Comparative Trust Management with Applications: Bayesian Approaches Emphasis

Krishnaprasad Thirunarayan, Pramod Anantharam, Cory Henson, Amit Sheth
Ohio Center of Excellence in Knowledge-enabled Computing - Kno.e.sis
Department of Computer Science and Engineering
Wright State University, Dayton, OH-45435
{tkprasad, pramod, cory, amit}@knoesis.org

ABSTRACT

Trust relationships occur naturally in many diverse contexts such as collaborative systems, e-commerce, interpersonal interactions, social networks, and semantic sensor web. As agents providing content and services become increasingly removed from the agents that consume them, the issue of robust trust inference and update becomes critical. There is a need to find online substitutes for traditional (direct or face-to-face) cues to derive measures of trust, and create efficient and robust systems for managing trust in order to support decision-making. Unfortunately, there is neither a universal notion of trust that is applicable to all domains nor a clear explication of its semantics or computation in many situations. We motivate the trust problem, explain the relevant concepts, summarize research in modeling trust and gleaning trustworthiness, and discuss challenges confronting us. The goal is to provide a comprehensive broad overview of the trust landscape, with the nitty-gritties of a handful of approaches. We also provide details of the theoretical underpinnings and comparative analysis of Bayesian approaches to binary and multi-level trust, to automatically determine trustworthiness in a variety of reputation systems including those used in sensor networks, e-commerce, and collaborative environments. Ultimately, we need to develop expressive trust networks that can be assigned objective semantics.

KEYWORDS: trust vs. reputation, trust ontology, gleaning trustworthiness, trust metrics and models (propagation: chaining and aggregation), social and sensor networks, collaborative systems, trust system attacks, beta-PDF, Dirichlet distribution, binary and multi-level trust.

1. INTRODUCTION

Trust relationships occur naturally in many diverse contexts such as collaborative systems, e-commerce, social interactions, (semantic) social networks, mobile ad hoc networks (MANETs), distributed systems, decision-support systems, and (semantic) sensor web. As the connections and interactions between humans and/or machines (collectively called agents) evolve, and as the agents providing content and services become increasingly removed from the agents that consume them, and as miscreants attempt to attack existing infrastructure, the issue of robust trust inference and update (collectively called *trust management*) becomes critical. There is no dearth of trust frameworks in the literature to represent and reason with trust information. However, use of these frameworks for trust computation in practice requires specification of how to glean (direct) trustworthiness values, determination of context-based trust thresholds, and justification of rules for (indirect/inferred) trust propagation (via chaining and aggregation), in the application context [1][2]. Even though trust is central to meaningful collaboration among machines, or among humans, or between machines and humans, there is neither a universal notion of trust that is applicable to all domains nor a clear explication of its semantics or computation in many situations. Furthermore, because Web, social networks and sensors often provide complementary and overlapping information about an activity or event that are critical for overall situational awareness, there is a unique need for developing an understanding of and techniques for managing trust that span all these information channels.

Towards filling some of the gaps in automating trust inference, we studied several Bayesian approaches to trust that are broadly applicable to machine and social sensor networks, MANETs, recommender systems, collaborative environments, etc. Ironically, a large number of approaches that develop Bayesian basis for trust using Beta probability distribution function (Beta-

PDF) do not coincide when we look at the details. Our comparative analysis of several notable approaches to trust formation and evolution revealed that there are significant differences in the nature of trust information these frameworks represent, in the details of trust composition rules, and their overall robustness. Furthermore, there are a number of situations where binary trust is restrictive and graded trust level information (e.g., poor, fair, good, very good, excellent) is available. So we discuss the generalization to multi-level trust and review several practical applications. We also discovered errors in an existing formalization of multi-level trust evolution, which we use to better motivate the mathematical basis for multi-level trust. Specifically, we summarize our findings and discuss formalization of multi-level trust based on Dirichlet distribution that generalizes Bayesian approaches to binary trust based on Beta-PDF and overcomes the capricious behavior of some of the existing Bayesian approaches to multi-level trust. To elucidate our approach, we present an algorithm for computing trust evolution on concrete examples that is intuitively satisfactory and that is robust with respect to well-known (trust system) attacks. The evaluation based on example traces obtained by experimenting with this algorithm seems more insightful than the traditional simulation studies that seem to confirm the obvious aggregate behavior. We also discuss existing works that apply Dirichlet distribution for formalizing multi-dimensional trust and for collaboration.

The objectives of this work are: (i) to illustrate the nature of trust occurring in different domains to rationalize why there is no universal notion of trust; (ii) to explain the details of Bayesian approaches to binary and multi-valued trust for automatic trust computation (that is, gleaning *direct* trust from first-hand interactions and then composing them to obtain *indirect/inferred* trust); (iii) to provide a comparative analysis and distributed trust computation algorithm for Bayesian approaches to trust in the context of sensor networks, to underscore the inherent complexity and subtlety involved in formalizing trust; and (iv) to provide a comprehensive discussion of attacks on trust systems. *Specifically, this work constructively demonstrates that providing probabilistic basis to trust networks is still open to multiple interpretations, and substantiates how seemingly similar approaches differ from each other in non-trivial ways.* For completeness, we recapitulate the fundamental concepts and terminology used in the trust literature, explaining their inter-relationships and distinctions. Our work *complements* several existing surveys on trust and reputation systems such as [3], [4], [5], [6], [7], [8], [9], [10], and [11]. Specifically, Marsh [3] presents an informal, qualitative analysis of the general notion of trust, and then develops a theory of computational trust. Unfortunately, the formalization is hard to apply in practice because of the difficulties in estimating numerical values for various parameters required by it. Grandison and Sloman [4] discuss trust classification and illustrate policy-based trust management in the context of sharing Internet resources and services. Artz and Gil [5] categorize published trust work abstractly under policy-based trust, reputation-based trust, general models of trust, or as addressing trust in information sources. Josang et al. [6] explain various trust concepts and summarizes practical trust and reputation systems for e-commerce. Yu et al. [7] presents a survey of trust and reputation management systems in wireless communication. Golbeck et al. [12] and Golbeck [13] explore trust representation and reasoning in social networks, specifically, computation and application of binary and continuous trust ratings. In the context of sensor networks, Buchegger and Le Boudec [8] propose and analyze a message-level protocol (called CONFIDANT) that detects and isolates misbehaving sensor network nodes, in order to improve the robustness and the performance of dynamic network packet routing, while Momani and Challa [10] provide a broad survey of trust in network domain distinguishing between security and trust, and providing a description of attacks at the network and packet level. In contrast, we discuss fewer approaches but in more depth, and focus on attacks on the trust system. Hussain et al. [9] provide a short qualitative summary of four different approaches to trust that embody Bayesian networks, and point out their shared short comings. Our work is a more substantial analysis of the related approaches. The recently published work, Govindan and Mohapatra [11], is a comprehensive survey of trust computing methods and trust dynamics in MANETs. Specifically, it provides a broad coverage of trust literature and attacks as it relates to MANETs. However, our detailed comparative analysis of binary trust utilizing our trust ontology concepts in Section 5, the precise analysis of why Quercia et al.'s B-Trust approach to multi-valued trust is problematic, the detailed development of a satisfactory approach to multi-valued trust in Section 6, and the illustration of different trust application areas are *complementary* to Govindan and Mohapatra [11]. The current paper extends Thirunarayan and Anantharam [14] (which is a broad tutorial introduction to trust networks) with a comprehensive theory and implementation of multi-valued trust using Dirichlet distribution.

The paper is organized as follows: In Section 2, we provide examples to motivate the trust problem. In Section 3, we elucidate characteristics of trust and explain related concepts. In Section 4, we discuss our trust ontology. In Section 5, we summarize trust research by showing illustrative examples of how to glean trustworthiness. These results may be adapted for

different collaboration contexts. In Section 6, we further elaborate on the existing Bayesian Approaches to binary and multi-level trust, including using Dirichlet distribution, due to its practical importance and widespread use. We also discuss various applications. In Section 7, we recapitulate our findings.

2. MOTIVATION

We present real-life examples to underscore the fundamental nature of trust problem.

2.1. Trust in Multiple Domains

Interpersonal Networks

• *With which neighbor should we leave our children over the weekend when we are required to be at the hospital?*

• *Who should be named as a guardian for our children in our Will?*

Note that (i) there is uncertainty and incompleteness in our knowledge about the unraveling situation, (ii) there is not only an expectation of a good outcome but also concern about a bad outcome, and (iii) there is a need for immediate action. Furthermore, the threshold for trust in the second case is significantly higher than the threshold for the first case.

Social Networks

> *–SUBJECT: [TitanPad] Amit Sheth invited you to an EtherPad document.*
> *–CONTENT: View it here:*
> *http://knoesis.titanpad.com/200*

The first author received the above email purportedly from the collaborator. Is this a genuine request, or a trap? This doubt arose because, in the past, we have collaborated using only Google Docs, and TitanPad was unfamiliar, and there was an urgent need to edit the shared document.

Similarly, one always has a nagging feeling about clicking on http://bit.ly-URL, or about relying on a product review (when only a few reviews are present).

Sensor Networks

Given a weather sensor network-based prediction of a potential tornado in the vicinity of a city, should we mobilize emergency response teams ahead of time?

This really depends on how much trust we have in the reliability of sensor nodes and the collaborative nature of the task.

When a van's TCS (Traction Control System) indicator light comes on intermittently, is the indicator light faulty or the traction control system? Similarly, when a van's Check Engine light comes on, is indicator light faulty or the transmission?

This again depends on how various subsystem functions are monitored. In fact, in our van's case, the TCS indicator light and the transmission were faulty.

Summarizing Examples

Trust/reputation systems provide mechanisms for *soft security*, in contrast with authentication and access control mechanisms that constitute *hard security*. In MANETs, trust enables dynamic determination of trustworthy routes, improving throughput and robustness against malicious nodes. Note that secure key distribution/authentication does not obviate the need for trust inference in case an attacker is able to subvert security mechanisms and somehow enter the network. In sensor networks, trust enables improving overall reliability and avoiding misbehaving nodes due to faults or transient vagaries of the environment. In cognitive radio networks, trust enables picking less noisy and less crowded channels. In e-Commerce, aggregated reputation promotes reward for honesty and penalty for deceit. In the context of Web, source trust can be crucial for result set ranking, data integration and conflict resolution. In collaborative environments, trust can be used to select, monitor and gauge suitability of a partner. Trust is also fundamental to interpersonal communication and social transactions.

In the context of applications that involve both humans and sensors systems, it is crucial to have trustworthy aggregation of all data and control actions. For example, the 2002 Uberlingen mid-air collision[1] occurred because the pilot of one of the planes trusted the human air traffic controller (who was ill-informed about the unfolding situation), instead of the electronic TCAS system (which was providing conflicting but correct course of action to avoid collision). See Investigation Report AZ001-1-2, German Federal Bureau of Aircraft Accidents Investigation, 2004.

[1] http://en.wikipedia.org/wiki/2002_Uberlingen_mid-air_collision (accessed 10/23/2012)

2.2. Common Issues Related to Trust

Trust inference is necessary for action in diverse situations, subject to uncertainty and potential for loss. In all the above examples and collaborative tasks, we have a *Trustor* who must choose whether and how much to trust a *Trustee,* an *Action* by which the trustor is choosing to be vulnerable to the trustee based on an assessment of trustee's nature, and a *Context* in which the potential negative consequences of betrayal outweigh any perceived positive results [15]. Besides context, time also plays an important part in determining and updating trust due to the dynamic nature of interactions and behavior evolution.

There are two sides to trust management: Trustor assesses trustee for dependability in a given context and then decides to act accordingly. On the other hand, trustee tries to come across in a positive light about its suitability, reliability, and quality of service.

In general, we track trust in order to: (i) predict future behavior; (ii) incentivize "good" behavior and discourage "bad" behavior; and (iii) detect malicious entities.

2.3. Distinguishing Issues Related to Trust Networks

We will use the term *machine networks* to lump together MANETs, sensor networks, cognitive radio networks, etc., *social networks* to lump together social media, social sensors/crowd-sourcing, e-commerce rating/review systems, recommender systems, collaborative environments, etc., and *interpersonal networks* to refer to people to people interactions. In interpersonal networks, trust is often subjective, while in machine networks, trust can be given an objective basis and approximated by trustworthiness. Social networks straddle these two extremes, so trust issues span the whole gamut as it applies to them. For example, a trustor may not know a trustee in a social sensing context (cf. Twitter), while a trustor may know trustee's relative level of competence and honesty in other contexts (cf. Facebook). Here, we do not address the issue of trust in the context of the web of documents (HTML Web) and the web of data (Semantic Web).

There is a large body of work proposing different trust frameworks for pervasive computational trust management that must be instantiated and customized for each specific application. In (Facebook-like) social networks and interpersonal networks, the justification for taking this framework-based approach is to accommodate subjectivity in dealing with uncertainty and varied context of use, due to differences in trustor's experiences, intensions, trust thresholds (that depend on risk tolerance and mitigating factors such as warranties and insurance), circle of recommenders, and alternative sources to satisfy the goal. Therefore, by its very nature, social interaction-based interpersonal trust is not amenable to automatic trust assessment, even though manual analysis can be used to elucidate important factors that influence decision making. On the contrary, in machine networks and in social networks that require determination of trustworthiness entirely from the overt behavior of a trustee, we need to pursue formalization of trust metrics and inferences that take into account context-dependent trust thresholds. Interaction-based trust inference can allow identification of nodes that are faulty, misbehaving (due to environmental effects) or malicious in machine networks, and sources that are prejudiced, ignorant, or malicious in crowd-sourced social networks.

3. TRUST-RELATED CONCEPTS

We recapitulate well-known definitions of trust concepts and briefly discuss their interrelationships.

3.1. Trust Definitions

(Psychology slant) *Trust* in a person is a commitment to an action based on a belief that the future actions of that person will lead to good outcome [16].

(Probability slant) *Trust* (or, symmetrically, distrust) is a level of subjective probability with which an agent assesses that another agent will perform a particular action, both before and independently of such an action being monitored [17].

3.2. Trustworthiness Definition

(Psychology Slant) *Trustworthiness* is a collection of qualities of an agent that leads them to be considered as deserving of trust from others (in one or more environments, under different conditions, and to different degrees) [15].

(Probability slant) *Trustworthiness* is the objective probability that the trustee performs a particular action on which the interests of the trustor depend.

3.3. Trust versus Trustworthiness

Trust disposition depends on potentially quantified trustworthiness qualities and context-based trust threshold. For example, in the context of trusting strangers, people in the West will trust for lower levels of trustworthiness than people in the Gulf [18].

Trustworthy system produces expected behavior and is not susceptible to subversion. In other words, trustworthiness is the assurance that a system will perform as expected for sustained collaboration despite environmental disruptions, human and operator errors, hostile attacks, and implementation errors.

3.4. Trust versus Reputation and Security

(Community-based) *reputation* is the community or public estimation of standing for merit, achievement, reliability, etc.[2] Alternatively, *reputation* is the opinion (or a social evaluation) of a community toward a person, a group of people, or an organization on a certain criterion [3]. (Cf., Brand-value, PageRank [19], eBay profile, etc.)

Reputation can be a basis for trust. However, they are different notions, as illustrated by Josang et al. [6].

> *I trust you because of your good reputation.*
> *I trust you despite your bad reputation.*
> *Do you still trust Toyota brand?*

Trust is local and subjective; reputation is global and objective. Security refers to resistance to attacks (on the trust management system).

Reputation is overloaded in that community-based reputation differs from temporal reputation-based process. The latter elicits trust for sustained good behavior over time.

4. TRUST ONTOLOGY

A trust network is a data structure that abstracts and formalizes information relevant to describing trust relationships. A trust inference algorithm computes trustworthiness information implicit in a trust network.

Consider the following fragment of English involving trust information for delegating work or responsibility, and its abstract representation in the form of a trust network shown in Figure 1 [1].

• Alice (A) trusts Bob (B) for recommending good car mechanic.
• Bob trusts Dick (D) to be a good car mechanic.
• Charlie (C) does not trust Dick to be a good car mechanic.
• Alice trusts Bob more than Charlie, for recommending good car mechanic.
• Alice trusts Charlie more than Bob, for recommending good baby sitter.

Formally, a *trust network* is a node-labeled, edge-labeled, in-edge ordered, directed graph data structure. In general, the *semantics of trust* can be captured by specifying the meaning of the trust network in terms of how "network elements and trust values" relate to or compose with each other using logic, probability theory, statistics, or path constraints. *Inference algorithms* are efficient graph-based procedures for querying or determining trust values.

In order to better understand trust concepts and relate various approaches to trust in the literature, we have developed a simple ontology of trust [20]. The trust ontology, as shown in Figure 2, is more a taxonomy than a formal semantic specification. However, we can specify the semantics of trust in a rigorous manner by formalizing trust inferences sanctioned by a trust network as shown later. Our goal here is to provide a unified vocabulary to abstract, compare and contrast different approaches. The trust ontology describes in more detail the primitive trust information (*trust metric*) carried by each edge label. Specifically, it captures the type, the value and the means to acquire the value for each edge. Trust inference algorithms (*trust models*) deal with how to compose primitive trust values associated with edges to obtain aggregated trust values over paths and subgraphs as discussed in Section 5.3. The trust relationship is a 6-tuple:(*trustor, trust type, trust value, trust scope, trust process, trustee*), where, *trust type* represents the nature of trust relationship, *trust value* quantifies the trustworthiness for comparison, *trust scope* represents the applicable context for trust, and *trust process* represents the method by which the trust value is created and maintained. See Figures 2 and 3 for details.

Trust Type: Trust type specifies the nature of the trust relationship. There are two trust types, referral trust (trust in belief) and functional/non-functional trust (trust in performance).

- *Referral Trust* (trust in belief) – Agent a1 has referral trust in agent a2 if a1 trusts a2's ability to recommend another agent.
- *(Non-)Functional Trust* (trust in performance) – Agent a1 has functional (dis)trust in agent a2 if a1 (dis)trusts agent a2's ability to perform an action.

[2] Dictionary.com
[3] Wikipedia.com

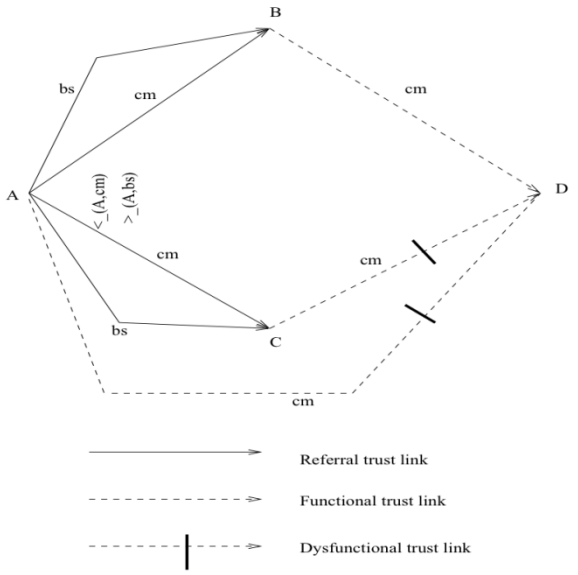

Figure 1: Example Trust Network

Trust Value: Trust value quantifies trust. This can be achieved using star rating, numeric rating, or partial order.

Traditionally, trust between users is modeled as a real number in [0,1] or [-1,1]. This facilitates trust computation, but is too fine-grained and imposes a total order. As stated by Guha et al. [21]: *While continuous-valued trusts are mathematically clean, from the standpoint of usability, most real-world systems will in fact use discrete values at which one user can rate another*. For instance, users often rate other users (such as vendors and reviewers) using star ratings. Epinions, provides a qualitative way of adding other users to a trust circle. Epinions, Ebay, Amazon, Facebook, etc. all use small sets for (dis)trust/rating values. We have formalized trust in terms of partial orders (that emphasizes *relative* trust) [1].

Trust Scope: Trust scope captures the context for which the trust information is applicable. We usually trust different agents for different subject matter or activity. For example, from Figure 1, Alice trusts Bob within the scope of recommending a good car mechanic.

Trust Process: Trust process specifies how trust values between pairs of agents are computed and is applicable to both primitive edges and composite paths.

- Trust process for primitive edges (i.e. for functional and referral edges):
  - *(Temporal) Reputation* – Trust values are computed based on past behavior over time.
  - *Policy* – Trust values are computed based on explicitly stated constraints.
  - *Evidence* – Trust values are computed based on seeking and verifying evidence.
  - *Provenance* – Trust values are computed based on lineage information.
- Trust process for composite edges (for admissible paths):
  - Trust values are determined via propagation (chaining and aggregation) specified as part of the trust model.

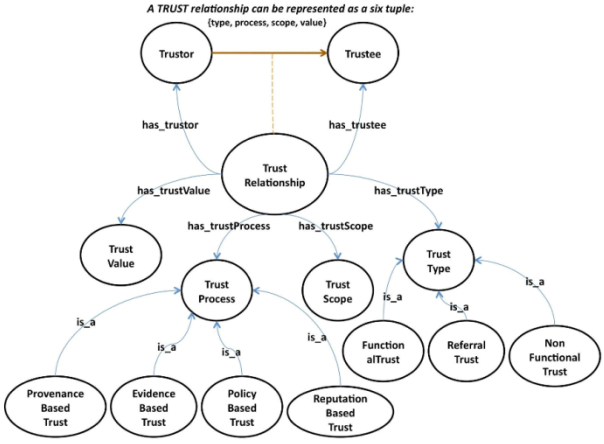

Figure 2: Trust Ontology

To provide a unified illustration of the trust processes consider hiring of a Search Engineer. A temporal reputation-based trust process is exemplified by the use of past job experience. A policy-based trust process can use scores on screening tests. An evidence-based trust process can use multiple interviews (phone, on-site, R&D team) for assessing the candidate's merits. A provenance-based trust process can consider the University from which the applicant graduated.

According to Mayer et al. [22], trust is a function of a trustee's perceived trustworthiness and of the trustor's propensity to trust. The trustor's propensity/disposition to trust, which is their willingness to be vulnerable, is both scope/context dependent, and highly subjective. For instance, Paul English [4] mentions four qualitative interpersonal trust dispositions: (i) *Suspicious still*: "Don't ever trust anyone, even after they have done something nice." (ii) *Suspicious until*: "Don't trust anyone until they prove themselves." (iii) *Trust until*: "Trust people until they screw up." (iv) *Trust still*: "Trust people even after they make mistakes, sometimes even when they hurt you."

[4] http://paulenglish.com/trust.html (accessed 10/23/2012)

In the rest of the paper, *we use this trust ontology to understand the abstract similarities and concrete differences among various approaches to trust, and to organize them*. For illustrative purposes, consider the following examples. *Trust type* is at the core of comparing and contrasting approaches to trust in sensor networks as discussed in detail in Section 5.1.3, especially because different approaches represent and reason with functional and referral trusts differently. *Trust values* take various forms as shown in Section 5, and require different reasoning strategies. Social networks and ecommerce sites use totally ordered discrete trust values (e.g., Golbeck [13], Amazon product and seller ratings), while Thirunarayan [1] proposes an alternative generalization to partial orders. In sensor networks, a trust value usually ranges over the unit interval [0,1] (e.g., [23][24][25]), while Josang [26] proposes the alternative generalization as a triple of values, standing for (belief, disbelief, uncertain), summing up to 1. *Trust scope* can be used to abstract and unify a number of approaches. Josang et al. [6] can be viewed as motivating different trust scopes relevant to understanding trust in ecommerce recommender systems, while Winkler [27] can be viewed as motivating different trust scopes relevant to virtual environments. *Trust processes* allow relating reputation systems used by ecommerce sites and reputation systems for sensor networks. Specifically, ecommerce sites aggregate trust in a vendor from different agents, while, in sensor networks, trust is gleaned by interacting with a sensor node over a period of time. These two approaches are logically distinct ways of aggregating trust that can be unified under the notion of trust process and in fact formalized similarly. In what follows, we use and illustrate the trust ontology concepts to organize and analyze various approaches to trust in different application areas.

5. GLEANING TRUSTWORTHINESS: ILLUSTRATING APPLICATION DOMAINS

We illustrate how to glean trustworthiness in different contexts. Direct trust, associated with trust edges, refers to trust determined using firsthand experiences (possibly over a period of time), while indirect trust, associated with trust paths, refers to trust determined using experiences of others via referrals [1][2]. Also note that, in spite of the distinctions articulated between trust, trustworthiness, and reputation in Section 3, we have deliberatively used the terms 'trust', 'trustworthiness' and 'reputation' interchangeably. This is to conform to the conventional overloaded use of the terms in the literature whose various senses can be easily disambiguated from the context.

Section 5.1 details how direct trust, both functional and referral, can be determined using a large number of observations through reputation-based process. Sections 5.1.1 and 5.1.2 describe the role of Beta-PDF in formalizing trust. Section 5.1.3 describes the various attacks that can befall a trust system. In order to illustrate the subtleties involved in trust computations, Section 5.1.4 shows how three seemingly similar approaches for the same problem, which are based on the same mathematical framework, can actually differ significantly in the trust inferences that they sanction. This underscores the difficulties in developing a universal notion of trust due to "clash of intuitions" even in a specific domain, and our analysis brings to fore the precise nature of differences.

Section 5.2 details how direct trust is determined using a policy-based process. For illustrative purposes, we cite several informal examples from Grandison and Sloman [4] and sketch automatic approaches used to glean trustworthiness of a Wikipedia article and a Web site.

Section 5.3 discusses how direct functional/referral trust among interacting users can be composed to infer indirect trust among users that have not interacted so far (and so lack firsthand experience). Our summary abstracts from a large number of trust propagation frameworks available in the literature.

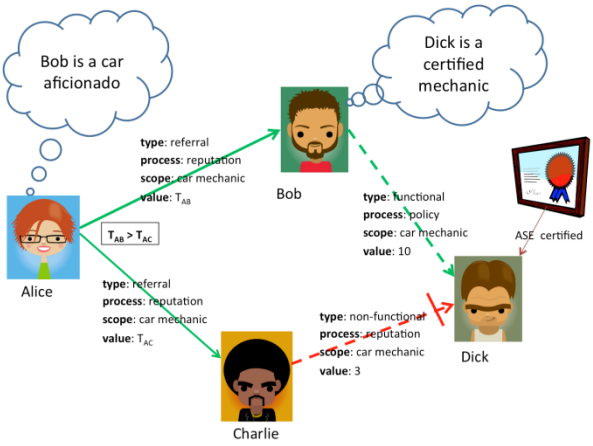

Figure 3: Example illustrating trust ontology

5.1 Direct Trust: Reputation-based Process

Direct trust can be inferred from a large number of observations made in two orthogonal ways: *over a period of time* or *by several agents*. Quantitative values for referral and functional trust in MANETs and sensor networks can be obtained using temporal reputation-based process. Both qualitative and quantitative information for referral and functional trust in product

rating systems can be obtained using community reputation-based process. We now motivate and discuss the Bayesian approach to formalizing reputation-based process that is in wide use.

5.1.1. Desiderata for Trustworthiness Computation Function

Initialization Problem: How do we get initial trust value?
Update Problem: How do we reflect the observed behavior in the current value dynamically?
Trusting Trust Issue: How do we mirror uncertainty in our estimates as a function of observations?
Efficiency Problem: How do we store and update values efficiently?

5.1.2. Beta Probability Density Function (PDF)

Beta-PDF provides a satisfactory mathematical foundation for reputation-based systems. Specifically, it formalizes prediction of trustworthiness probability from a sequence of binary events. We briefly review Beta-PDF, its role and benefits, below.

Let x be the probability of a binary event. If the prior distribution of x is uniform, then the Beta-PDF gives posterior distribution of x after observing α-1 occurrences of event with probability x and β-1 occurrences of the complementary event with probability (1-x).

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,du}$$

$$= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\,x^{\alpha-1}(1-x)^{\beta-1}$$

$$= \frac{1}{\mathrm{B}(\alpha,\beta)}\,x^{\alpha-1}(1-x)^{\beta-1}$$

$$\mathrm{E}(X) = \frac{\alpha}{\alpha+\beta}$$

$$\mathrm{E}(X^2) = \frac{\alpha(\alpha+1)}{(\alpha+\beta)(\alpha+\beta+1)}$$

$$\mathrm{Var}(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

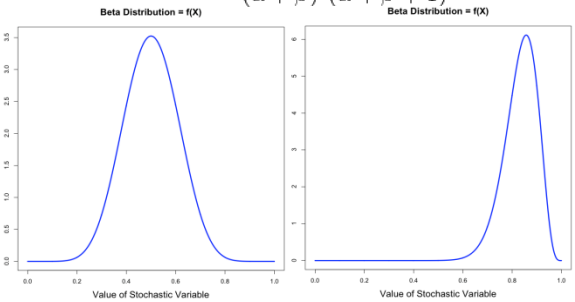

Figure 4: Beta-PDF(α=10;β=10) and Beta-PDF(α=25,β=5)

Specifically, let a (potentially unfair) coin have probability x of coming up with heads, and probability (1-x) of coming up with tail. Suppose we perform (r + s) coin tosses and the coin turns up with heads r times and with tails s times. Then the Beta-PDF[5] with parameters (r+1, s+1) provides the *best estimate* of the distribution of the probability x given these observations. Figure 4 depicts two example Beta-PDFs – one for (r,s) = (9,9) and another for (r,s) = (24,4).

In general, dynamic trustworthiness of a sensor or a vendor can be characterized using Beta-PDF Beta(α,β) gleaned from total number of correct (supportive) r = (α-1) and total number of erroneous (opposing) s = (β-1) observations so far, and the overall trustworthiness (reputation) can be equated to its mean: α/(α+β). *The Beta-PDF is intuitively satisfactory, mathematically precise, and computationally tractable, for formalizing direct trust from a collection of observations.* Specifically, it addresses all our requirements as follows:

Initialization Problem: It assumes that all probability values are equally likely.
Update Problem: It updates (α, β) by incrementing α for every correct (supportive) observation and β for every erroneous (opposing) observation.
Trusting Trust[6] *Issue*: It peaks at the mean. The variance diminishes with the number of observations.
Efficiency Problem: It stores/updates only two numbers.

We have developed an application to determine trust in weather sensor data and inferences based on them using the Mesowest [7] Weather Dataset for ~800 stations collected for a blizzard during 4/1-6/03. We used quality flags (OK, CAUTION, SUSPECT) associated with observations from a sensor station over time to derive reputation of a sensor by applying Beta-PDF [28]. The demo located at [29] is a visualization of the trust evolution.

5.1.3. Comparative Analysis of Bayesian Approaches to Binary Trust

We discuss details of several Bayesian approaches to binary trust based on Beta-PDF derived from experience[8] sequences and evaluate their robustness with respect to

---

5 http://en.wikipedia.org/wiki/Beta_distribution (accessed 10/23/2012)

6 Ken Thompson's Turing Award lecture titled "Reflections on Trusting Trust"

7 http://mesowest.utah.edu/index.html (accessed 10/23/2012)

8 The term *experience* is used for equivalent terms such as action, event, observation, interaction, service, utility, satisfaction-level etc. Similarly, the term *success* and *failure* are used for good and bad respectively.

the following well-known security attacks. These approaches can potentially be adapted to determine the trustworthiness of a collaborating partner using a centralized or distributed system. This discussion is meant to clarify their similarities and differences.

a. *Ballot-stuffing attack*: Majority of the recommenders collude to unfairly promote the trustworthiness of an undeserving trustee.

b. *Bad-mouthing attack*: Majority of the recommenders collude to unfairly denigrate the trustworthiness of a victim.

c. *Newcomer and Sybil attacks*: In newcomer attack, a malicious trustee creates new identity to avoid detection by a trust system that tracks history of interactions. In Sybil attack, a malicious trustee creates multiple fake identities to exert undue adverse influence.

d. *Sleeper and On-Off attacks*: A malicious trustee acquires high reputation/trust by behaving well for long durations and then behaving bad intermittently. The sleeper attack is also called *betrayal attack*, and on-off attack is also called *inconsistency attack*.

e. *Conflicting behavior attack*: In conflicting behavior attack, the attacker uses "divide and conquer" strategy by providing conflicting recommendations on a trustee to multiple trustworthy sources. When a victim seeks recommendations from these trustworthy sources, which faithfully transmit the attacker's views, the victim ends up getting conflicting recommendations on the trustee, thereby causing it to incorrectly reduce its trust in a subset of trustworthy sources (recommenders). This hampers the overall "morale".

Denko-Sun's Approach for MANETs [24]: Direct (functional[9]) trust in a trustee by a trustor is based on the number of success experiences s and number of failure experiences f witnessed by the trustor, and indirect (referral[10]) trust via recommendations from nodes 1 through k is based on the total number of success experiences s^r and total number of failure experiences f^r reported by the recommenders. Cumulative trust is obtained by summing both direct and indirect counts as follows:

$$(s + s^r + 1) / (s + s^r + 1) + (f + f^r + 1)$$

$$\text{where } s^r = \sum_{i=1}^{k} {s_i}^r \text{ and } f^r = \sum_{i=1}^{k} f_i^r$$

[9] Functional trust a.k.a. trust in performance

[10] Referral trust a.k.a. trust in belief

Each node maintains, for each peer (and for the implicit context of packet forwarding), these numbers. In practice, to improve security, it separates direct experiences from recommendations (which are indirect experiences), and separates recommendations from different recommenders even though the recommender identity is ignored. As a result, it can weight direct experiences more heavily than recommendations and drop extreme recommendations, to improve robustness. This approach can overcome ballot-stuffing and bad-mouthing attacks if malicious recommenders are a minority. It cannot deal with sleeper and on-off attacks, Sybil and newcomer attacks, and conflicting behavior attacks because it does not track recommender identity.

Ganeriwal et al.'s Approach for Sensor Networks [23]: Recall that the (α,β) parameters associated with the Beta-PDF can be obtained from success experiences s and failure experiences f as $(\alpha,\beta) = (s+1,f+1)$. (s_j^{new},f_j^{new})-values to compute trust of trustor i in trustee j are obtained by combining (a) the direct experiences (s_j,f_j) by trustor i with trustee j, and (ii) the indirect experiences (s_j^k,f_j^k) by node k with trustee j weighted by (s_k,f_k), the direct experiences by trustor i with node k, using chaining/discounting rule given in [34] as shown below.

$$s_j^{new} = s_j + (2*s_k*s_j^k) / ([(f_k+2)+(s_j^k+f_j^k+2)] + 2*s_k)$$

$$f_j^{new} = f_j + (2*s_k*f_j^k) / ([(f_k+2)+(s_j^k+f_j^k+2)] + 2*s_k)$$

Note that, while computing indirect trust, this rule modulates the contribution of a recommendation in proportion to the trustworthiness of the recommender. In contrast, Denko and Sun [24] ignores recommender identity completely.

In Ganeriwal et al. [23], each trustor maintains, for each trustee (and for all experiences combined, irrespective of the context), the (s,f)-values. The approach does not distinguish between functional and referral trust (and hence, does not maintain separate context-indexed counts). However, it does modify recommendations from a node using the trust in the recommender as shown above. As a result, this approach can overcome ballot-stuffing and bad-mouthing attacks as proved in [23]. By decaying/forgetting these counts over time (using a multiplicative factor $d^{(t-t0)}$, where $0<d<1$ and t0 is the start time), it can be made robust to sleeper and on-off attacks. However, it cannot deal with Sybil and newcomer attacks, and conflicting behavior attack. In contrast with Denko and Sun [24], Ganeriwal et al. [23] approach does not distinguish between different contexts (including functional and referral trust) and derives indirect trust by chaining a pair of edges using the discounting rule of Josang and Ismail's Beta reputation system.

Sun et al.'s Approach for MANETs [24]: Each trustor maintains, for each trustee that it has experience with, two separate direct trust: functional (for packet

forwarding) and referral (for recommendations). In the absence of direct functional trust information in a trustee, it computes cumulative indirect functional trust by pooling multiple recommendations for the trustee, via paths obtained by chaining referral edges followed by a functional link. Sun et al. [24] makes at least four novel contributions among others: (i) It uses an information theoretic formulation to devise a non-linear map of trust probability in [0,1] to a trust value in [-1,+1], thereby amplifying the effect of changes to trust probability on the trust value at the extremes. (ii) It provides axioms for trust models and trust composition rules that satisfy these axioms, as explained in Section 5.3.3 and Figures 10, 11 and 12. Effectively, it learns a local trust network dynamically and reasons over it using chaining and aggregation rules, which makes it more general than the approaches in [23] [24] discussed earlier. Unfortunately, Sun et al. [24] does not unambiguously specify the details of trust computation for arbitrary networks. Furthermore, we observe that top-down view of trust propagation is non-local (that is, meaning of a node is not entirely determined by the meanings of their immediate neighbors). (iii) It provides algorithmic details of their implementation for MANETs and an experimental simulation of it [24]. (iv) It analyzes various attacks on trust networks in depth and evaluates robustness of their approach to these attacks. Specifically, it overcomes ballot-stuffing, bad-mouthing, sleeper and on-off attacks, but not Sybil and newcomer attacks (which requires key-based infrastructure to overcome), and conflicting behavior attack (which is susceptible to recommender trust vulnerability).

In general, to deal with Sybil attacks, an orthogonal mechanism to generate and verify Security Tokens[11] for authentication is necessary.

5.1.4. Illustration using a Minimal Example

In order to shed light on the qualitative and quantitative differences in the Bayesian approaches to trust discussed so far, we consider a simple trust network shown in Figure 5 that involves two functional edges (one between A and B labeled F(5,10) and another between B and C labeled F(25,5)) and one referral edge (between A and B labeled R(12,2)), where the pair of numbers refers to the number of success experiences and the number of failure experiences respectively. As explained later, this example is adequate to surface the differences in the expressive power of the aforementioned approaches.

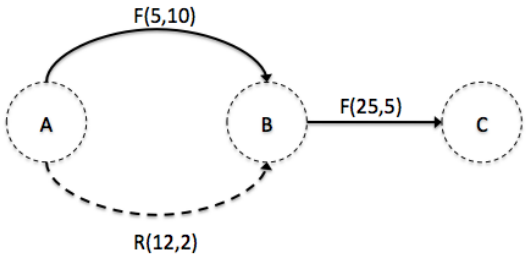

Figure 5: A generic, minimal trust network to compare and contrast different Bayesian approaches to binary trust

Denko-Sun's Approach [24]: This approach was proposed for the specific context of MANETs. A node infers functional trust in another node by aggregating its direct experiences and the direct experiences of its neighbors. For computing functional trust of A in B, we consider the direct edge F(5,10) obtaining the trust value as ((5+1)/(5+1+10+1)) = 0.35. For the purpose of inferring functional trust of A in C, we consider the composite edge F(5+25,10+5) obtaining the trust value as ((30+1)/(30+1+15+1)) = 0.66.

Ganeriwal et al.'s Approach [23]: In this approach, the context and type of interaction is not explicitly represented; only the total number of success experiences and the total number of failure experiences are retained. For the purposes of inferring trust of A in B, we consider two direct edges F(5,10) and R(12,2) to obtain the cumulative edge (5+12,10+2). This yields the net trust value as ((5+12+1)/(5+12+1+10+2+1)) = 0.58. For the purposes of inferring trust of A in C, we need to chain the direct trust T(5+12,10+2) with direct trust T(25,5) using Josang-Ismail discounting rule obtaining the effective number of success and failure experiences as T(s,f), where

$$s = 0 + (2*17*25) / ([(12+2) + (25+5+2)] + 2*17) = 10.625$$

$$f = 0 + (2*17*5) / ([(12+2) + (25+5+2)] + 2*17) = 2.125$$

The net trust value of A in C is
(10.625+1)/(10.625+1+2.125+1) = 0.79.

Sun et al.'s Approach [24]: This approach represents both functional and referral trust edges faithfully though it maps probability p in [0,1] to trust value in [-1,+1] using the following mapping:

T(trustee : trustor, action) =

if 0.5 <= p

then 1 – H(p) /* 0.5 <= p <= 1 */

else H(p) – 1 /* 0 <= p <= 0.5 */

where $H(p) = -p \log_2(p) - (1-p) \log_2(1-p)$

[11] http://en.wikipedia.org/wiki/Security_token (accessed 10/23/2012)

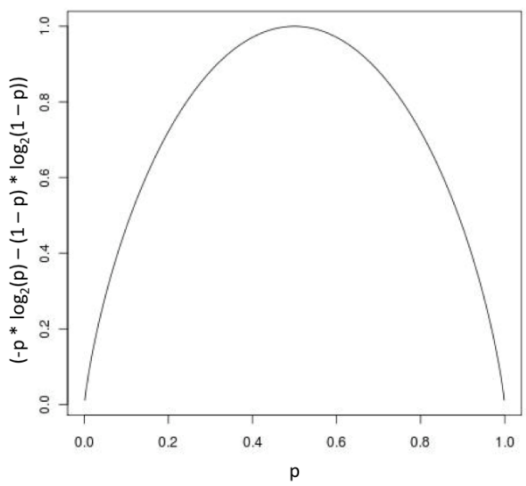

Figure 6: Uncertainty as a function of probability

This mapping provides an information theoretic interpretation of trustworthiness probability. Specifically, the probability values 0 and 1 imply certainty, while 0.5 implies absolute uncertainty. See Figure 6. This non-linear mapping amplifies the effect of changes to trust probability on the trust value at the extremes. That is, a change in probability near 0.5 has less effect on trust value than the same change near 0 or 1.

To determine functional trust of A in C, we need to chain the referral trust of A in B with functional trust of B in C, by multiplying their trust values. The referral trust probability of A in B is ((12+1)/(12+1+2+1)) = 0.81 and the functional trust probability of B in C is ((25+1)/(25+1+5+1)) = 0.81. Hence, the information-theoretic trust of A in B is 0.3 and that of B in C is 0.3 (obtained using the above mapping of trust probability in [0,1] to information-theoretic trust value in [-1,+1]). Furthermore, the composite trust of A in C is 0.3*0.3 = 0.09 (obtained using the product rule). See Table 1 for a comparative summary, which shows that differences can arise in the absence of expressive trust networks and an objective theory of trust.

Table 1. Comparison of functional trust values (from A to C in Figure 5)

	Denko-Sun's Approach *(prob. [0,1])*	Ganeriwal et al.'s Approach *(prob. [0,1])*	Sun et al.'s Approach *(inf. th.[-1,1])*
Functional trust value from A to C	0.66	0.79	0.09

5.2. Direct Trust: Policy-based Process

Grandison and Sloman [4] provides several informal examples of policy-based trust. Similarly, we routinely use training programs and certifications as the basis for inferring policy-based trust.

A general approach to trust assessment uses (i) domain dependent qualities for determining trustworthiness based on content (data) and on external cues (metadata), and (ii) domain independent mapping to trust values or levels through quantification and classification [30].

For example, trustworthiness of Wikipedia articles can be assessed based on domain dependent content-based quality factors such as references to peer-reviewed publications, proportion of paragraphs with citation, and article size, and metadata-based credibility factors such as author connectivity, edit pattern and development history, revision count, proportion of reverted edits (including normal reversals and those due to vandalism), mean time between edits, and mean edit length. Trustworthiness can be quantified in a domain independent way using *dispersion degree* score that captures the extent of deviation from the mean. For evaluation metric, normalized discounted cumulative gain (NDCG) can be used to compare ranking based on trust levels (determined from trustworthiness scores) to gold standard classification.

Another example is the estimation of a website's trustworthiness based on the criticality of data exchanged with it. Specifically, each of the following pieces of information carries with it different level of sensitivity: email address, username and password, phone number, home address, date of birth, social security number, etc. Intuitively, a piece of data is critical if it is exchanged with a small number of highly trusted sites [31].

5.3. Indirect Trust: Variety of Trust Metrics and Models

Trust between a pair of users/collaborators can be gleaned on the basis of their similarity, where similarity can be quantified in a number of ways such as using average difference in ratings, overall correlation of ratings, and correlation on extremes [32]. In fact, *collaborative filtering* uses similarity measures (such as profile-based, item-ratings based, item-category based) between a user and others to predict item-ratings by the user. This approach is items-agnostic and scales well over time with large number of items. However, it suffers from (i) *data sparsity problem* when a small number of items are common between users, (ii) *cold start user*

problem when a user has rated only a small number of items, and (iii) *copy-profile vulnerability* where an attacker can create a targeted-user-like profile to manipulate recommendations.

Trust-aware Recommender Systems (TaRS) use explicit/direct trust between users to predict implicit/indirect trust between users through chaining [33]. TaRS overcomes limitations of collaborative filtering because trust propagation improves coverage, a single trust edge from a new user can enable a user to inherit several "parental" recommendations, and fake identities are not trusted by an active user.

5.3.1. Trust Propagation Frameworks

There are a host of approaches in the literature that present *trust management frameworks* and formalize *trust propagation* along chained paths, *trust aggregation* from multiple sources, and *overriding* [1][34][21][35][36][16][26][36][37][38]. *However, in the absence of an objective semantics of trust, it is very difficult to evaluate various approaches to trust for validity. This is made worse by the lack of transparent examples of trust computations that show all the consequences of a specified approach.* In a number of situations, it is possible to reverse engineer framework parameters to reflect any desirable semantics of a trust network, making the comparison of frameworks even harder.

5.3.2. Trust Propagation Algorithms

Broadly speaking, trust propagation algorithms work on DAGs extracted from potentially cyclic trust networks and fall into two categories: top-down and bottom-up. In *top-down* approach, trust value for a source in a target is predicted by aggregating trust values in the target inherited from *source's "trusted" parents* weighted with trust value in the corresponding parent [2]. In *bottom-up* approach, trust value for a source in a target is predicted by aggregating trust scores in target inherited from *target's "trusted" neighbors* weighted with trust value in

(a) Same Interpretation (b) Different Interpretation

Figure 7: Comparative analysis example: top-down vs. bottom-up

the corresponding neighbor [38]. For instance, the two approaches cited above interpret Figure 7(a) similarly with q trusting s. On the other hand, they interpret Figure 7(b) differently with the top-down approach being ambiguous about q trusting s, while the bottom-up approach concludes that q distrusts s.

Figure 8 illustrates the TidalTrust algorithm where the trust computation is top-down and uses weighted averages. Specifically, T(E,Sink) = T(C,Sink) = 2, T(B,Sink) = (3*2+6*5)/(3+6) = 4, and T(Source,Sink) =(4*4+2*7)/(4+2)=5.

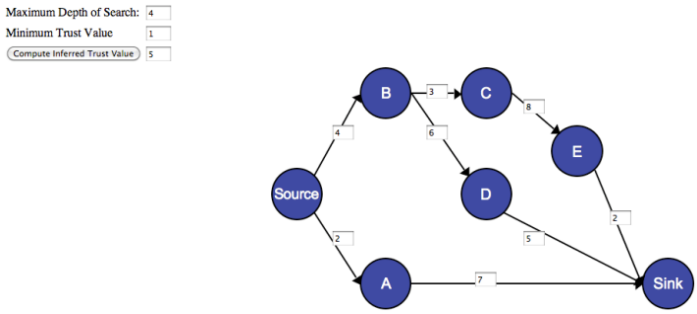

Figure 8: TidalTrust Trust Computation Example

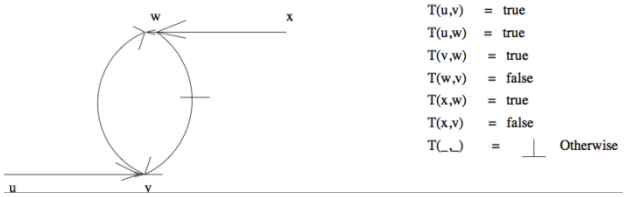

Figure 9: Cyclic Trust Network

Figure 9 shows a well-founded cyclic trust network and binary trust conclusions.

5.3.3. Trust Propagation Rules: Axioms for Trust Models

As explained in Section 5.1.3, Sun et al. [34] describes an interesting approach to trust computation by first providing an axiomatic basis for trust models and then providing concrete rules for combining trust values as reproduced below.

Rule 1: Concatenation propagation does not increase trust. For example, to satisfy Rule 1, one can use $T(A_1,C_1) = R_1 * T_2$ if $R_1 > 0$ and $T_2 > 0$.

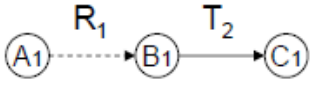

Figure 10: Illustration for Rule 1 - Chaining Trust

Rule 2: Multipath propagation does not reduce trust. For example, to satisfy Rule 2, one can combine the trust

values on the two paths as $T(A_2,C_2)$ = (*R1(R1*T2)*+**R1(R1*T2)**) / (*R1* + **R1**), where the *italicized* values refer to the upper path and **boldface** values refer to the lower path in case one wants to consider different trust values.

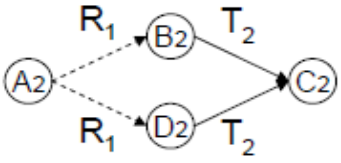

Figure 11: Illustration for Rule 2 - Aggregating Trust

Rule 3: Trust based on multiple referrals from a single source should not be higher than that from independent sources. That is, $T(A_1,C_1) <= T(A_2,C_2)$.

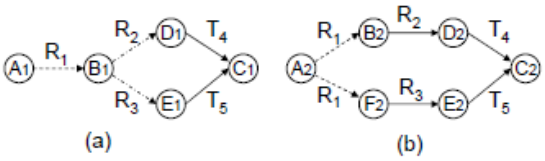

Figure 12: Illustration for Rule 3 - Propagating Trust

Unfortunately, the axioms have limited applicability and do not unambiguously specify trust computation over an arbitrary trust network.

Beta-reputation system [39] chains opinions o_1 and o_2 (where opinion o_i has three components [belief b_i, disbelief d_i, uncertainty u_i]) to obtain discounted opinion o_3 as $b_3 = b_1 * b_2$, $d_3 = b_1 * d_2$, and $u_3 = d_1 + u_1 + b_1 * u_2$.

6. A BAYESIAN APPROACH TO MULTI-LEVEL TRUST

Section 6 develops a Bayesian approach to multi-valued trust based on Dirichlet distribution. Section 6.1 motivates the need for formal underpinnings by showing the downside of an ad hoc approach to multi-valued trust. Section 6.2 then provides the relevant Bayesian theory (Section 6.2.1), the data structures used (Section 6.2.2) and the details of a robust trust computation algorithm (Sections 6.2.3 and 6.2.4) by adapting the B-Trust approach of Quercia et al. [40]. For clarity, Sections 6.2.5 illustrates the multi-valued trust inference algorithm on concrete examples, and Section 6.3 analyzes its robustness to well-known attacks. Section 6.4 succinctly depicts a comparative analysis of different approaches to multi-level trust, while Section 6.5 discusses the practical applications of multi-level trust. Section 6.6 covers application of trust to collaborative environments.

Quercia et al. [40] generalizes binary trust metric used so far to K-level discrete trust metric, where K refers to the number of trust /experience levels. This work is exemplary in the way it develops the entire approach, providing details of local data structures employed, trust formation, trust evolution, evaluation of security, and experimental simulation. Unfortunately, we discovered that the default initialization (that rightly captures complete ignorance of initial trust probability) and the given Bayesian trust evolution rules, which seem satisfactory when considered in isolation, destructively interfere with each other when used together. As a result, the trust probability vector remains *fixed* (incorrectly) in response to any experience sequence. The fundamental problem can be traced to the fact that traditional Bayes' rule computes a conditional probability on the basis of already provided two prior probabilities and one conditional probability, while in Quercia et al. [40], we are also required to dynamically learn the latter conditional probability. *Unless and until we find a satisfactory interpretation of an experience level in terms of its effect on trust distribution, and account for an experience level directly in terms of trust distribution, we will not have an acceptable/defensible model of trust.* After developing several ad hoc fixes, we discovered that founding multi-level trust metric evolution on Dirichlet distribution [12], a significant departure from the way Bayes' rule is used in Quercia et al. [40], yielded an approach that preserved its strengths, while simultaneously overcoming its limitations as discussed below. We also review other approaches to formalizing multi-level trust using Dirichlet distribution including applications to MANETs, e-commerce and collaborative environments.

6.1. Illustrating Limitations of B-Trust Approach using Examples

We recapitulate just enough details of Quercia et al. [40] not only to illustrate its capricious behavior but also to provide a roadmap for how to describe a trust framework and its implementation. Specifically, we focus on functional trust and skip referral trust, whose computation also exhibits similar behavior.

For a K-level trust metric, each node maintains locally a K-length *Direct Trust Vector* and a K x K *Direct Experience Matrix,* to store information about trust level probabilities and experience level counts respectively, for computing direct (functional) trust between a pair of peers for each context using Bayes' Rule, as described below:

Direct Trust Vector dtv: *Peers* × *Contexts* × *Peers* → *Probability-Vector*$_K$

[12] http://en.wikipedia.org/wiki/Dirichlet_distribution (accessed 10/23/2012)

That is, dtv(x,c,y) = $(d_1,d_2,\ldots,d_K)$ where d_i = Probability that x has direct trust at level i in y in context c. (By definition, $d_1 + \ldots + d_K = 1$.)

Direct Experience Matrix dem: Peers × Contexts × Peers → $Count\text{-}Matrix_{K \times K}$

That is, dem(x,c,y) = $((ec_{11},\ldots,ec_{1K}),\ldots,(ec_{K1},\ldots,ec_{KK}))$ where ec_{ij} = Count of x's experience at level j with y on the basis of direct trust at level i in context c.

To reflect complete ignorance via uniform distribution, we set the probability vector to $(1/K,\ldots,1/K)$ making all trust levels equally likely to start with, and we set all the elements of the matrix dem to the same value for uniformity (where the initial magnitude determines the duration of persistence of the bootstrapping phase and is irrelevant for the problem we wish to discuss).

Trust Update: According to Quercia et al. [40], the direct experience matrix is changed in response to new experiences, and the direct trust vector is recomputed to reflect these changes. The probabilities are updated by applying Bayes' rule, where DE refers to the current level of direct experience of x while interacting with y, with current trust level of DT in context c:

$$p(DE(x,c,y) = j,\ DT(x,c,y) = i)$$

$$= \quad p(DE(x,c,y) = j \mid DT(x,c,y) = i) * p(DT(x,c,y) = i)$$

$$= \quad p(DT(x,c,y) = i \mid DE(x,c,y) = j) * p(DE(x,c,y) = j)$$

Renaming p(DT(x,c,y) = i) as **prior-prob-for-xcy-i** and p(DT(x,c,y) = i | DE(x,c,y) = j) as **posterior-prob-for-xcy-i**, the equation can be rearranged as a Bayesian inference/update rule:

$$p(DE(x,c,y) = j \mid DT(x,c,y) = i) * \textbf{prior-prob-for-xcy-i} = \textbf{posterior-prob-for-xcy-i} * p(DE(x,c,y) = j)$$

$$\textbf{posterior-prob-for-xcy-i} = \textbf{prior-prob-for-xcy-i} * [\, p(DE(x,c,y) = j \mid DT(x,c,y) = i) \ / \ p(DE(x,c,y) = j) \,]$$

The quantity **posterior-prob-for-xcy-i** corresponds to the inferred probability that the direct trust of x in y is at level i subsequent to the direct experience at level j.

The exact computation of the various probabilities can be expressed in terms of the counts [39]. Note that the probability p(DE(x,c,y) = j | DT(x,c,y) = i) is determined by row i of the count-matrix dem, and the probability p(DE(x,c,y) = j) is determined as a prior probability weighted summation of each row's contribution.

$$p(DE(x,c,y) = j \mid DT(x,c,y) = i) \ = \ (ec_{ij} / \sum_{n=1}^{K} ec_{in})$$

$$p(DE(x,c,y) = j) = \sum_{n=1}^{K} d_n * p(DE(x,c,y) = j \mid DT(x,c,y) = n)$$

where, dtv(x,c,y) = $(d_1,d_2,\ldots,d_K)$ and

dem(x,c,y) = $((ec_{11},\ldots,ec_{1K}),\ldots,(ec_{K1},\ldots,ec_{KK}))$

Experience Update: In response to x's direct experience with y at level j, each entry in column j of dem is updated as follows: **for i in [1,K]: $ec_{ij} = ec_{ij} + dtv_i$.** (Equivalently, ec[i,j] = ec[i,j] + dtv[j].) The rationale seems to be that because only trust probability distribution (as opposed to exact direct trust level) is available, we can distribute the 1-unit of direct experience at level j among column j entries in proportion to the trust distribution, as a way to assimilate new experience. Unfortunately, for the given row-symmetric initializations (that is, dtv(x,c,y)=(1/K,…1/K) and dem=((1,…1),…,(1,…,1)), or for all i: $d_i = 1/K$ and for all i,j: $ec_{ij} = 1$) and the proposed row-symmetric updates, *the Bayesian inference leaves direct trust vector value unaltered irrespective of the level of experience*. For example, for K = 4 and initial trust vector dtv=(0.25,0.25,0.25,0.25), all experience level sequences [1,1,1], [1,4,1,4], [1, 1, 4, 4, 4, 4, 1, 1, 1], [2,3,2,3], etc. leave the trust vector unchanged [13] at (0.25,0.25,0.25,0.25), which is intuitively unsatisfactory. In other words, the nature of experience sequence has no impact on the trust level, which defeats the original purpose of trust evolution. The root cause of this unacceptable behavior is the fact that Bayesian inference is founded on existing background knowledge summarized in terms of two prior probabilities and one conditional probability, while, in the approach at hand, we are acquiring background knowledge from scratch as we go along. Our ad hoc fixes to the experience update issue allows us to evolve trust probability vector in ways that reflect experience faithfully qualitatively (e.g., poor quality (low-level) experience leads to distrust (low-level trust)), but these fixes do not pass muster when its quantitative behavior is scrutinized. Instead, we discovered that evolution of multi-level trust metric in response to multi-level experience can be formalized satisfactorily by rectifying the Bayesian foundation to be used as described below.

[13] This result can be argued purely on the basis of symmetry and induction as opposed to performing numerical calculations.

6.2. An Approach to Multi-level Trust Metric Evolution Based on Dirichlet Distribution

Josang and Haller [41] were the first to formalize and analyze a theory of multi-valued trust by generalizing binary trust metric [39][23][24][24] to K-level trust metric using Dirichlet Distribution[14] [42]. This approach evolves multi-valued trust in an intuitively satisfactory manner in response to experience sequences. K refers to the number of trust/experience levels. For example, Amazon's 5-star trust metric can be interpreted as signifying (very untrustworthy, untrustworthy, neutral, trustworthy, very trustworthy) or (very dissatisfied, dissatisfied, neutral, satisfied, very satisfied). The approach developed here formalizes a distributed, robust, lightweight, computational trust that takes into account context, subjectivity, and time, by adapting Quercia et al. [40]. Below we describe Dirichlet Distribution that serves as the mathematical foundation for multi-level trust (with emphasis on informal exposition of its formalization and applicability), local data structures employed for trust representation and reasoning, trust formation and evolution, and evaluation of its security. We also provide concrete examples of trust evolution rather than performing experimental simulation because the former provides greater insight into how trust evolves in response to an experience sequence, beyond mere sanity check on aggregate behavior that experimental simulations provide. As an aside, note that the entire development also provides a realistic (and pedagogically significant) illustration of the benefits of reusing a well-developed mathematical theory as opposed to inventing a novel approach that may have lurking idiosyncratic behavior.

6.2.1. Dirichlet PDF

Dirichlet PDF provides a satisfactory mathematical foundation for reputation-based systems that use multi-level trust metric. Let $x = (x_1,\ldots, x_K)$, where each x_i is the probability that the trust is at level i, for a K-level trust metric. By definition, $(x_1 + \ldots + x_K = 1)$. For example, if Amazon 5-star rating system has 50 people giving 5-stars, 20 people giving 4-stars, 5 people giving 3-stars, 5 people giving 2-stars, and 20 people giving 1-star, then the 5-level trust metric probability vector is (0.5,0.2,0.05,0.05,0.2). The probability of an experience sequence e1,...,em, to occur (where an experience at level e is a *realization* of trust at level e, that is, the result of the implicit trust at level e and leads to an explicit trust at level e) is $(x_{e1} * \ldots * x_{em})$. The total probability of experience-level sequences, with c_1 counts of level 1 experience, …, c_K count of level K experience, is:

$$(\prod_{i=1}^{K} x_i^{c_i}) * \frac{(c_1 + \ldots + c_K)!}{c_1! * \ldots * c_K!}$$

The α_i's = $c_i - 1$ are the associated Dirichlet distribution parameters. The first term corresponds to the probability associated with a single experience sequence satisfying the counts constraint, and the second term corresponds to the number of distinct experience sequences that satisfy the counts constraint (= total number of sequences / total number of duplicates).

The Dirichlet distribution, which is the PDF for $x = (x_1,\ldots, x_K)$ given parameters $(\alpha_1,\ldots,\alpha_K)$, is as follows (where, the Γ-function generalizes the factorial function for more general treatment):

$$f(x_1,\ldots,x_{K-1};\alpha_1,\ldots,\alpha_K) = \frac{\Gamma(\sum_{i=1}^{K}\alpha_i)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}(\prod_{i=1}^{K} x_i^{\alpha_i - 1})$$

$$\Gamma(n) = (n-1)! \text{ for all positive integers } n$$

Note that, for parameters $(\alpha_1,\ldots,\alpha_K)$ where each $\alpha_i - 1$ corresponds to the count of experiences at level i, the ratio $(f(x_1,\ldots x_{K-1}) / f(y_1,\ldots y_{K-1}))$ gives the relative likelihood of $(x_1,\ldots x_K)$ and $(y_1,\ldots y_K)$ describing the true state of affairs. [Note that because $(x_K = 1 - (x_1,\ldots,x_{K-1}))$, the plot of PDF in a K-dimensional space yields a (K-1) dimensional surface; specifically a (K-1) simplex, which is generalizes a line (K=2), a triangle (K=3), and a tetrahedron (K=4) to K-dimensions .]

If the prior distribution of x is uniform, then the Dirichlet family of distribution shown below gives posterior distribution of x after α_i-1 occurrences of level i experience with probability x_i, for each i in [1, K]: $f(x_1,\ldots x_K;\alpha_1,\ldots \alpha_{K-1})$.

In general, *a posteriori* PDF can be computed from a *priori* PDF to show that the *shape* (relative magnitudes of the various point probability densities) of the Dirichlet PDF is preserved by the outcomes conforming to multinomial distribution as follows (where un-subscripted letters c, x, α, etc. stand for vectors and the + operation stands for vector addition):

[14] http://en.wikipedia.org/wiki/Dirichlet_distribution (accessed 10/23/2012)

$$prob(x|c) = \frac{prob(c|x)prob(x)}{prob(c)}$$

$$f(x|c) \sim Multinomial(c|x) * Dirichlet(x|\alpha)$$

$$f(x|c) \sim \prod_{i=1}^{K} x^{c_i} * \prod_{i=1}^{K} x^{(\alpha_i - 1)}$$

$$f(x|c) \sim \prod_{i=1}^{K} x^{(\alpha_i + c_i - 1)}$$

$$f(x|c) \sim Dirichlet(x|\alpha + c)$$

In Bayesian statistics, this property is captured by the statement: *The Dirichlet distribution is a conjugate prior for the multinomial distribution*. This important property permits an efficient way to update the estimated distribution as a result of a new experience by just incrementing the corresponding parameter, without altering the structure/shape of the distribution. *If the prior distribution is different from the Dirichlet distribution, then it will be conceptually hard to comprehend and computationally inefficient to compute the posterior distribution, in general*. The fact that uniform distribution captures initial ignorance, and is a special case of the Dirichlet distribution, makes it a satisfactory starting point.

Figure 13 shows a visualization of Dirichlet distribution using six combinations of (α_1,α_2,α_3) (K=3) via projection [43]. The three diagrams in the top row represent symmetric, uniform distributions concentrated at (1/3,1/3,1/3) to varying degree. The variation in the color signifies that as we go from left to right, our confidence in the estimated (trust) probabilities is increasing because we have more samples (experiences) to back them up. The first two diagrams in the bottom row show asymmetric situations with concentration points being skewed to the dimensions with higher proportion of samples. The third diagram in the bottom row cannot be realized in our application, even though the formal machinery can deal with fractional α's.

The distribution of dynamic trustworthiness of a node can be characterized using Dirichlet-PDF(α_1,...,α_K) gleaned from total number of experiences (α_i-1) at level i, for all i in [1,K]. *The best estimate for the overall trustworthiness (reputation) is the mean vector (α_1/α_0,...,α_K/α_0), and the best estimate for our confidence in individual mean is its variance as shown below:*

$$\alpha_0 = \sum_{i=1}^{K} \alpha_i \; ; \quad \textbf{Mean}(x_i) = \alpha_i/\alpha_0 \; ;$$

$$\textbf{Variance}(x_i) = [\alpha_i*(\alpha_0-\alpha_i)] / [\alpha_0^2(\alpha_0+1)]$$

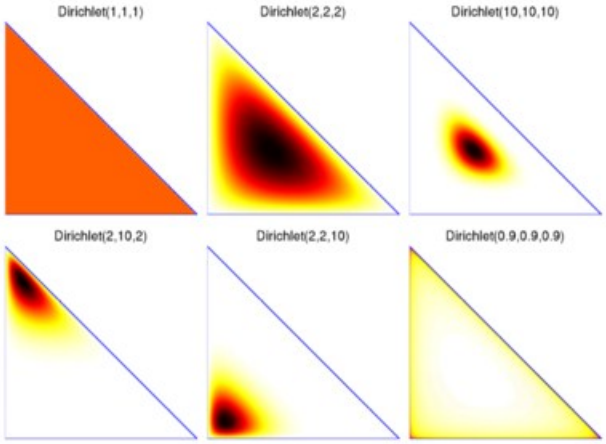

Figure 13: Visualization of Dirichlet distribution: Six Examples

6.2.2. Local Data Structures

We describe the data structures that each trustor holds to store relevant information to compute direct (functional) and indirect (referral) trust in a trustee. (Note that trustor and trustee are of the same type Peers.)

(1) Each trustor maintains locally, for each trustee and each context, a Direct Trust Vector, which is a probability vector of length K.

Direct Trust Vector dtv: $Peers \times Contexts \times Peers \rightarrow Probability\text{-}Vector_K$

That is, dtv(px,c,py) = (d_1,d_2,…,d_K) where d_i = Probability that trustor px has direct trust at level i in trustee py in context c. (As expected, $d_1 + \ldots + d_K = 1$.)

(2) Each trustor maintains locally, for each trustee and each context, a Direct Experience Vector, which is a count vector of length K.

Direct Experience Vector (dev): $Peers \times Contexts \times Peers \rightarrow Count\text{-}Vector_K$

That is, dev(px,c,py) = (ec_1,…,ec_K) where ec_i = Count of trustor px's direct experience at level i with trustee py in context c.

(3) Each trustor maintains locally, for each trustee and each context, a Recommended Trust Vector, which is a probability vector of length K.

Recommended Trust Vector (rtv): $Peers \times Contexts \times Peers \rightarrow Probability\text{-}Vector_K$

That is, rtv(px,c,py) = (r_1,r_2,…,r_K) where r_i = Probability that trustor px has recommended trust at level i in trustee py in context c. (As expected, $r_1 + \ldots + r_K = 1$.)

(4) Each trustor maintains locally, for each trustee and each context, a Sent Recommendation Vector, which is a count vector of length K.

Sent Recommendation Vector (srv): Peers × Contexts × Peers → Count-Vector$_K$

That is, srv(px,c,py) = (sr_1,…,sr_K) where sr_i = Count of trustor px's received recommendations at level i in trustee py in context c. Note that the identity of a recommender is lost in the process of aggregating counts.

(5) *Initialization*: To reflect complete ignorance via uniform distribution, we set the probability vectors dtv and rtv to (1/K,…,1/K), and the elements of the count vectors dev and srv to (0,…,0).

6.2.3. Trust Formation

The overall trust vector can be obtained as a weighted combination of direct trust vector and recommended trust vector. The weight can be determined in terms of (i) confidence value, which is the variance of the vector elements from its mean, depicting intrinsic uncertainty $\Sigma(di - 1/K)^2/(K-1)$ and (ii) relative preference for direct experience over recommendations. The former component is objective, while the latter component is subjective. The trust decision required for action also depends on context-based trust threshold that takes into account subjective risk tolerance and mitigating warranties.

6.2.4 Trust Evolution

The direct trust vector should be updated for each new experience, and similarly, the recommended trust vector should be updated for each newly received recommendation. Because Dirichlet distribution is the conjugate prior of the multinomial distribution, we just maintain the counts of the direct experience and sent recommendations, and compute most likely estimate of direct trust probabilities and recommended trust probabilities respectively as shown. (For brevity, we focus only on computing direct trust. Computation of recommended trust is similar.)

Simple Scheme (Bag-based):

For a new experience at level i,

$dev(px,c,py) = (ec_1,...,ec_K)$ is updated to

$dev^{new}(px,c,py) = (ec_1,..., ec_i+1,...,ec_K)$

and the corresponding $dtv(px,c,py)$ is updated to

$dtv^{new}(px,c,py) = (d_1,d_2,...,d_K)$

where $di = ec_i+1 / (ec_1 + ... + ec_k+1)$ and

$dj = ec_j / (ec_1 + ... + ec_k+1)$

for each j in $[1,K]$ and $j =/= i$.

To improve the robustness of the trust management system, (i) the trust is aged by attenuating the counts with time to reduce the effect of past experiences, and (ii) the trust is skewed using differential weighting of counts, to penalize low-level [15] experience (cf. failure) much more than reward complementary high-level experience (cf. success).

Robust Scheme (Timed and Skewed Decay):

To incorporate differential aging of experience counts (to incorporate long term memory for low-level experience and short term memory for high-level experience), we can use a decay vector $(\lambda_1,...,\lambda_K)$, where $1 >= \lambda_1 >= ... >= \lambda_K > 0$, and the modified update rules:

For a new experience at level i,

$dev(px,c,py) = (ec_1,...,ec_K)$ is updated to

$dev^{new}(px,c,py) = (ec_1,..., ec_i + 1,...,ec_K)$.

For every clock tick (with context-based delay),

$dev(px,c,py) = (ec_1,...,ec_K)$ is updated to

$dev^{new}(px,c,py) = (\lambda_1*ec_1,..., \lambda_K*ec_K)$

For every clock unit and for every new experience,

$dtv(px,c,py)$ is updated to

$dtv^{new}(px,c,py) = (d_1,d_2,...,d_K)$

where $d_i = ec_i / (ec_1 + ... + ec_k)$

for each i in $[1,K]$.

(*Subtlety*: In our Python script that computes trust using robust scheme (not shown here), the counts saturate at 1 rather than monotonically diminish to 0 with time, to reflect *ignorance* after long periods of inactivity.)

6.2.5. Evolution of Trust Distribution for Various Experience Sequences

In order to provide better insight into how the direct trust distribution vector evolves, we present final direct trust vectors for different experience sequences in Table 2, and trace evolution of trust distribution vector for a specific experience sequence in Table 3. We then highlight notable characteristics of this approach.

Table 2: Trust Distribution Vector for Different Experience Sequences with K= 4 [1,…,4] and initial value (0.25,0.25,0.25,0.25)

Experience Sequence	Final Trust Distribution (Simple Scheme)	Final Trust Distribution (Robust Scheme)
[1,1,1]	(0.57,0.14,0.14,0.14)	(0.55,0.15,0.15,0.15)
[1,4,1,4]	(0.38,0.12,0.12,0.38)	**(0.42,0.14,0.14,0.29)**
[1, 1, 4, 4, 4, 4, 1, 1]	(0.42,0.08,0.08,0.42)	(0.5,0.1,0.1,0.3)
[1, 1, 4, 4, 4, 4, 1, 1, 1]	(0.53,0.07,0.07,0.33)	(0.64,0.1,0.1,0.17)
[2,3,2,3]	(0.12,0.38,0.38,0.12)	**(0.16,0.4,0.3,0.14)**

[15] Low-level (resp. high-level) experience is synonymous with low-quality (resp. high-level) experience.

Table 3: Evolution of Trust Distribution for Experience Sequence (1,1,1,K,K,K,K,1,1,1)

Exper-ience Sequence Value	Trust Distribution Trace (Simple Scheme)	Trust Distribution Trace (Robust Scheme)
	(0.25,0.25,0.25,0.25)	(0.25,0.25,0.25,0.25)
1	(0.4,0.2,0.2,0.2)	(0.4,0.2,0.2,0.2)
1	(0.5,0.17,0.17,0.17)	(0.53,0.165,0.155,0.15)
1	**(0.57,0.14,0.14,0.14)**	**(0.55,0.15,0.15,0.15)**
K	(0.5,0.125,0.125,0.25)	(0.5,0.13,0.12,0.25)
K	(0.44,0.11,0.11,0.33)	(0.46,0.13,0.13,0.28)
K	**(0.4,0.1,0.1,0.4)**	**(0.42,0.12,0.11,0.35)**
K	(0.36,0.1,0.1,0.45)	(0.37,0.12,0.12,0.38)
1	(0.42,0.08,0.08,0.41)	(0.47,0.11,0.11,0.31)
1	**(0.46,0.08,0.08,0.38)**	**(0.53,0.11,0.11,0.24)**
1	**(0.5,0.07,0.07,0.35)**	**(0.6,0.1,0.1,0.2)**
1	(0.53,0.07,0.07,0.33)	(0.65,0.1,0.1,0.14)
K	(0.5,0.06,0.06,0.37)	(0.6,0.1,0.1,0.2)
1	**(0.53,0.06,0.06,0.35)**	**(0.64,0.1,0.1,0.17)**

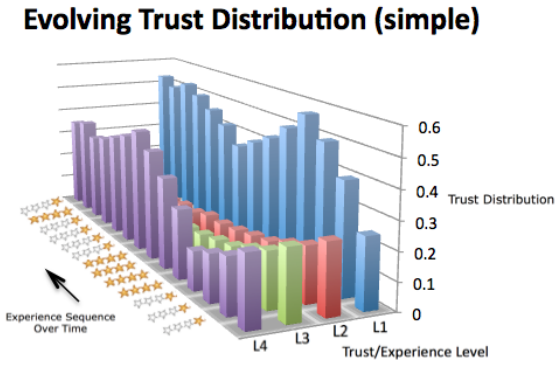

Figure 14: Evolution of Trust Distribution for simple scheme

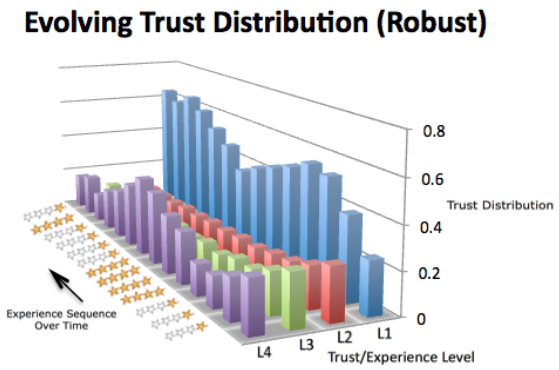

Figure 15: Evolution of Trust Distribution for robust scheme

Figures 14 and 15 depict trust evolution for simple and robust scheme respectively for the experience sequence shown in Table 3.

6.3. Analysis and Security

We analyze the characteristics and the robustness of the Dirichlet distribution-based multi-level trust management approach.

(1) *Symmetry:* The formalization is symmetric with respect to each trust level. For example, for K = 4, the final trust distribution for the experience sequences culminating in (1,4,1,4) and (3,2,3,2) is (0.375,0.375,0.125,0.125) and (0.125,0.375,0.375,0.125), respectively, which captures similar trust distribution pattern. Note that the experience levels are faithfully "preserved" in the updated trust distribution, rather than smeared across trust levels. That is, complementary extreme behavior (credulous interpretation) is treated as different from ignorance (skeptical interpretation).

(2) *Effect of order of experience*: The Simple Scheme is sensitive to the counts of various experience levels but it cannot distinguish their permutation, while the Robust Scheme is sensitive to the order of experiences and is dynamic. Specifically, the recent experience levels have more pronounced effect on the current trust level than prior experience levels. However, to control the rate or extent of memory decay beyond initialization requires context- and application-based tuning.

(3) *Differential aging of trust*: The Robust Scheme ages the trust distribution by decaying counts associated with different levels of trust differently in response to clock ticks. This enables one to have longer memory for "failures" compared to "successes", and more "successes" are needed to offset "failures". To move the trust distribution closer to (1/K,…,1/K) due to long inaction, the experience count saturates to 1 over time.

(4) *Security issues*: We analyze robustness of the proposed approach to various attacks.

a. *Ballot-stuffing attack*: If a majority of the recommenders collude to promote a trustee that provides low-level experience, it can be countered only through more reliable direct experience that gets reflected as low-level direct trust. Unfortunately, the low-level experience will be forgotten over a period of time. This is reasonable if the low-level experience is a result of transient phenomenon or occasional misbehavior, but is not ideal to deal with more persistent fault or malicious behavior.

b. *Bad-mouthing attack*: If a majority of the recommenders collude to avoid a trustee that can provide high-level experience, it can be countered only if a trustor seeks direct experience with the victim trustee in spite of low trust and discovers a contradiction. This situation may be forcibly realized when trusted nodes are unavailable for interaction.

c. *Sybil and Newcomer attacks*: The trust framework does not assist in preventing these attacks. Instead, their mitigation requires a separate authentication infrastructure.

d. *Sleeper and on-off attacks*: The trust framework is well-suited to prevent these attacks as illustrated by the Robust Scheme, although it does require manual control over the memory window and selective weighting of different experience levels as a function of time and application, as shown above.

e. *Conflicting behavior attack*: Recall that, in conflicting behavior attack, the attacker uses "divide and conquer" strategy and provides conflicting recommendations on a trustee to multiple trustworthy sources. When a victim seeks recommendations from these trustworthy sources, which faithfully transmit the attacker's views, the victim ends up getting conflicting recommendations on the trustee, thereby causing it to incorrectly reduce its trust in a subset of trustworthy recommenders. The given trust framework does not track trust in each recommender separately (but instead, it lumps them all together). So ironically, because of this limitation, conflicting behavior attack does not have the intended effect of reducing trust in the intermediaries. The attack does degenerate to bad-mouthing attack however.

(5) *Tracing vs. Experimental Simulation:* We avoid performing any experimental simulation because it does not provide any new insight beyond sanity check. This is because if the simulation framework is set-up in such a way that low-trust nodes provide low throughput, and experiment always selects highest-trust nodes or nodes with a probability that is proportional to their trust value, to communicate, the overall performance is bound to improve. Instead, we have tried to trace and visualize the evolution of multi-level trust on diverse concrete examples, to get a better insight into its behavior.

6.4. Comparative Analysis Tabular Summary

The proposed multi-valued trust inference algorithm and its high-level relationship to several binary trust inference algorithms are summarized in Table 4. In what follows, we recapitulate important characteristics of these approaches which also accounts for their robustness to various attacks as discussed in Section 5.1.3 and 6.3.

In Denko and Sun [24], functional trust is aggregated using information from immediate neighbors and once removed nodes reachable through referral edges. It ignores recommender identity completely. As such, it cannot be as robust w.r.t attacks as the other approaches because it is unable to filter out referrals from just the malicious nodes. In Ganeriwal et al. [23], no distinction is made between functional and referral trust, and trust scope is not explicit. Thus, the computed trust and robustness to attacks are based on coarse-grain, cumulative trust, which is appropriate only in a single trust scope. Sun et al. [24] maintains separate functional and referral trust, and provides an axiomatic basis for their trust model (that is, for trust propagation via chaining and aggregation), which is robust w.r.t. attacks. Unfortunately, the axioms have limited applicability and do not unambiguously specify trust computation over an arbitrary trust network (a la others including Josang and Ismail [39], Thirunarayan et al. [1], Golbeck and Hendler [16], etc). Quercia et al. [40] generalize binary trust to multi-valued trust and separate functional and referral trust for different trust scopes. Unfortunately, the Bayesian formulation does not evolve the primitive trust values in a satisfactory manner. Our approach to multi-valued trust, discussed in Section 6.2, improves upon Quercia et al. [40] by providing a satisfactory probabilistic basis for trust computation and evolution founded on Dirichlet distribution, and with acceptable robustness characteristics as discussed in Section 6.3.

Table 4: Comparative Analysis of various approaches to binary and multi-level trust

APPROACH/ METRIC	Trust Type / Context	Trust Model / Foundation	Robustness to Attacks
D[24] / Binary	Functional / One	Trivial chaining / Beta-PDF	Limited Ballot-stuffing; Bad-mouthing
G[23] / Binary	Functional / Indistinguishable	Josang-Ismail discounting /	Ballot-stuffing; Bad-mouthing;

		Beta-PDF	Sleeper and On-off
S[25] / Binary	Functional + Referral / One	Limited chaining and aggregation / Beta-PDF	Ballot-stuffing; Bad-mouthing; Sleeper and On-off
Q[40] / Multi-level	Functional + Referral / Multiple	No / Bayesian Ad Hoc	Ballot-stuffing; Bad-mouthing; Sleeper and On-off; Sybil
Ours / Multi-level	Functional + Referral / Multiple	No / Dirichlet-PDF	Ballot-stuffing; Bad-mouthing; Sleeper and On-off; Conflicting behavior

6.5. Other Applications of Trust Based on Dirichlet Distribution

The pioneering work of Josang and Haller [41] uses the Dirichlet distribution analyzed above as the basis for multi-level reputation system for e-commerce. Their paper also presents: (i) A counterintuitive consequence of using uniform distribution as a prior on the rate of assimilation of experience sequence if the number of levels is very large (e.g., 100 similar experiences for an 100-level trust metric leads to an expected probability of only ½ for the corresponding trust level rather than a substantially higher value); (ii) A better visualization of the results; (iii) Simple special cases that permit closed form solution for expected trust in the presence of trust decay over time; (iv) Different representations of reputation score; and (v) A potential practical application of multi-level trust to browsers by introducing a toolbar for rating Web pages by clients and for displaying recommendation summaries for subsequent use by other clients, similarly to the star-ratings (and reviews) provided on e-commerce web sites such as Amazon.com. This approach to ranking based on explicit client ratings has been called *critical surfer model* in contrast with random surfer model based on hyperlinks and intentional surfer model based on actual visits [41].

Yang and Cemerlic [44] discusses the application of Dirichlet reputation to sharing resources among unknown peers in a collaborative environment, to minimize risk in usage control. Each requestor is evaluated for its suitability as a collaborator on the basis of directly observed behavior and (possibly discounted) peer recommendations (shared regularly among neighbors). The Dirichlet distribution is used to characterize multiple dimensions of an interaction such as being friendly, selfish, malicious, etc. The paper does not however explicitly specify deviation test or decision thresholds for multi-valued trust metric or choice of window-size for dealing with varying trustworthiness.

Reece et al. [45] proposes a probabilistic approach to computational trust for multi-dimensional contracts with correlated dimensions (e.g., timeliness, quality of service, quantity, etc.) The work demonstrates that taking into account correlation among different dimensions gives superior trust estimates and makes it robust with respect to rumors [16]. Specifically, tracking provenance of recommendation and separating recommendations as private and shared can avoid double counting in decentralized reputation systems. These ideas can also be applied to other frameworks such as Thirunarayan et al. [1].

Fung et al. [46] adapts Dirichlet-based trust management to collaborative host-based intrusion detection networks (HIDN) (i) to detect intrusions such as worms, viruses, denial-of-service attacks, malicious logins, etc., (ii) to detect malicious/compromised nodes, and (iii) to improve security. For this purpose:

(a) It segregates HIDN nodes into two lists: probation[17] list and acquaintance list, to ensure that recommendations are sought only from (mature) nodes with some track record. It length limits these lists using trust value and associated confidence for scalability reasons.

(b) It uses both intrusion consultations (recommendations) and (novel) test messages to assess trustworthiness. The latter messages are "bogus" requests of known type used as gold standard to assess trustworthiness of a response, and effectively, the responder.

(c) It uses Dirichlet-based multi-level trust model with forgetting factor, where the experience level is determined by discretizing satisfaction feedback computed from expected answer, received answer, and for a test message, its difficulty level.

(d) It secures the trust system against well-known attacks. Security against Sybil attack requires additional authentication mechanism, while probation list and forgetting factor improves robustness against newcomer attack and betrayal

[16] In data fusion research, rumor propagation (or data incest) refers to double counting of recommendations from the same source via different paths.

[17] Cf. Nursery in generational garbage collectors

(sleeper) attack respectively. Dynamic test message rate secures against collusion (bad-mouthing) and inconsistency (on-off) attacks. Specifically, test message rate is increased when a node starts behaving dishonestly or has higher trust uncertainty.

6.6 Additional Sample Applications of Trust in Collaborative Environments

Grid and P2P computing systems enable sharing of computing resources. Traditional techniques to secure these systems include sandboxing, encryption, and other access control and authentication mechanisms. As discussed in Azzedin and Maheswaran [47], trust information can be incorporated into these systems to specify consumer preferences and requirements regarding resources and their producers for an application, yielding trust-aware resource management systems. Scheduling algorithms in such systems face additional load balancing challenges to deal with trust constraints. Azzedin and Maheswaran [48] evaluates a trust model for P2P systems that (i) supports multi-level contextual trust, (ii) distinguishes direct/functional and indirect/referral trust, (iii) captures dynamism through temporal decay, and (iv) successfully detects "bad" domains. Azzedin and Ridha [49] investigate "honesty checking schemes" for detecting bogus recommendations and assessing recommenders. This is analogous to detecting bad-mouthing and ballot-stuffing attacks. They also consider recommenders that are inconsistent, that is, change their recommendation strategy. This is analogous to detecting sleeper and on-off attacks.

Bessis et al. [50] and Brown et al. [51] propose a trust-based cooperative grid communities using self-led critical friend model. Functional trust in a node is obtained by taking the average of the product of functional trust in the node from a common neighbor with the latter's referral trust. The trust is decayed by specifying half-life. Critical friends of a node are neighbors that have a trust score higher than a context-determined threshold. These are used to grow critical friends' community for resource sharing and job scheduling.

Trust is crucial for collaboration in pervasive environments [52] where an agent may encounter other agents in a distributed and possibly hostile environment. In Ajayi et al. [53], the access control policy in a distributed environment is a function of inter-organizational trust. Specifically, the Dynamic Trust Negotiation (DTN) model supports dynamic allocation of security policies in collaborative environments. With increased growth of Virtual Organizations (VO) as a result of geographically fragmented, networked and independent organizations, resources such as IT and humans are shared by these organizations [54]. Trust plays an important role in assessing risks and choosing best collaborators. Trust has been a focus of research on virtual collaboration in distributed teams, e-commerce, e-learning, and telemedicine. Interpersonal trust is also critical for cooperation among teams of scientists, technologists, engineers, and managers.

Winkler et al. [27] present taxonomy of trust indicators (analogous to trust scope in Section 4) relevant to reputation of VO. Specifically, they formalize a Bayesian networks approach to reputation for trust indicator aggregation and trust update with temporal decay.

There is contemporary interest in gleaning interpersonal trust from physical, linguistic, and behavioral features available through interactions, and influencing trustworthiness by manipulating/adapting external presentation and perception [15]. For example, van't Wout and Sanfey [55] illustrates the effect of facial social cues on perceived trustworthiness and eventually on strategic decision making, while Wang and Emurian [56] explores characteristics that influence online trust formation, and applies that for the design of trust-inducing features of e-commerce Websites. The study of cross-cultural differences in trustworthiness qualities and trust thresholds to better understand what aspects improve *influence* and what aspects flag *manipulation* is gaining importance is today's well-connected world.

The research challenges and directions outlined above are applicable to distributed collaborative systems because the collaborators that provide content and services are often remote from end-users and partners, and trust inference is essential for basing decisions in the absence of direct knowledge about each other.

Rotter [57] defines interpersonal trust as expectancy held by an individual or a group that the word, promise, verbal or written statement of another individual or group can be relied on. He explores what personal traits, such as religious beliefs, age, need, and gullibility, can be used to predict trustworthiness, and how trust and knowledge of deception-related situations can influence specific behaviors in a given situation. Yakovleva et al. [58] investigates interpersonal trust in various dyadic relationships such as virtual dyads vs co-located dyads, sheds light on the reciprocal influences of trust and empirically shows characteristics that determine trustworthiness (such as ability, integrity, and benevolence). It also shows that initial trust may vary for individuals based on propensity to trust, and in collaborative environments, reciprocal effects influence trust in dyadic relationships.

McKnight et al. [59] discuss multidimensional nature of trust in e-commerce. For instance, they distinguish trust in a vendor to deliver on commitments, from trust in vendor's ethical use of consumer data, to trust in Internet communication being secure. (Our ontology tries to accommodate such distinctions using trust scope.) It also explains and illustrates, in detail, the nature of initial trust in an unfamiliar trustee, factors that influence trust formation such as characteristics of a trustee (such as competence and integrity) and trustor's disposition to trust (such as faith in humanity and benevolence).

Deception is the betrayal of trust, and ironically, trust makes us prone to deception. Knowing what features are used to glean trustworthiness can also assist in avoiding detection while deceiving. Deception is an important issue in the context of e-commerce, both from the buyer's perspective (caveat emptor) and from the seller's perspective (caveat venditor/mercator). According to Castelfranchi and Tan [60], in hybrid situations where artificial agents interact with human agents, it is important that artificial agents can reason about the trustworthiness and deceptive actions of their human counter parts. In fact, agents in virtual communities are and will be designed and trained to deceive, and people will be deceived by and will deceive artificial agents. Lappas [61] regards writing fake reviews as a form of attack on reputation-based system and provides an attacker's perspective on creating authentic-looking and impactful reviews (that can harm or boost an item's reputation as desired). Lappas [61] formalizes and evaluates impact and authenticity of a review (the latter in terms of the three factors -- stealth (which is the ability to blend in), coherence (which refers to the consistency between numeric/star-rating and the textual description) and readability (measured using Flesh-Reading Ease formula)). Anantharam et al. [62] discusses a scalable and adaptive machine learning approach to detect topically anomalous tweets that propagate self-serving content using trending topics.

7. CONCLUSIONS

In this work, we have provided simple examples to motivate practical trust issues, explained salient features that characterize trust and distinguished it from related concepts such as trustworthiness, reputation, security, belief, etc. We have also discussed our trust ontology to situate different approaches in the literature, and showed illustrative examples of gleaning trustworthiness. Finally, we touched upon some research challenges for modeling trust and inferring trustworthiness in the context of interpersonal, sensor and social networks, and collaborative systems.

Due to the practical significance of Bayesian approaches to automatic trust prediction, we have presented a comparative analysis of various approaches to gleaning trustworthiness in machine networks (including ad hoc mobile networks, sensor networks, etc.) and their robustness to well-known attacks. We have focused on different trust metrics and types (functional vs. referral), data structures to represent trust networks and related trust information, Beta-PDF and Dirichlet distribution for direct trust computation, trust models for trust propagation and evolution in response to different behaviors. We expect comparative analysis to spur development of expressive trust networks that make explicit various choices or their resolutions objectively. Ultimately, the holy grail of trust research is to develop expressive trust frameworks that have both declarative/axiomatic and computational specification, and to devise methodologies for instantiating them for practical use, by justifying automatic trust/trustworthiness inference in terms of application-oriented semantics of trust.

ACKNOWLEDGEMENT

We thank the reviewers for their insightful suggestions that have improved the organization and presentation of our work.

REFERENCES

[1] K. Thirunarayan, D. K. Althuru, C. A. Henson, and A. P. Sheth, “A Local Qualitative Approach to Referral and Functional Trust”, The 4th Indian International Conference on Artificial Intelligence (IICAI-09), pp. 574-588, December 2009.

[2] K. Thirunarayan and R. Verma. “A Framework for Trust and Distrust Networks”, Web 2.0 Trust Workshop (W2Trust), June 2008.

[3] S. P. Marsh, “Formalising Trust as a Computational Concept”, Ph.D. Dissertation, University of Stirling, 1994.

[4] T. Grandison, and M. Sloman: “A Survey of Trust in Internet Applications”, IEEE Communications Surveys and Tutorials 3(4), pp. 2-16, 2000.

[5] D. Artz, and Y. Gil: “A Survey of Trust in Computer Science and the Semantic Web”, J. Web Semantics. 5(2), pp. 58-71, 2007.

[6] A. Jøsang, R. Ismail, and C. Boyd. “A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems”, 43(2), pp. 618-644, 2007.

[7] H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato. “A Survey of Trust and Reputation Management Systems in

Wireless Communications", Proceedings of the IEEE 98: 10, pp. 1755-1772, 2010.

[8] Sonja Buchegger, Jean Yves Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation of nodes-fairness in dynamic ad-hoc networks", Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Network and Computing (MobiHOC 2002), Lausanne, Switzerland, pp. 226-236, June 2002.

[9] Hussain, F.K.; Chang, E.; Hussain, O.K. "State of the Art Review of the Existing Bayesian-Network Based Approaches to Trust and Reputation Computation", Second International Conference on Internet Monitoring and Protection (ICIMP 2007), 4 pages, July 2007.

[10] Mohammad Momani and Subhash Challa, "Survey of Trust Models in Different Network Domains", International Journal of Ad hoc, Sensor & Ubiquitous Computing, September 2010, Volume 1, Number 3.

[11] Kannan Govindan and Prasant Mohapatra, "Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey", IEEE Communications Surveys and Tutorials, pp. 279-298, 2012.

[12] J. Golbeck, B. Parsia, and J. A. Hendler: "Trust Networks on the Semantic Web", Cooperative Information Agents VII, pp. 238-249, 2003.

[13] J. Golbeck: "Computing and Applying Trust in Web-based Social Networks", Ph.D. Dissertation, University of Maryland, 2003.

[14] K. Thirunarayan and P. Anantharam, "Trust Networks: Interpersonal, Sensor, and Social," In: Proceedings of 2011 International Conference on Collaborative Technologies and Systems (CTS 2011), Philadelphia, Pennsylvania, USA, pp. 8 pages, May 23-27, 2011.

[15] A. Russell, "TRUST Proposers' Day Briefing IARPA-BAA-10-03 Overview", IARPA.

[16] J. Golbeck, and J. Hendler, "Inferring binary trust relationships in Web-based social networks," ACM Transactions on Internet Technology, pp. 497-529, Vol. 6, No. 4, 2006.

[17] D. Gambetta, "Can We Trust Trust?," In Trust: Making and Breaking Cooperative Relations (1988).

[18] I. Bohnet, B. Herrmann, and R. Zeckhauser, "Trust and the Reference points for Trustworthiness in Gulf and Western Countries," Vol. 125, No. 2, pp. 811-828, May 2010.

[19] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," Computer Networks, Vol. 30, No. (1-7), pp. 107–117, 1998.

[20] P. Anantharam, C. A. Henson, K. Thirunarayan, and A. P. Sheth, "Trust Model for Semantic Sensor and Social Networks: A Preliminary Report," National Aerospace & Electronics Conference (NAECON), Dayton Ohio, July 2010.

[21] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," The 13th international Conference on World Wide Web (WWW '04), pp. 403-412, 2004.

[22] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An Integrative Model of Organizational Trust". Academy of Management Review, 20, pp. 709-734, 1995.

[23] S. Ganeriwal, L. Balzano, and M. B Srivastava, "Reputation-based Framework for High Integrity Sensor Networks", ACM Transactions on Sensor Networks (TOSN), Vol. 4, Issue. 3, pp. 1-37, June 2008.

[24] M. K. Denko, and T, Sun: "Probabilistic Trust Management in Pervasive Computing". *EUC(2)* 2008: 610-615.

[25] Y. Sun, W. Yu, Z. Han, and K.J.R Liu, "A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks", In: Proceedings of IEEE INFOCOM '06, Barcelona, Spain, Apr. 2006.

[26] A. Jøsang. "Fission of Opinions in Subjective Logic", The 12th International Conference on Information Fusion (FUSION 2009), Seattle, July 2009.

[27] Till J. Winkler, Jochen Haller, Henner Gimpel, Christof Weinhardt, "Trust Indicator Modeling for a Reputation Service in Virtual Organizations", ECIS 2007, pp. 1584-1595.

[28] C. Henson, K. Thirunarayan, A. Sheth. "An Ontological Approach to Focusing Attention and Enhancing Machine Perception on the Web". Applied Ontology, vol. 6(4), pp. 345-376, 2011.

[29] "Trusted Perception Cycle" [Demo], Available: http://www.youtube.com/watch?v=lTxzghCjGgU (accessed 10/23/2012)

[30] Sai T. Moturu and Huan Liu. "Quantifying the Trustworthiness of Social Media Content", Distributed and Parallel Databases,Vol. 29, No. 3, pp. 239-260, 2011.

[31] M. d'Aquin, S. Elahi, and E. Motta, "Semantic monitoring of personal web activity to support the management of trust and privacy," SPOT 2010: 2nd Workshop on Trust and Privacy on the Social and Semantic Web, Heraklion, Greece, May 2010.

[32] U. Kuter and J. Golbeck, "Semantic Web Service Composition in Social Environments," 8th International Semantic Web Conference, ISWC 2009, Vol. 5823, pp. 344-358, Chantilly, VA, USA, October 2009.

[33] P. Massa and P. Avesani, "Trust-aware recommender systems", ACM Conference on Recommender Systems

(RecSys, 2007), pp. 17-24, 2007.

[34] Y. L. Sun, W. Yu, Z. Han, and K. J. R. Liu, "Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks", IEEE Journal on Selected Areas in Communications, Vol. 24, Issue 2, pp. 305-316, Feb 2006.

[35] M. Richardson, R. Agrawal and P. Domingos, "Trust Management for the Semantic Web", The Second International Semantic Web Conference, pp. 351–368, 2003.

[36] P. Massa,and P. Avesani, "Controversial users demand local trust metrics: an experimental study on epinions.com community", The 25th American Association for Artificial Intelligence Conference, pp. 121-126, 2005.

[37] U. Kuter and J. Golbeck, "SUNNY: A New Algorithm for Trust Inference in Social Networks Using Probabilistic Confidence Models", The Twenty-Second AAAI Conference on Artificial Intelligence, pp. 1377-1382, Vancouver, British Columbia, Canada, July 2007.

[38] V. G. Bintzios and T. G. Papaioannou and G. D. Stamoulis, "An Effective Approach for Accurate Estimation of Trust of Distant Information Sources in the Semantic Web", Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2006), pp. 69-74, Lyon, France, June 2006.

[39] A. Jøsang and R. Ismail, "The Beta Reputation System", The 15th Bled Electronic Commerce Conference, Bled, Slovenia, June 2002.

[40] D. Quercia, S. Hailes, and L. Capra, "B-Trust: Bayesian Trust Framework for Pervasive Computing", In: Lecture Notes in Computer Science, 2006, vol. 3986, pp. 298-312.

[41] A. Jøsang, and J. Haller, "Dirichlet Reputation Systems", The 2nd International Conference on Availability, Reliability and Security (ARES 2007), pp. 112-119, 2007

[42] B. A. Frigyik, A. Kapila, and M. R. Gupta, "Introduction to the Dirichlet Distribution and Related Processes", UWEE Tech Report UWEETR-2010-0006.

[43] T. J. O'Donnell, and N. D. Goodman, and Andreas Stuhlmueller, and the Church Working Group, "Models with Unbounded Complexity", Probabilistic Models of Cognition Tutorial.

[44] L. Yang and A. Cemerlic, "Integrating Dirichlet reputation into usage control", In Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies (CSIIRW '09), Frederick Sheldon, Greg Peterson, Axel Krings, Robert Abercrombie, and Ali Mili (Eds.). ACM, New York, NY, USA, Article 62 , 4 pages, 2009.

[45] S. Reece, A. Rogers, S. Roberts, and Nicholas R. Jennings, "Rumours and Reputation: Evaluating Multi-dimensional Trust within a Decentralised Reputation System", Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems (AAMAS '07), Article 165, 8 pages, 2007.

[46] C. J. Fung, J. Zhang, I. Aib, and R. Boutaba, "Dirichlet-Based Trust Management for Effective Collaborative Intrusion Detection Networks", Computer Engineering, 8(2), 79-91, 2011.

[47] F. Azzedin, and M. Maheswaran: "Integrating Trust into Grid Resource Management Systems", ICPP 2002, pp. 47-54, 2002.

[48] F. Azzedin and M. Maheswaran: "Trust Modeling for Peer-to-Peer Based Computing Systems", IPDPS 2003, 10 pages, 2003.

[49] F. Azzedin and A. Ridha: "Feedback Behavior and its Role in Trust Assessment for Peer-to-Peer Systems", Telecommunication Systems 44(3-4), pp. 253-266, 2010.

[50] N. Bessis, Y. Huang, P. Norrington, A. Brown, P. Kuonen, and B. Hirsbrunner, "Modelling of a Self-led Critical Friend Topology in Inter-cooperative Grid Communities", International Journal of Simulation Modelling Practice and Theory, Elsevier, Volume 19, Issue 1, ISSN: 1569-190X, pp. 5-16, 2011.

[51] A. Brown, P. Sant, N. Bessis, T. French and C. Maple, Modelling "Self-led Trust Value Management in Grid and Service Oriented Infrastructures: A Graph Theoretic Social Network Mediated Approach", International Journal of Systems and Service Oriented Engineering, IGI, Volume 1, Issue 4, ISSN: 1947-3052, pp. 1-19, 2010.

[52] V. Cahill, B. Sh, E. Gray, N. Dimmock, A. Twigg, J. Bacon, C. English, W. Wagealla, S. Terzis, P. Nixon, C. Bryce, G. D. M. Serugendo, J.-marc Seigneur, M. Carbone, K. Krukow, C. Jensen, Y. Chen, and M. Nielsen, "Using Trust for Secure Collaboration in Uncertain Environments", IEEE Pervasive Computing, Volume 2, pp. 52-61, 2003.

[53] O. Ajayi, R. O. Sinnott, and A. Stell, "Trust Realisation in Multi-domain Collaborative Environments", The 6th IEEEACIS International Conference on Computer and Information Science ICIS, pp. 906-911, July 2007.

[54] K. Michel, L. Romain, B. Francois, and B. Abdelmalek, "A Trust-based Virtual Collaborative Environment", Journal of Digital Information Management, Source Volume: 6 Source Issue: 5, Oct. 2008.

[55] M. van 't Wout, and A. G. Sanfey, Friend or Foe: "The Effect of Implicit Trustworthiness Judgments in Social Decision-Making". Cognition, 108, pp. 796-803, 2008.

[56] Y. D. Wang, and H. H. Emurian, "An Overview of

Online Trust: Concepts, Elements, and Implications". Computers in Human Behavior, pp. 105-125, 2005.

[57] J. B. Rotter, "Generalized Expectancies for Interpersonal Trust", American Psychologist, 26, pp. 443-452, 1971.

[58] M. Yakovleva, R. Reilly, and R. Werko. "Why do we Trust? Moving beyond Individual to Dyadic Perceptions", Journal of Applied Psychology, 95, pp. 75-91, 2010.

[59] D. McKnight, V. Choudhury, and C. Kacmar, "Developing and Validating Trust Measures for E-commerce: An Integrative Typology", Information Systems Research, 13, pp. 334-359, 2002.

[60] C. Castelfranchi and Y. H. Tan: "The Role of Trust and Deception in Virtual Societies, International Journal of Electronic Commerce", Vol. 6, No. 3, pp. 55-70, 2002.

[61] T. Lappas, "Fake Reviews: The Malicious Perspective", In: Proceedings of 17th International Conference on Applications of Natural Language to Information Systems, pp. 23-34, June 2012.

[62] P. Anantharam, K. Thirunarayan, and A. Sheth, "Topical Anomaly Detection for Twitter Stream", In the Proceedings of ACM Web Science 2012, pp. 11-14, June 2012.