating the clustering approach quality. In fact, in the literature, there exist three types of measures for clustering evaluation: Internal criterion; External criterion and Relative criterion [Vendramin *et al.*, 2010].

- *Internal criterion:* This criterion assigns best scores to approaches producing clusters with low similarity between objects in different clusters and high similarity between objects in the same cluster.
- *External criterion:* This criterion is used to compare the clustering results against external benchmark.
- *Relative criterion:* This criterion requires the definition of the user need. For example, some users need a faster clustering algorithm poorly performing on internal criterion than another slow algorithm that performs excellent based on an internal criterion.

Indeed, there does not exist consistent evaluation clustering method integrating the relative criterion [Portmann, 2012]. However, for the internal and external criteria, there exist several validation methods [ZHAO, 2012]. In our experiments, we adapt the internal criteria indexes of Dunn [Dunn, 1974] as well as that of Davies and Bouldin [Davies et Bouldin, 1979] for our exclusive clustering algorithm *RepC*. These two criteria consider a clustering algorithm as good and successful whenever it generates clusters with high intra-cluster homogeneity, good inter-cluster separation and high connectedness between neighbouring data objects. For our fuzzy algorithm, we use the Xie-Beni index [Xie et Beni, 1991] which presents a fuzzy validity criterion based on a validity function which identifies overall compact and separate fuzzy partitions. As external criterion, we adapt the F-score accuracy [Rijsbergen, 1979] to compare the similarity of two clustering results.

The Dunn Index

The Dunn Index, I_D, identifies clusters which are well separated and compact. The goal is therefore to maximize the inter-cluster distance while minimizing the intra-cluster distance. As shown in Eq. 7.10, I_D is the report between the maximum distance separating two users classified together and the minimum distance between two users classified separately. For a good clustering, I_D should be as high as possible.

$$I_D = min_{1 \leq i \leq n}[min_{1 \leq j \leq n, i \neq j}(\frac{d(i,j)}{max_{1 \leq k \leq n} d'(k)})] \tag{7.10}$$

With:

- $d(i,j)$: the distance between clusters i et j
- $d'(k)$: the diameter of cluster k

The Davies and Bouldin Index

Davies and Bouldin Index, I_{DB}, identifies clusters which are far from each other. It is defined by the average of cluster evaluation measures for all the clusters as described in Eq. 7.11. For a good clustering, the I_{DB} should be as low as possible.

$$I_{DB} = \frac{1}{n} \sum_{i=1}^{n} max_{i \neq j}(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}) \tag{7.11}$$

With:

- n: the number of clusters.
- c_i: the centroid of i^{th} cluster
- σ_i: the average distance between objects of cluster i and the centroid c_i
- $d(c_i, c_j)$: the distance between centroids c_i and c_j

The Xie-Beni index

The Xie-Beni index is a fuzzy clustering index, but it is also applicable to hard clustering [Desgraupes, 2013]. It is defined, as shown in Eq. 7.14, as the quotient between the mean quadratic error and the minimal squared distances between the centers of the clusters.

Consider a fuzzy partition of the data set $X = \{x_j; j = 1, .., n\}$ where n is the number of objects and $v_i (i = 1, ..., n_c)$ the centers of the clusters. Also, consider u_{ij} the membership of data object j with respect to cluster i.

$$d_{ij} = u_{ij} \left\| x_i - v_j \right\| \tag{7.12}$$

The fuzzy deviation d_{ij} of x_j from cluster i, is defined, as shown in Eq. 7.12, as the distance between x_j and the center of cluster weighted by the fuzzy membership of data object j belonging to cluster i. Whereas, the separation of the fuzzy partitions, as shown in Eq. 7.13, is defined as the minimum distance between cluster centers.

$$D_{min} = min \left\| v_i - v_j \right\| \tag{7.13}$$

$$I_{XB} = \frac{\sum_{i=1}^{c} \sum_{j=1}^{n} d_{ij}}{n.D_{min}} \tag{7.14}$$

It is clear that small values of I_{XB} are expected for compact and well-separated clusters.

F-score

We adopt the commonly used metric in information retrieval, F-score metric, defined in Eq. 7.15, to compare the accuracy of the proposed methods. It is based on precision and recall metrics defined successively in Eq. 7.16 and Eq. 7.17.

$$F - Score = \frac{2 \times (Pecision \times Recall)}{(Pecision + Recall)} \tag{7.15}$$

The precision is the number of users correctly attributed to one cluster with regard to the total number attributed to this cluster proposed by the algorithm.

$$Precision = \frac{(Number\ of\ users\ correctly\ attributed\ to\ the\ cluster)}{(Number\ of\ users\ attributed\ to\ the\ cluster)} \tag{7.16}$$

The recall is defined by the number of users correctly attributed to one cluster with respect to the number of users that really belong to this cluster.

$$Recall = \frac{(Number\ of\ users\ correctly\ attributed\ to\ the\ cluster)}{(Number\ of\ users\ belonging\ to\ the\ cluster)} \tag{7.17}$$

The higher the recall and precision are, the more desirable the measures are for good algorithm performance. Thus, we make use of F-score to indicate our algorithms performances. Obviously, high F-score values are desirable.

7.6.2 Data set description

To test the validity of our clustering algorithm *RepC* and FCR, we executed our algorithms on two data sets.

Twitter Data: We use a data set [1] collected from the real social network Twitter. This data set containing more than 250000 users and 320000 relations, uses a social labelled graph. Each node of the graph presents a Twitter member and each edge denotes the number of retweets one user gives to another user.

Random Data: We create a dataset that produces until one million users overlay network with random values of reputation between 0 and 1.

7.6.3 Performance Study

The experiments, presented in the remainder, aim at comparing our proposed clustering algorithms based on internal and external indexes shown in Section 7.6.1 using tha Twitter dataset. In addition, with the random dataset that produces until one million of users, the experiments aim at comparing the algorithms on the basis of their respective running times.

Twitter Data

To test the performance of our algorithms, we conduct different experiments. Firstly, to test the *RepC* algorithm, we process the dataset and we run the programs computing the I_D and I_{DB} indexes by varying the number of users. Secondly, we use the I_{XB} index to compare *RepC* with FCR for different number of users. Finally, we compute the F-score measure for both *RepC* and FCR to find to which degree they provide more relevant results.

As shown in Table 7.7, after 5000 users, as far as the number of users increases, the I_D decreases and the I_{DB} increases. This finding is due to a decrease in the minimum distance inter-cluster and an increase in the maximum diameter intra-cluster. This is caused by the rise in the number of cluster users leading to

[1] https://snap.stanford.edu/data/higgs-twitter.html

$\#Users$	I_D	I_{DB}
100	0.25	0.52
500	0.25	0.53
1000	0.32	0.48
5000	0.38	0.45
10000	0.31	0.45
50000	0.31	0.56
100000	0.24	0.92
200000	0.23	1.23

Table 7.7: The I_D and I_{DB} Cluster validity values for the *RepC* algorithm

$\#Users$	FCR	*RepC*
100	0.63	0.59
500	0.61	0.55
1000	0.56	0.53
5000	0.51	0.42
10000	0.52	0.41
50000	0.58	0.45
100000	0.62	0.48
200000	0.66	0.49

Table 7.8: The I_{XB} Cluster validity values for the *RepC* and FCR algorithms

the cluster's expansion (resp. an increase in a cluster diameter), and so a higher degree of clusters overlap (resp. a decrease in the distance between clusters).

Results in Table 7.8, show that compared to the FCR algorithm, *RepC* is more effective. Indeed, whatever the given number of users, the *RepC* gives better I_{XB} values. In fact, each fuzzy cluster of a partition is considered as a fuzzy set, and the whole data set is the universe for them. Then, the separation between clusters can be modelled by the similarity between all these fuzzy sets. Thus, a low similarity means a better separation. In the case of the *RepC* algorithm, each reputation value belongs to only one class, which decreases the similarity between the different fuzzy sets.

In order to assess the improvement introduced by the fuzzy algorithm FCR, we simulate the Twitter dataset first with $RepC$. We then compare the results vs those obtained for the fuzzy case. We run the programs computing the F-score metric for both $RepC$ and FCR methods by varying the number of users.

To compute the F-score values, we define the importance degree I_D notion presenting the reputation of one user in Twitter. In fact, the more a user is reputable and important in the network, the higher the number of his shared (retweeted) tweets is.

The simulation results, as seen in Figure 7.6, show that the F-score values for both $RepC$ and FCR increase as the number of users increases. In fact, by the increasing number of users, direct and indirect trust values, considered to compute reputation values, increase leading to a rise in the authentication success trust rate. Then reputation values are more accurate. However, if the number of users increases sharply (more then 50000 users), the number of users correctly attributed to clusters by the $RepC$ algorithm decreases generating a decrease of precision and recall values and consequently a decline of the F-score. This is not the case of FCR which can tolerate and keep producing correct results with an increasing of the number of users. We notice here that with the addition of the fuzzy notion, an improvement is achieved over the F-score values that reach 0.9 with 200000 users. In fact, due to the membership function in fuzzy systems, one user can belong to more than one cluster. Thus, the probability of finding the adequate cluster increases engendering a rise in the number of users correctly attributed to the cluster. This finding enhances the tolerance of fuzzy algorithms to imprecise data.

Random Data

With processing the random dataset, the comparison between $RepC$ and FCR algorithms is done on the basis of their respective execution times as far as the number of users varies.

On the basis of experiments, as shown in Fig. 7.7, it is merely visible that the FCR algorithm is taking more time for computation than that of the $RepC$ algorithm. In fact, $RepC$ is based on the one of the simplest algorithm K-means that works really well with large datasets. However, fuzzy clustering algorithm includes much more fuzzy logic based calculations, so its computational time

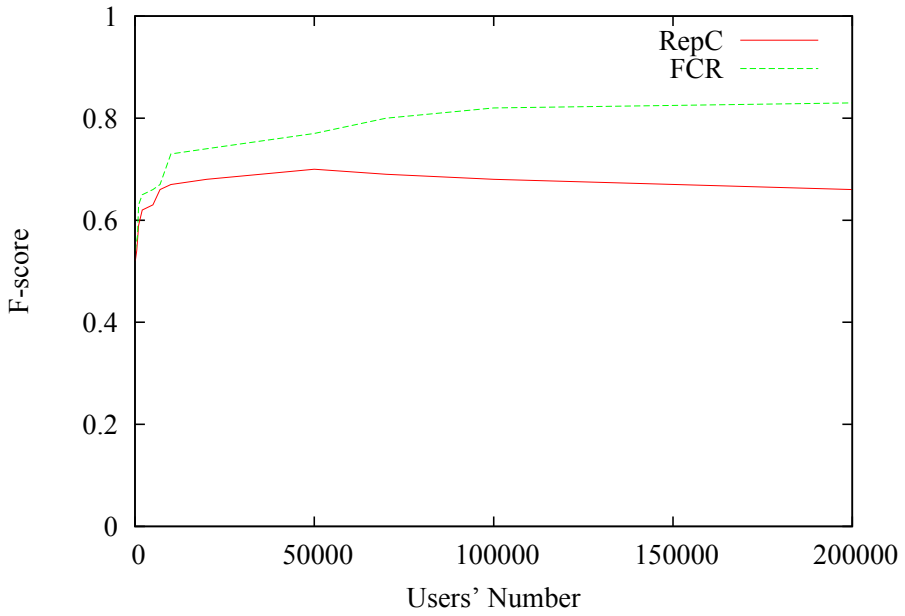


Figure 7.6: F-score results for *RepC* and FCR with varying the number of users

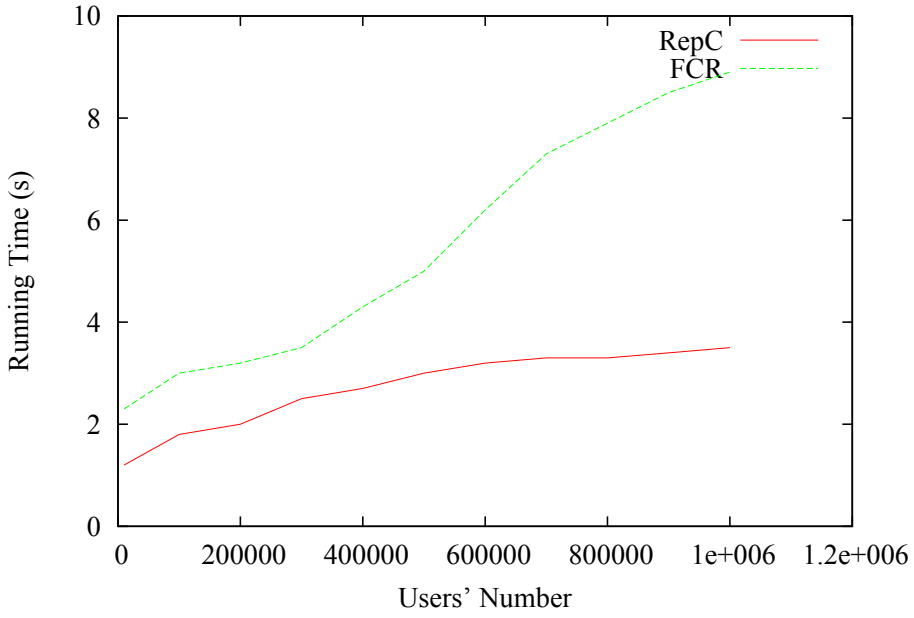


Figure 7.7: *RepC* Vs FCR in terms execution time with varying the number of users

increases consequently. So, we can conclude the fact that the *RepC*'s performance is better than that of FCR in terms of computational time.

7.7 Conclusion

In this chapter, we have proposed two new clustering reputation algorithms *RepC* and FCR based on the trust network generated in previous chapters. These algorithms classify an OSN users into clusters by their trust similarity such that most trustworthy users belong to the same cluster. It is worth of mention, favouring a particular clustering algorithm is solely dependent on the type of the data to be clustered and the purpose of the clustering applications. Our hard clustering algorithm *RepC* is suitable for users preferring exclusive clustering results. However, our fuzzy clustering algorithm FCR is suitable for overlapping clustering task. In our situation, trust and reputation are imprecise data, and fuzzy logic is conceptually, flexible and tolerant to imprecise data and uncertainty. For those purposes, we generally prefer our membership value based clustering algorithm FCR. Nonetheless, individuals in social networks can choose to associate other users in only one class using *RepC* or to different classes with different membership values using FCR.

Conclusions and Future Works

In this chapter, we present the conclusions and the future works of this research work. We begin by reminding the general problem and the objectives. Then, we point out the main contributions of this thesis, described in detail along this manuscript. . Finally, we address the perspectives of this work.

7.8 General Problem and Objectives

In recent years, OSNs have become ubiquitous and important for content sharing. And yet, thanks to their public accessibility, these sites are now playing a prominent role for users in sharing opinions, thoughts, information, and experience and connecting to their friends and families. However, because of this public accessibility, OSNs' users face many problems related to trust. In fact, in OSNs, most users do not have direct interactions previously, thus, they are not able to make difference between malicious and benevolent users when needing to benefit from a service or to interact with others. Therefore, trust and reputation become one of the most important indications for users' decision making and become significant and necessary for a successful social network.

7.9 Contributions

In this thesis, in order to produce effective and efficient trust and reputation management models to provide a reasonable trust value, five major contributions have been proposed. The contributions are summarised below.

The first contribution of the work proposed in this thesis is the extraction of a complex trust-oriented users' interests. In fact, social networks contain complex social information, including social relationships, social interests and preferences that play a fundamental role in defining trusted social connections. Taking this into account, we proposed an original process that builds a network of users grouped according to their interests by examining their interactions with a folksonomy site. This process is split into two phases: the tag's context recognition (a) and Interests extraction of users (b).

(a) Using free-tagging makes ambiguity handling a compulsory issue. In fact, users can make many grammatical, syntactic and other mistakes. To tackle this issue, a pre-processing step has been proposed. Then, a process for

recognizing the contexts of tags by linking them to DBpedia concepts has been proposed and applied to the previous step's resulting set of unambiguous tags.

(b) Folksonomies and OSNs do not explicitly state shared conceptualizations, nor do they force users to use the same tags. However, the usage of tags of users with similar interests tends to converge to a shared vocabulary. In our work, a users' semantic-clustering algorithm has been proposed identifying the specific interests of users.

The second contribution of the work proposed in this thesis is the evaluation of direct trust between OSNs' users.

(a) Our proposed approach, called IRIS, considers social activities of users including their social relationships, preferences and interactions to measure direct trust values.

(b) We addressed the lack of trust details in the FOAF data by enriching it using the information available on trust social networks given by our method IRIS. Our FOAF enrichment method allows users to indicate a level of trust for people they directly know.

Our third contribution of the work proposed in this thesis is a novel model of trust inference in trust-oriented social networks, called TISoN.

(a) A trust path selecting algorithm (TPS) has been proposed since evaluating trust via all the social trust paths in a large-scale trust network is computationally infeasible. With TPS, a source user can specify his trust evaluation criteria.

(b) After identifying the trustworthy social trust path, we have proposed a method understanding how trust is propagated along the trust path in order to compute the indirect trust value given to the target.

The fourth contribution of the work presented in this thesis is reputation management in OSNs.

(a) We have introduced a new clustering reputation algorithm, RepC, based on a trust network. This algorithm classifies an OSN users into clusters by their trust similarity such that most trustworthy users belong to the same cluster. In order to obtain a more reasonable reputation evaluation result, we have proposed a second algorithm, FCR, which is a fuzzy extension of RepC.

(b) By enriching the FOAF ontology with social reputation information, we have shown that reputation can be merged between different users from multiple social networks.

7.10 Future Works

This thesis mainly concentrates on the trust and reputation management in OSNs. Indeed, in the one hand, we have analysed and discussed different existent works. In the other hand, to address some of the current limitations, we have proposed different approaches addressing four aspects of social trust including direct trust, inferred trust, reputation and linking trust with the semantic Web. However, we think that a set of issues, for building more effective trust community in social networks, is yet to be developed and explored by our proposed solutions. In specific, we propose the following issues that can improve and be further explored.

(1) Trust relations between two users in OSNs are dynamic. Therefore, in order to acquire a more reasonable trust computing result, in addition to the current proposed trust management models, we plan to study a dynamic and powerful method to compute the updated trust situation based on analysing and investigating the conversations between users in OSNs in real-time.

(2) Our trust and reputation methods can be useful on evaluating mobile applications. Indeed, a mobile device has evolved into an open platform to install and execute various software packages and applications. Which mobile application is more trustworthy for a user to acquire, download, install, consume or recommend can be a vital issue that impacts its final success. Thus, our trust and reputation management methods can, for example, help a user to find the most trustworthy application for online reservation of accommodations. In addition, we can, via user-device interactions, enrich our methods by extracting new useful information and properties that can be considered in developing a trust metric in mobile systems.

Bibliography

[Abdessalem et BenDhia, 2011] Abdessalem, T. et BenDhia, I. (Athens, Greece, June 12-16, 2011). A reachability-based access control model for online social networks. *In Proceedings of the First ACM SIGMOD Workshop on Databases and Social Networks, DBSocial' 11*, pages 31–36.

[Abdul-Rahman et Hailes, 2000] Abdul-Rahman, A. et Hailes, S. (2000). Supporting trust in virtual communities. *In Proceedings of the Hawaii International Conference on System Sciences, USA*, pages 4–7.

[Adamatti *et al.*, 2013] Adamatti, D. F., Castelfranchi, C. et Falcone, R. (2013). Structural transitivity of trust in academic social networks using agent-based simulation. *In Proceedings of the Eleventh European Workshop on Multi-Agent Systems (EUMAS'13), Toulouse, France*, pages 97–111.

[Adler, 2001] Adler, P. S. (2001). Market, hierarchy, and trust: The knowledge economy and the future of capitalism. *Organization Science*, 12(2):215–234.

[Al-Oufi *et al.*, 2012] Al-Oufi, S., Kim, H.-N. et El-Saddik, A. (2012). A group trust metric for identifying people of trust in online social networks. *Expert Systems with Applications*, 39(18):13173–13181.

[Ali *et al.*, 2007] Ali, B., Villegas, W. et Maheswaran, M. (Richmond Hill, Ontario, Canada, October 22-25, 2007). A trust based approach for protecting user data in social networks. *In* Lyons, K. A. et Couturier, C., éditeurs : *Proceedings of the 2007 conference of the Centre for Advanced Studies on Collaborative Research, CASCON' 07*, pages 288–293. IBM.

[Angeletou *et al.*, 2009] Angeletou, S., Sabou, M. et Motta, E. (2009). Folksonomy enrichment and search. *In Proceedings of The 6th European Semantic Web Conference, ESWC' 09*, pages 801–805.

[Ashri *et al.*, 2005] Ashri, R., Ramchurn, S. D., Sabater, J., Luck, M. et Jennings, N. R. (July 25-29, 2005). Trust evaluation through relationship analysis. *In*

4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), trecht, The Netherlands, pages 1005–1011. ACM.

[Baase et Van Gelder, 2000] Baase, S. et Van Gelder, A. (2000). *Computer algorithms: introduction to design and analysis*. Addison-Wesley.

[Beatty *et al.*, 2011] Beatty, P., Reay, I., Dick, S. et Miller, J. (2011). Consumer trust in e-commerce web sites: A meta-study. *ACM Comput. Surv.*, 43(3):1–46.

[Beckmann *et al.*, 1990] Beckmann, N., Kriegel, H.-P., Schneider, R. et Seeger, B. (1990). The r*-tree: An efficient and robust access method for points and rectangles. *In Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, SIGMOD '90, pages 322–331, New York, NY, USA. ACM.

[Benantar, 2006] Benantar, M. (2006). *Access Control Systems: Security, Identity Management and Trust Models*. New York, NY: Springer.

[Berkhin, 2002] Berkhin, P. (2002). Survey of clustering data mining techniques. Rapport technique.

[Berkhin, 2006] Berkhin, P. (2006). A survey of clustering data mining techniques. Rapport technique.

[Bezdek, 1981] Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Advanced applications in pattern recognition. Plenum Press, New York.

[Bhuiyan, 2010] Bhuiyan, T. (2010). A survey on the relationship between trust and interest similarity in online social networks. *Journal of Emerging Technologies in Web Intelligence*, 2(4):291–299.

[Bhuiyan *et al.*, 2010a] Bhuiyan, T., Jøsang, A. et Xu, Y. (2010a). Managing trust in online social networks. *In Handbook of Social Network Technologies*, pages 471–496.

[Bhuiyan *et al.*, 2010b] Bhuiyan, T., Josang, A. et Xu, Y. (2010b). *Trust and Reputation Management in Web-based Social Network*. Web Intelligence and Intelligent Agents Zeeshan-Ul-Hassan Usmani (Ed.), ISBN: 978-953-7619-85-5, InTech, DOI: 10.5772/8375.

[Bizer *et al.*, 2009] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R. et Hellmann, S. (2009). Dbpedia - a crystallization point for the web of data. *The International Journal on Semantic Web*, 7(3):154–165.

[Bojars *et al.*, 2008] Bojars, U., Passant, A., Breslin, J. G. et Decker, S. (2008). Social network and data portability using semantic web technologies. *In Proceedings of the 2nd Workshop on Social Aspects of the Web (SAW 2008) at BIS2008*, pages 5–19.

[Bora et Gupta, 2014] Bora, D. J. et Gupta, A. K. (2014). A comparative study between fuzzy clustering algorithm and hard clustering algorithm. *Computing Research Repository , CoRR*, abs/1404.6059.

[Boyd et Ellison, 2007] Boyd, D. M. et Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230.

[Brickley et Miller, 2010] Brickley, D. et Miller, L. (2010). FOAF Vocabulary Specification 0.97. Namespace document.

[Bush, 1996] Bush, V. (1996). As we may think (reprint). *Interactions*, 3(2):35–46.

[Carminati et Ferrari, 2009] Carminati, B. et Ferrari, E. (Crystal City, Washington, USA, November 11-14, 2009). Enforcing relationships privacy through collaborative access control in web-based social networks. *In Proceedings of the 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom' 09*, pages 1–9.

[Chen *et al.*, 2011] Chen, D., Chang, G., Sun, D., Li, J., Jia, J. et Wang, X. (2011). Trm-iot: A trust management model based on fuzzy reputation for internet of things. *Computer Science and Information Systems*, (20):1207–1228.

[Christianson et Harbison, 1996] Christianson, B. et Harbison, W. S. (1996). Why isn't trust transitive? *In Proceedings of the Security Protocols Workshop, Cambridge, United Kingdom*, pages 171–176.

[Christianson et Harbison, 1997] Christianson, B. et Harbison, W. S. (1997). Why isn't trust transitive? *In Proceedings of the International Workshop on Security Protocols*, pages 171–176, London, UK. Springer-Verlag.

[Cook *et al.*, 2005] Cook, K. S., Yamagishi, T., Cheshire, C., Cooper, R., Matsuda, M. et Mashima, R. (2005). Trust building via risk taking: A cross-societal experiment. *Social Psychology Quarterly*, 68:121–142.

[Davies et Bouldin, 1979] Davies, D. L. et Bouldin, D. W. (1979). A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(2):224–227.

[Davis et Jr, 2010] Davis, I. et Jr, E. V. (2010). Relationship: A vocabulary for describing relationships between people. *http://purl.org/vocab/relationship*.

[Day, 1992] Day, W. H. E. (1992). *Complexity theory: An introduction for practitioners of classification*, chapitre 6, pages 199–235. World Scientific Publishing.

[Demartini, 2007] Demartini, G. (2007). Finding experts using wikipedia. *In Proceedings of the 2nd International ISWC+ASWC Workshop on Finding Experts on the Web with Semantics, Busan, Korea,*, pages 33–41.

[Desgraupes, 2013] Desgraupes, B. (2013). *Clustering Indices*. University Paris Ouest, Lab Modal'X, France.

[Diederich et Iofciu, 2006] Diederich, J. et Iofciu, T. (2006). Finding communities of practice from user profiles based on folksonomies. *In EC-TEL Workshops*.

[D.J., 2009] D.J., B. (2009). A socal network prespective on industral/organizational psychology. *Industrial/Organizational Handbook*.

[Doerr *et al.*, 2007] Doerr, M., Ore, C.-E. et Stead, S. (2007). The cidoc conceptual reference model: a new standard for knowledge sharing. *In Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual modeling*, volume 83 de *ER '07*, pages 51–56, Darlinghurst, Australia.

[DuBois *et al.*, 2011] DuBois, T., Golbeck, J. et Srinivasan, A. (2011). Predicting trust and distrust in social networks. *In Proceedings of the third IEEE International Conference on Privacy, Security, Risk and Trust (PASSAT),and the third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011*, pages 418–424.

[Dumbill, 2002] Dumbill, E. (2002). XML watch: Finding friends with XML and RDF. Rapport technique, IBM Developers Works.

[Dumouchel, 2005] Dumouchel, P. (2005). Trust as an action. *European Journal of Sociology / Archives Européennes de Sociologie*, 46:417–428.

[Dunn, 1973] Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

[Dunn, 1974] Dunn, J. C. (1974). Well separated clusters and optimal fuzzy-partitions. *Journal of Cybernetics*, 4:95–104.

[Ester *et al.*, 1995] Ester, M., Kriegel, H.-P. et Xu, X. (1995). A database interface for clustering in large spatial databases. *In* Fayyad, U. M. et Uthurusamy, R., éditeurs : *KDD*, pages 94–99. AAAI Press.

[Faatz et Steinmetz, 2004a] Faatz, A. et Steinmetz, R. (2004a). Ontology enrichment evaluation. *In Proceedings of the 14th International Conference Engineering Knowledge in the Age of the Semantic Web (EKAW2004)*, pages 497–498, Whittlebury Hall, Northamptonshire, UK.

[Faatz et Steinmetz, 2004b] Faatz, A. et Steinmetz, R. (2004b). Precision and recall for ontology enrichment. *In Proceedings of the Workshop on Ontology Learning and Population, (ECAI2004)*, Sevilla, Spain.

[Garcia-Silva *et al.*, 2011] Garcia-Silva, A., Corcho, O., Alani, H. et Gomez-Perez, A. (2011). Review of the state of the art: Discovering and associating semantics to tags in folksonomies. *Knowledge Engineering Review*, 26(4).

[García-Silva *et al.*, 2011] García-Silva, A., Jakob, M., Mendes, P. N. et Bizer, C. (2011). Multipedia: enriching dbpedia with multimedia information. *In Proceedings of the Sixth International Conference on Knowledge Capture (K-CAP2011)*, pages 137–144.

[Giannakidou *et al.*, 2008] Giannakidou, E., Koutsonikola, V. A., Vakali, A. et Kompatsiaris, Y. (2008). Co-clustering tags and social data sources. *In The Ninth International Conference on Web-Age Information Management, WAIM'08*, pages 317–324.

[Gimpel *et al.*, 2008] Gimpel, J. G., Karnes, K. A., McTague, J. et Pearson-Merkowitz, S. (2008). Distance-decay in the political geography of friends-and-neighbors voting. *Political Geography*, 27:231–252.

[Golbeck, 2006a] Golbeck, J. (2006a). Combining provenance with trust in social networks for semantic web content filtering. *In Proceedings of the International Conference on Provenance and Annotation of Data*, IPAW'06, pages 101–108, Berlin, Heidelberg. Springer-Verlag.

[Golbeck, 2006b] Golbeck, J. (Pisa, Italy, May 16-19, 2006b). Generating predictive movie recommendations from trust in social networks. *In Proceedings of The Fourth International Conference iTrust'06*, pages 93–104.

[Golbeck et Hendler, 2006] Golbeck, J. et Hendler, J. A. (2006). Inferring binary trust relationships in web-based social networks. *ACM Trans. Internet Techn.*, 6(4): 497–529.

[Golbeck, 2005] Golbeck, J. A. (2005). Computing and applying trust in web-based social networks. *PhD thesis, Faculty of the Graduate School of the University of Maryland, College Park, USA.*

[Grandison, 2003] Grandison, T. W. A. (2003). Trust management for internet applications. *PhD thesis, Imperial College of Science, Technology and Medicine University of London, UK.*

[Gruber, 2005] Gruber, T. (volume 3, 2005). Ontology of folksonomy: A mash-up of apples and oranges. *The International Journal on Semantic Web and Information Systems*, pages 1–11.

[Guth *et al.*, 2006] Guth, W., Mengel, F. et Ockenfels, A. (2006). An evolutionary analysis of buyer insurance and seller reputation in online markets. *Theory and Decision*, 63(3):265–282.

[Guy et Tonkin, 2006] Guy, M. et Tonkin, E. (2006). Folksonomies: Tidying up tags? *D-Lib, January*, volume 12(1).

[Hamdi *et al.*, 2013] Hamdi, S., Bouzeghoub, A., Gançarski, A. L. et Yahia, S. B. (July 16-18, 2013). Trust inference computation for online social networks. *In 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013, Melbourne, Australia*, pages 210–217.

[Hamdi *et al.*, 2012a] Hamdi, S., Gançarski, A. L., Bouzeghoub, A. et BenYahia, S. (June 25-27, 2012a). Iris: A novel method of direct trust computation for generating trusted social networks. *In Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2012, Liverpool, United Kingdom*, pages 616–623.

[Hamdi *et al.*, 2012c] Hamdi, S., Gançarski, A. L., Bouzeghoub, A. et Yahia, S. B. (July 16-20, 2012c). Enriching the dbpedia ontology with shared conceptualizations from folksonomies. *In 36th Annual IEEE Computer Software and Applications Conference, COMPSAC 2012, Izmir, Turkey*, pages 551–556.

[Hamdi *et al.*, 2012b] Hamdi, S., Gançarski, A. L., Bouzeghoub, A. et Yahia, S. B. (July 4-6, 2012b). Enriching ontologies from folksonomies for elearning: Dbpedia

case. *In 12th IEEE International Conference on Advanced Learning Technologies, ICALT 2012, Rome, Italy*, pages 293–297.

[Hamdi *et al.*, 2011] Hamdi, S., Gançarski, A. L., Bouzeghoub, A. et Yahia, S. B. (October 19-21, 2011). Semantic clustering of users based on shared conceptualizations in folksonomies. *In International Conference on Computational Aspects of Social Networks, CASoN 2011, Salamanca, Spain,*, pages 201–206.

[Hardin, 2002] Hardin, R. (2002). Trust and trustworthiness. *Russell Sage Foundation.*

[Hardin, 2004] Hardin, R. (2004). *Trust and Trustworthiness.* G - Reference,Information and Interdisciplinary Subjects Series. Russell Sage Foundation.

[Hartigan et Wong, 1979] Hartigan, J. A. et Wong, M. A. (1979). A K-means clustering algorithm. *Applied Statistics*, 28:100–108.

[Heath et Bizer, 2011] Heath, T. et Bizer, C. (2011). *Linked Data.* published by Morgan & Claypool in the series Synthesis Lectures on the Semantic Web: Theory and Technology.

[Horwitz, 2013] Horwitz, J. (2013). Semiocast: Pinterest now has 70 million users and is steadily gaining momentum outside the us. Rapport technique, Retrieved February 27, 2014 from http://thenextweb.com/socialmedia/2013/07/10/semiocast-pinterest-now-has-70-million-users-and-is-steadily-gaining-momentum-outside-the-us/#!xB92V.

[Jain et Dubes, 1988] Jain, A. K. et Dubes, R. C. (1988). *Algorithms for clustering data.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[Jain *et al.*, 1999] Jain, A. K., Murty, M. N. et Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323.

[Jaschke *et al.*, 2008] Jaschke, R., Hotho, A., Schmitz, C., Ganter, B. et Stumme, G. (2008). Discovering shared conceptualizations in folksonomies. *The International Journal of Web Semantics*, 6(1):38–53.

[Javanmardi *et al.*, 2015] Javanmardi, S., Shojafar, M., Shariatmadari, S. et Ahrabi, S. S. (2015). Fr trust: A fuzzy reputation-based model for trust management in semantic p2p grids. *Int. J. Grid Util. Comput.*, 6(1):57–66.

[Jiang et Wang, 2011] Jiang, W. et Wang, G. (Changsha, China, November 16-18, 2011). Swtrust: Generating trusted graph for trust evaluation in online social networks. *In Proceedings of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications TrustCom 2011*, pages 320–327. IEEE Computer Society.

[Jiang *et al.*, 2014] Jiang, W., Wang, G. et Wu, J. (2014). Generating trusted graphs for trust evaluation in online social networks. *Future Generation Computer Systems,*, 31:48 – 58. Special Section: Advances in Computer Supported Collaboration: Systems and Technologies.

[Jøsang *et al.*, 2003] Jøsang, A., GRAY, E. et KINATEDER, M. (2003). Predicting trust and distrust in social networks. *In Proceedings of the 1st Workshop on Formal Aspects in Security and Trust (FAST'03)*, pages 9–22.

[Jøsang *et al.*, 2006] Jøsang, A., Marsh, S. et Pope, S. (May 16-19, 2006). Exploring different types of trust propagation. *In Proceedings of the 4th International Conference of Trust Management, iTrust'06, Pisa, Italy*, pages 179–192.

[Jøsang et Pope, 2005] Jøsang, A. et Pope, S. (2005). Semantic constraints for trust transitivity. *In Proccedings of the second Asia-Pacific Conference on Conceptual Modelling (APCCM'05), Newcastle, NSW, Australia*, pages 59–68.

[JÃ¸sang et Pope, 2003] JÃ¸sang, A. et Pope, S. (2003). Analysing topologies of transitive trust. *In Proceedings of the Second Usenixl Conference on File and Storage Technologies , FAST 2003, Mars 31-April 2, 2003, San Francisco, California, USA*.

[Kamvar *et al.*, 2003] Kamvar, S. D., Schlosser, M. T. et Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in p2p networks. *In Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA. ACM.

[Kaufman et Rousseeuw, 1990] Kaufman, L. et Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley.

[Kollock, 1994] Kollock, P. (1994). The Emergence of Exchange Structures: An Experimental Study of Uncertainty, Commitment, and Trust. *The American Journal of Sociology*, 100(2):313–345.

[Kunegis *et al.*, 2009] Kunegis, J., Lommatzsch, A. et Bauckhage, C. (2009). The slashdot zoo: Mining a social network with negative edges. *In 18th International World Wide Web Conference*, pages 741–741.

[Kurdi, 2015] Kurdi, H. A. (2015). Honestpeer: An enhanced eigentrust algorithm for reputation management in {P2P} systems. *Journal of King Saud University - Computer and Information Sciences*, 27(3):315 – 322.

[Kuter et Golbeck, 2007] Kuter, U. et Golbeck, J. (Vancouver, British Columbia, July 22-26, 2007). Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. *In Proceedings of the Twenty-Second Conference on Artificial Intelligence, AAAI' 07*, pages 1377–1382.

[Lawler et Yoon, 1996] Lawler, E. et Yoon, J. (1996). Commitment in exchange relations: Test of a theory of relational cohesion. *Am Sociol Rev*, 61(1):89–108.

[Lee et Oh, 2015] Lee, J. et Oh, J. C. (2015). A node-centric reputation computation algorithm on online social networks. *Applications of Social Media & Social Network Analysis*, pages 1–22.

[Lesani et Montazeri, 2009] Lesani, M. et Montazeri, N. (2009). Fuzzy trust aggregation and personalized trust inference in virtual social networks. *Computational Intelligence journal*, 25(2):51–83.

[Lewis et Weigert, 1985] Lewis, J. D. et Weigert, A. (1985). Trust as a Social Reality. *Social Forces*, 63(4):967–985.

[Lichtenstein et Slovic,] Lichtenstein, S. et Slovic, P. *The Construction of Preference*. Cambridge University Press.

[Limpens *et al.*, 2009] Limpens, F., Gandon, F. et Buffa, M. (2009). Linking folksonomies and ontologies for supporting knowledge sharing: a state of the art. Rapport technique, INRIA, Institut National de Recherche en Informatique et Automatique.

[Liu, 2013] Liu, G. (2013). Trust management in online social networks. *PhD thesis, Department of Computing, Faculty of Science, Macquarie University, Sidney, Australia*.

[Liu *et al.*, 2010] Liu, G., Wang, Y. et Orgun, M. A. (July, 11-15, 2010). Optimal social trust path selection in complex social networks. *In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA*.

[Liu et Sun, 2010] Liu, Y. et Sun, Y. L. (2010). Anomaly detection in feedback-based reputation systems through temporal and correlation analysis. *In Proceedings of the 2010 IEEE Second International Conference on Social Computing*, SOCIALCOM '10, pages 65–72, Washington, DC, USA. IEEE Computer Society.

[Luhmann, 1979] Luhmann, N. (1979). *Trust and Power*. Wiley.

[Mansell, 2013] Mansell, L. (2013). Instagram hits 100m monthly active users. Rapport technique, Retrieved February 27, 2014 from http://www.geekshut.com/instagram-hits-100m-monthly-active-users/8930.

[Mansell et Collins, 2005] Mansell, R. et Collins, B. (2005). Trust and crime in information societies. *Edward Elgar Publishing*.

[Marsh, 1992] Marsh, S. (1992). Trust in distributed artificial intelligence. *In Proceedings of the fourth European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW '92, S. Martino al Cimino, Italy*, pages 94–112.

[Marsh, 1994] Marsh, S. P. (1994). Formalising trust as a computational concept. Rapport technique, Ph.D. thesis, University of Stirling.

[Massa et Avesani, 2005] Massa, P. et Avesani, P. (2005). Controversial users demand local trust metrics: An experimental study on epinions.com community. *In Proceedings of the Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, AAAI 2005, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 121–126.

[Massa et Souren, 2008] Massa, P. et Souren, K. (2008). Trustlet, open research on trust metrics. *In* Flejter, D., Grzonkowski, S., Kaczmarek, T., Kowalkiewicz, M., Nagle, T. et Parkes, J., éditeurs : *BIS 2008 Workshop Proceedings*, pages 31–43.

[McBride, 2013] McBride, S. (2013). Start-up pinterest wins new funding, $2.5 billion valuation. Rapport technique, Retrieved February 27, 2014 from http://www.reuters.com/article/2013/02/21/net-us-funding-pinterest-idUSBRE91K01R20130221.

[Mika, 2007] Mika, P. (2007). Ontologies are us: A unified model of social networks and semantics. *Journal of Web Semantics*, 5(1):5–15.

[Miller, 2012] Miller, R. (2012). *Intimate Relationships,* 6^{th} *Edition*. McGraw-Hill Education.

[Mislove *et al.*, 2007] Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P. et Bhattacharjee, B. (2007). Measurement and analysis of online social networks. *In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42. ACM.

[Molm *et al.*, 2009] Molm, L. D., Schaefer, D. R. et Collett, J. L. (2009). Fragile and resilient trust: Risk and uncertainty in negotiated and reciprocal exchange. *Sociological Theory*, 27(1):1–32.

[Mori *et al.*, 2004] Mori, J., Matsuo, Y., Ishizuka, M. et Faltings, B. (2004). Keyword extraction from the web for foaf metadata. *In Proceedings of Workshop on Friend of a Friend, Social Networking and the Semantic Web, Galway, Ireland.*

[MUI, 2003] MUI, L. (2003). Computational models of trust and reputation: Agents, evolutionary games, and social networks. Rapport technique, Ph.D. thesis, Massachusetts Institute of Technology.

[Navigli et Velardi, 2006] Navigli, R. et Velardi, P. (2006). Ontology enrichment through automatic semantic annotation of on-line glossaries. *In Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2006), LNAI*, volume 4248, pages 126–140, Podebrady, Czech Republic.

[Nepal *et al.*, 2011] Nepal, S., Sherchan, W. et Paris, C. (Changsha, China, November 16-18, 2011). Strust: A trust model for social networks. *In Proceedings of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications TrustCom' 11*, pages 841–846. IEEE Computer Society.

[NEWMAN, 2005] NEWMAN, R. (2005). Tag ontology design. *http ://www.holygoat.co.uk/owl/redwood/0.1/tags/ visited on september 2011.*

[Ng et Han, 1994] Ng, R. T. et Han, J. (1994). Efficient and effective clustering methods for spatial data mining. *In Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 144–155, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Noorian et Ulieru, 2010] Noorian, Z. et Ulieru, M. (2010). The state of the art in trust and reputation systems: A framework for comparison. *J. Theor. Appl. Electron. Commer. Res.*, 5(2):97–117.

[Palchykov *et al.*, 2012] Palchykov, V., Kaski, K., Kertész, J., Barabási, A.-L. et Dunbar, R. I. M. (2012). Sex differences in intimate relationships. *CoRR*, abs/1201.5722.

[Passant, 2007] Passant, A. (2007). Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs. *In Proceedings of the First International Conference on Weblogs and Social Media (ICWSM2007)*, Boulder, Colorado.

[Portmann, 2012] Portmann, E. K. (2012). The fora framework: A fuzzy grassroots ontology for online reputation management. *PhD thesis, Faculty of Sciences, University of Fribourg, Switzerland.*

[Prud'hommeaux et Seaborne, 2006] Prud'hommeaux, E. et Seaborne, A. (April 2006). *SPARQL query language for RDF. W3C Recommendation.* http://www.perceive.net/schemas/20021119/trust/default.htm.

[Resnick *et al.*, 2000] Resnick, P., Kuwabara, K., Zeckhauser, R. et Friedman, E. (2000). Reputation systems. *Communications of the ACM*, 43(12):45–48.

[Richardson *et al.*, 2003] Richardson, M., Agrawal, R. et Domingos, P. (Sanibel Island, Florida, USA, October 20-23, 2003). Trust management for the semantic web. *In Proceedings of the 2nd International Semantic Web Conference, ISWC' 03,*, pages 351–368.

[Richters et Peixoto, 2010] Richters, O. et Peixoto, T. P. (2010). Trust transitivity in social networks. *Clinical Orthopaedics and Related Research, CORR' 10*, 468(12).

[Rijsbergen, 1979] Rijsbergen, C. J. V. (1979). *Information Retrieval.* Butterworth-Heinemann, Newton, MA, USA, 2nd édition.

[Rousseau *et al.*, 1998] Rousseau, D. M., Sitkin, S. B., Burt, R. S. et Camerer, C. (1998). Not so different after all: A cross-discipline view of trust. *Academy of Management Review*, 23(3):393–404.

[Ruohomaa *et al.*, 2007] Ruohomaa, S., Kutvonen, L. et Koutrouli, E. (2007). Reputation management survey. *In Proceedings of the Second International Conference on Availability, Reliability and Security, ARES'07.*, pages 103–111.

[Rusli, 2013] Rusli, E. M. (2013). Instagram pictures itself making money. Rapport technique, Retrieved February 27, 2014 from http://online.wsj.com/news/articles/SB10001424127887324577304579059230069305894.

[Shekarpour et Katebi, 2010] Shekarpour, S. et Katebi, S. D. (2010). Modeling and evaluation of trust with an extension in semantic web. *Journal of Web Semantics*, 8(1):26–36.

[Sherchan *et al.*, 2013] Sherchan, W., Nepal, S. et Paris, C. (2013). A survey of trust in social networks. *ACM Computing Surveys*, 45(4):47.

[Singh et Tomar, 2009] Singh, R. R. et Tomar, D. S. (2009). Approaches for user profile investigation in orkut social network. *International Journal of Computer Science and Information Security*, 6(2).

[Suna *et al.*, 2011] Suna, H., Yua, H., Yanga, N. et Lib, L. (2011). Expression and evaluation of reputation based on fuzzy set theoryâ‹†. *Journal of Information & Computational Science*, 8(1):139–138.

[Szomszor *et al.*, 2008a] Szomszor, M., Alani, H., Cantador, I., O'Hara, K. et Shadbolt, N. (2008a). Semantic modelling of user interests based on cross-folksonomy analysis. *In Proceedings of the International Semantic Web Conference*, pages 632–648.

[Szomszor *et al.*, 2008b] Szomszor, M., Cantador, I. et Alani, H. (2008b). Correlating user profiles from multiple folksonomies. *In Proceedings of the 19th ACM Conference on Hypertext and Hypermedia, Hypertext'08*, pages 33–42.

[Sztompka, 1999] Sztompka, P. (1999). *Trust: A Sociological Theory*. Cambridge Cultural Social Studies. Cambridge University Press.

[Taherian *et al.*, 2008] Taherian, M., Amini, M. et Jalili, R. (2008). Trust inference in web-based social networks using resistive networks. *In Proceedings of the 3rd International Conference on Internet and Web Applications and Services, ICIW 2008, 8-13 June 2008, Athens, Greece*, pages 233–238.

[Tajeddine *et al.*, 2011] Tajeddine, A., Kayssi, A., Chehab, A. et Artail, H. (2011). Fuzzy reputation-based trust model. *Applied Soft Computing*, 11(1):345 – 355.

[Trabelsi *et al.*, 2010] Trabelsi, C., BenJrad, A. et BenYahia, S. (2010). Bridging folksonomies and domain ontologies: Getting out non-taxonomic relations. *In Proceedings of the 10th IEEE International Conference on Data Mining Workshops, ICDM2010*, pages 369–379, Sydney, Australia. IEEE Computer Society.

[Trabelsi *et al.*, 2011] Trabelsi, C., Jelassi, N. et BenYahia, S. (2011). Auto-complétion de requêtes par une base générique de règles d'association triadiques. *In Proceedings of The 8th French Information Retrieval Conference, CORIA' 11*, pages 9–24.

[Vendramin *et al.*, 2010] Vendramin, L., Campello, R. J. G. B. et Hruschka, E. R. (2010). Relative clustering validity criteria: A comparative overview. *Stat. Anal. Data Min.*, 3(4):209–235.

[Vitiello, 2002] Vitiello, E. (July 2002). *Trust: A module for defining trust relationships in FOAF*. http://www.perceive.net/schemas/20021119/trust/default.htm.

[Wang et Wu, 2011] Wang, G. et Wu, J. (2011). Multi-dimensional evidence-based trust management with multi-trusted paths. *Future Generation Computer Systems*, 27(5):529–538.

[Xie et Beni, 1991] Xie, X. L. et Beni, G. (1991). A validity measure for fuzzy clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(8):841–847.

[Yao, 2004] Yao, W. (2004). Trust management for widely distributed systems. *PhD thesis,the University of Cambridgee, Jesus College,Computer Laboratory, Cambridge, United Kingdom.*

[Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338 – 353.

[Zajonc, 2001] Zajonc, R. B. (2001). Mere Exposure: A Gateway to the Subliminal. *Current Directions in Psychological Science*, 10:224–228.

[Zhang *et al.*, 2006] Zhang, Y., Chen, H. et Wu, Z. (2006). A social network-based trust model for the semantic web. *In Proceedings of the Third International Conference on Autonomic and Trusted Computing*, ATC'06, pages 183–192, Berlin, Heidelberg. Springer-Verlag.

[ZHAO, 2012] ZHAO, Q. (2012). Cluster validity in clustering methods. *PhD thesis,University of Eastern, Finland.*

[Ziegler et Golbeck, 2007] Ziegler, C.-N. et Golbeck, J. (2007). Investigating interactions of trust and interest similarity. *Decision Support Systems journal*, 43(2):460–475.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Nouveaux Modèles pour la Gestion de Confiance et de Réputation dans les Réseaux Sociaux.

Mots clés : Réseaux sociaux; confiance directe; confiance indirecte; réputation

Résumé : La nature ouverte et décentralisée des réseaux sociaux en ligne les rend vulnérables à l'apparence des utilisateurs malveillants. Par conséquent, la protection des utilisateurs éventuels des réseaux sociaux contre ceux malveillants est très primordiale. Ainsi, la notion de la confiance est importante car elle impacte fortement la décision d'un utilisateur de choisir les autres utilisateurs avec lesquels il peut interagir. Cette thèse a pour but de fournir des méthodes de gestion de confiance et de réputation efficaces permettant d'évaluer la confiance et la réputation des utilisateurs des réseaux sociaux. Nous avons proposé quatre contributions. La première contribution permet d'extraire les intérêts des utilisateurs où les informations contextuelles, sociales et complexes sont prises en compte. Ensuite, nous proposons une approche de gestion de la confiance directe appelée IRIS. La troisième contribution dans cette thèse est la proposition d'un modèle d'inférence de la confiance, TISoN, pour la gestion de la confiance indirecte dans les réseaux sociaux. En effet, il est nécessaire d'évaluer la confiance entre deux utilisateurs n'ayant pas une connaissance directe. La quatrième contribution consiste à proposer deux algorithmes de gestion de réputation dans les réseaux sociaux. Nous avons proposé d'abord, un nouvel algorithme de classification des utilisateurs exclusif basé sur le réseau de confiance, appelé RepC. Ensuite, nous avons proposé l'algorithme FCR, une version floue de RepC. Les approches proposées ont été validées sur des bases de données réelles et aléatoires. Les résultats expérimentaux ont démontré que nos algorithmes sont les meilleurs par rapport à ceux proposés dans la littérature.

Title : Computational Models of Trust and Reputation in Online Social Networks

Keywords : Social networks; direct trust; indirect trust; reputation

Abstract: The open and decentralized nature of the OSNs makes them vulnerable to the appearance of malicious users. Therefore, prospective users face many problems related to trust. Thus, effective and efficient trust evaluation is very crucial for users' decision-making. It provides valuable information to OSNs users, enabling them to make difference between trustworthy and untrustworthy ones. This thesis aims to provide effective and efficient trust and reputation management methods to evaluate trust and reputation of OSNs users, which can be divided into the following four contributions. The first contribution presents complex trust-oriented users' contexts and interests' extraction, where the complex social contextual information is taken into account in modeling. Second, we propose the IRIS (Interactions, Relationship types and Interest Similarity) trust management approach allowing the generation of the trust network and the computation of direct trust. The third contribution of this thesis is trust inference in OSNs. In fact, it is necessary and significant to evaluate the trust between two participants whom have not direct interactions. We propose a trust inference model called TISON (Trust Inference in Social Networks) to evaluate Trust Inference within OSNs. The fourth contribution of this thesis consists on the reputation management in OSNs. To manage reputation, we proposed two new algorithms. We introduce a new exclusive algorithm for clustering users based on reputation, called RepC, based on trust network. In addition, we propose a second algorithm, FCR, which is a fuzzy extension of RepC. For the proposed approaches, extensive experiments have been conducted on real or random datasets. The experimental results have demonstrated that our proposed algorithms generate better results, in terms of the utility of delivered results and efficiency, than do the pioneering approaches of the literature.