Chapter 2

Trust Models

Multi-agent systems consist of a large number of intelligent, interactive and (partially) autonomous agents that must cooperate to complete a certain task, often too difficult to solve for an individual agent. Such systems are used in a wide range of applications, ranging from mobile environments [73], over the creation of crowd-related effects for movies\(^1\), to online trading [57]. Multi-agent systems can often benefit from a trust system, especially when the circumstances do not allow for perfect information about the interaction partners’ behavior and intentions [117]. They may for example incorporate a trust network to monitor and control the behavior of the agents that participate in a process, think e.g. of an online market place such as eBay. Another nice illustration can be found in [66], in which a trust network is used to alleviate the problem of corrupt sources in peer-to-peer file-sharing networks by keeping track of the peers’ trustworthiness. With the advent of the Semantic Web [12], even more applications and systems will need solid trust mechanisms. The Semantic Web is an extension of the current web where content is annotated (see RDF\(^2\) and OWL\(^3\)) such that machines and computers are able to understand its meaning and reason with it. Hence, since more and more intelligent agents will take over human tasks in the future, they also require an automated way of inferring trust in each other, see for instance [123].

Nowadays, effective models already play an important role in many Web 2.0 applications. Question answering systems can compute trust indications along with the answers based on how much trust the user puts into certain sources [153], recommender systems can produce suggestions more tailored to the users’ tastes (chap-

---

\(^1\)Massive Software, see www.massivesoftware.com

\(^2\)Resource Description Framework, see www.w3.org/TR/rdf-primer

\(^3\)Web Ontology Language, see www.w3.org/TR/owl-features
ter 6), consumer review sites can show personalized orderings of reviews based on which people the user trusts (think of Epinions), etc.

In the first part of this book (chapters 2-4) we will focus on the most general use of trust models; trust in social networks will be discussed in detail in the second part. Therefore, in the remainder of this chapter, we will use the term ‘agent’ to refer to the people/machines that participate in a certain process, and only use the term ‘user’ in the specific context of social network applications.

A lot of agent and social applications (will) use, in one way or another, a web of trust that allows agents to express trust in other agents. Trust recommendations derived from these networks are supposed to help them to develop their own opinions about how much they may trust other agents and sources. However, despite recent developments in the area, most of the trust models and metrics proposed so far tend to lose potentially important trust-related knowledge.

In the following sections, we give an overview of existing trust models (Sec. 2.1) and explain their shortcomings with regard to preserving trust provenance information (Sec. 2.2). These are serious drawbacks in large networks where many users are unknown to each other and might provide contradictory information. Therefore, to meet the needs for a framework that can help agents to make better informed (trust) decisions, we propose a new trust model in which trust values are derived from a bilattice that preserves valuable trust provenance information including partial trust, partial distrust, ignorance and inconsistency (Sec. 2.3 and 2.4).

2.1 Classification of Trust Models

Trust and trust models have been used in many fields of computer science, and also in a wide range of applications; a nice overview can be found in [7] in which Artz and Gil classify trust research in four major areas: models that use policies to establish trust (enforcing access policies, managing credentials, etc.), general trust models such as [26] and [158], models for trust in information sources such as [153], and reputation-based trust models. The first category deals with ‘hard’ trust, which involves identity verification and authorization. However, it is not because an agent is who he claims to be, that everyone automatically should trust his actions, statements or intentions; some agents might trust a particular agent while others do not. In this book, we will not handle the ‘security side’ of trust, but focus on ‘soft’, interpersonal trust; trust that can be computed among two individuals in
a network. In particular, we will mainly discuss trust models and metrics that belong to Artz and Gil’s last category. This category includes, among others, research that uses the history of an agent’s actions or behavior (see e.g. [66, 104]), and work that computes trust over social networks, such as [50, 86]. The trust-enhanced recommender techniques that we will describe in Chap. 6 all belong to this class.

Trust models come in many flavors and can be classified in several ways. We focus on two such classifications, namely probabilistic versus gradual approaches, and representations of trust versus representations of both trust and distrust. Table 2.1 shows some representative references for each class.

A **probabilistic** approach deals with a single trust value in a black or white fashion — an agent or source can either be trusted or not — and computes a probability that the agent can be trusted. In such a setting, a higher suggested trust value corresponds to a higher probability that an agent can be trusted. Examples can, among others, be found in [153] in which Zaihrayeu et al. present an extension of an inference infrastructure that takes into account the trust between users and between users and provenance elements in the system, in [123] where the focus is on computing trust for applications containing semantic information such as a bibliography server, or in contributions like [78] in which a trust system is designed to make community blogs more attack-resistant.

Trust is also often based on the number of positive and negative transactions between agents in a virtual network, such as in Kamvar et al.’s Eigentrust for peer-to-peer (P2P) networks [66], or Noh’s formal model based on feedbacks in a social network [102]. Both [62] and [113] use a subjective logic framework (discussed later on in this section) to represent trust values; the former for quantifying and reasoning about trust in IT equipment, and the latter for determining the trustworthiness of agents in a peer-to-peer system.

On the other hand, a **gradual** approach is concerned with the estimation of trust values when the outcome of an action can be positive to some extent, e.g. when provided information can be right or wrong to some degree, as opposed to being either right or wrong (e.g. [1, 23, 38, 50, 82, 88, 134, 158]). In a gradual setting, trust values are not interpreted as probabilities: a higher trust value corresponds to a higher trust in an agent, which makes the ordering of trust values a very important factor in such scenarios. Note that in real life, too, trust is often interpreted as a gradual phenomenon: humans do not merely reason in terms of ‘trusting’ and ‘not trusting’, but rather trusting someone ‘very much’ or ‘more or less’. Fuzzy logic [69, 152] is very well-suited to represent such natural language labels which represent vague intervals rather than exact values. For instance, in [134] and [76], fuzzy
linguistic terms are used to specify the trust in agents in a P2P network, and in a social network, respectively. A classical example of trust as a gradual notion can be found in [1], in which a four-value scale is used to determine the trustworthiness of agents, viz. very trustworthy – trustworthy – untrustworthy – very untrustworthy.

The last years have witnessed a rapid increase of gradual trust approaches, ranging from socio-cognitive models (for example implemented by fuzzy cognitive maps in [26]), over management mechanisms for selecting good interaction partners on the web [134] or for open and dynamic environments (e.g. [119] or Almenárez et al.’s PTM [4]), to representations for use in mobile environments [82] or recommender systems [38, 86], and general models tailored to Semantic Web applications [156].

While trust is increasingly getting established, the use and modeling of distrust remains relatively unexplored. Although recent publications [27, 40, 129] show an emerging interest in modeling the notion of distrust, models that take into account both trust and distrust are still scarce. Most approaches completely ignore distrust (see for example [76, 78, 80, 99, 123, 153]), or consider trust and distrust as opposite ends of the same continuous scale (see e.g. [1, 42, 134]). However, in agent network theory there is a growing body of opinion that distrust cannot be seen as the equivalent of lack of trust [19, 32, 85]. Moreover, work in the psychology area has repeatedly asked for a re-examination of the assumption that positive- and negative-valent feelings are not separable [16, 112, 116], and some researchers even claim that trust and distrust are not opposite, but related dimensions that can occur simultaneously [20, 79, 94].

To the best of our knowledge, there is only one probabilistic model that considers trust and distrust simultaneously: in Jøsang’s subjective logic [60, 62], an opinion includes a belief $b$ that an agent is to be trusted, a disbelief $d$ corresponding to a belief that an agent is not
to be trusted, and an uncertainty $u$. The uncertainty factor leaves room for ignorance, but the requirement that the belief $b$, the disbelief $d$ and the uncertainty $u$ sum up to 1, rules out options for inconsistency even though this might arise quite naturally in large networks with contradictory sources.

Examples of gradual models that represent trust and distrust as two separate values can be found in [23, 50, 115]. De Cock and Pinheiro Da Silva [23] propose to model the trust network as an intuitionistic fuzzy relation [9], but the same remark w.r.t. inconsistency applies to their model too: the sum of the membership degree (trust $t$, in $[0,1]$) and the non-membership degree (distrust $d$, in $[0,1]$) must be less or equal than 1. The pair can then be represented as an interval $[t, 1-d]$. This approach is somewhat similar to Prade’s work [115] where trust evaluations are represented as an interval in a bipolar trust scale. However, the interval is seen as an imprecise evaluation of the degree of trust [115], rather than an evaluation of trust and distrust independently (like in [23] and [60]). The latter is also the approach taken by Guha et al., who use a couple $(t,d)$ with a trust degree $t$ and a distrust degree $d$, both in $[0,1]$. To obtain the final suggested trust value, they subtract $d$ from $t$ [50]. As we explain later on, potentially important information is lost when the trust and distrust scales are merged into one.

In the next section, we point out the importance of a provenance-preserving trust model. It will become clear that current models (only taking into account trust or both trust and distrust) are either not capable of properly handling inconsistency, or cannot differentiate unknown agents from malicious agents, although these problems can possibly have a large effect on the (ranking of) trust estimation, recommendations, etc.

### 2.2 Trust Provenance

The main aim in using trust networks is to allow agents to form trust opinions on unknown agents or sources by asking for a trust opinion from acquainted agents. Existing trust network models usually apply suitable trust propagation and aggregation operators to compute the resulting trust estimation. But in passing on this trust value to the inquiring agent, often valuable information on how this value is obtained is lost.

Agent opinions, however, may be affected by provenance information exposing how trust values have been computed. For example, a trust opinion in a source from a fully informed agent is quite different from a trust estimation from an agent who does not know the sources too well but has no evidence to distrust it. Unfortunately, in current models, agents
cannot really exercise their right to interpret how trust is computed since most models do not preserve trust provenance.

Trust networks are typically challenged by two important problems influencing trust opinions. Firstly, in large networks it is likely that many agents do not know each other, hence there is an abundance of ignorance. Secondly, because of the lack of a central authority, different agents might provide different and even contradictory information, hence inconsistency may occur. Below we explain how ignorance and inconsistency may affect trust estimations. The first two examples illustrate the need for a provenance-preserving trust model in agent networks and on the Semantic Web, while the last example focuses on its application in the recommender system area.

**Example 2.1 (Ignorance without provenance).** Agent $a$ needs to establish an opinion about both agents $c$ and $d$ to find an efficient web service. To this end, agent $a$ calls upon agent $b$ for trust opinions on agents $c$ and $d$. Agent $b$ completely distrusts agent $c$, hence agent $b$ trusts agent $c$ to degree 0 in the range $[0, 1]$, where 0 is full absence of trust and 1 full presence of trust. On the other hand, agent $b$ does not know agent $d$, hence $b$ trusts $d$ to the degree 0. As a result, agent $b$ returns the same trust opinion to $a$ for both agents $c$ and $d$, namely 0, but the meaning of this value is clearly different in both cases.

With agent $c$, the lack of trust is caused by a presence of distrust, while with agent $d$, the absence of trust is caused by a lack of knowledge. This provenance information is vital for agent $a$ to make a well informed decision. For example, if agent $a$ has a high trust in $b$, $a$ will not consider agent $c$ anymore, but might ask for other opinions on agent $d$. Models working with only one value cannot cope with this kind of situations. A trust model that takes into account both trust and distrust (i.e., two values) could be a possible solution. However, as the examples below illustrate, the existing approaches fall short in other scenarios.

**Example 2.2 (Ignorance without provenance).** Agent $a$ needs to establish an opinion about agent $c$ in order to complete an important bank transaction. Agent $a$ may ask agent $b$ for an opinion of $c$ because agent $a$ does not know anything about $c$. In this case, $b$ is an agent that knows how to compute a trust value of $c$ from its web of trust. Assume that $b$ has evidence for both trusting and distrusting $c$. For instance, let us say that $b$ trusts $c$ to degree 0.5 in the range $[0, 1]$ where 0 is absence of trust and 1 is full presence of trust; and that $b$ distrusts $c$ to the degree 0.2 in the range $[0, 1]$ where 0 is full absence of distrust...
and 1 is full presence of distrust. Another way of saying this is that \( b \) trusts \( c \) at least to the extent 0.5, but also not more than 0.8. The length of the interval \([0.5,0.8]\) indicates how much \( b \) lacks information about \( c \).

In this scenario, by getting the trust value 0.5 from \( b \), agent \( a \) is losing information indicating that \( b \) has some evidence to distrust \( c \) too. Models working with only one value cannot correctly represent this kind of situations. The problem can be solved by the models that take into account two values, and in particular Guha et al.’s [50]. However, their approach has one main disadvantage: agent \( b \) will pass on a value of \( 0.5 - 0.2 = 0.3 \) to \( a \). Again, agent \( a \) is losing valuable trust provenance information indicating, for example, how much \( b \) lacks information about agent \( c \).

**Example 2.3 (Contradictory information).** A stranger tells you that a particular movie was very bad. Because you do not know anything about this person, you make inquiries with two of your friends who are acquainted with him. One of them tells you to trust him, while the other tells you to distrust that same person. In this case, there are two equally trusted friends that tell you the exact opposite thing. In other words, you have to deal with inconsistent information.

This example illustrates how inconsistencies may arise: when an agent in the trust network inquires for a trust estimation about another agent, it often happens that he does not ask one agent’s opinion, but several. Then these information pieces, coming from different sources and propagated through different propagation chains, must be combined together into one new trust value representing the opinion of all the agents, which is not always an easy task when conflicting evidence has been gathered. Nevertheless, this information must be represented unambiguously as it may indicate that it is not possible to take decisions based on the obtained trust value.

Note that models that work only with trust and not with distrust are again not expressive enough to represent these cases adequately. Taking e.g. 0.5 (the average) as an aggregated trust value is not a good solution for Ex. 2.3, because then we cannot differentiate this case from the partial trust situation in which both of your friends trust the movie recommender to the extent 0.5, which indicates that the recommender is somewhat reliable. Furthermore, what would you answer if someone asks you if the stranger can be trusted? A plausible answer is: “I don’t really know, because I have contradictory information about him”. Note that this is fundamentally different from “I don’t know, because I have no information about him”. Hence, an aggregated trust value of 0 is not a suitable option either, as it could
Trust Networks for Recommender Systems

imply both inconsistency and ignorance.

Previous models considering both trust and distrust degrees do not offer the option of representing (partial) inconsistency, even though this might arise quite naturally in large networks with contradictory sources. Jøsang’s subjective logic [60] for example cannot cope with this scenario because the belief and disbelief have to sum up to 1. A similar remark applies to De Cock and Pinheiro da Silva’s intuitionistic fuzzy model [23]. Guha et al. [50] do not impose a restriction on the trust and distrust degrees but their approach suffers from another kind of shortcoming, as Ex. 2.2 illustrated.

2.3 Trust Score Space

The examples in the previous section indicate the need for a model that preserves information on whether a ‘trust problem’ is caused by presence of distrust or rather by lack of knowledge, as well as whether a ‘knowledge problem’ is caused by having too little or too much, i.e. contradictory, information. In other words, we need a model that, on one hand, is able to represent the trust an agent may have in another agent, and on the other hand, can evaluate the contribution of each aspect of trust to the overall trust opinion. As a result, such a model will be able to distinguish between different cases of trust provenance.

To this end, we propose a new framework in which trust values are derived from a bilattice [35]. Since their introduction by Ginsberg in 1988, much attention has been paid to bilattices and their applications. It has for instance been shown that bilattices are useful for providing semantics to logic programs (see e.g. [28]), and as underlying algebraic structures of formalisms for reasoning with imprecise information (see e.g. [21, 33]). The use of these bilattices results in a gradual model for (trust,distrust)-couples. We call such couples trust scores.

**Definition 2.1 (Trust Score).** A trust score \((t,d)\) is an element of \([0,1]^2\), in which \(t\) is called the trust degree, and \(d\) the distrust degree.

Trust scores will be used to compare the degree of trust and distrust an agent may have in other agents in the network, or to compare the uncertainty that is contained in the trust scores. This information can e.g. be used in the ranking mechanisms of a recommender system, a file-sharing system, and so on; for example by giving preference to recommendations/files from sources that are trusted more, or to opinions that are better informed. To this aim, we introduce the trust score space as a model that allows to compare and preserve
information about the provenance of trust scores.

**Definition 2.2 (Trust score space, Trust-Distrust and Knowledge ordering).** The trust score space

\[ \mathcal{B} \mathcal{L} = ([0, 1]^2, \leq_{td}, \leq_k, \neg) \]

consists of the set \([0, 1]^2\) of trust scores, a trust-distrust ordering \(\leq_{td}\), a knowledge ordering \(\leq_k\), and a negation \(\neg\) defined by

\[
(t_1, d_1) \leq_{td} (t_2, d_2) \text{ iff } t_1 \leq t_2 \text{ and } d_1 \geq d_2
\]

\[
(t_1, d_1) \leq_k (t_2, d_2) \text{ iff } t_1 \leq t_2 \text{ and } d_1 \leq d_2
\]

\[
\neg(t_1, d_1) = (d_1, t_1)
\]

for all \((t_1, d_1)\) and \((t_2, d_2)\) in \([0, 1]^2\), while \(d_1\) is the distrust degree.

One can verify that the structure \(\mathcal{B} \mathcal{L}\) is a bilattice in the sense of Ginsberg [35], that is \(([0, 1]^2, \leq_{td})\) and \(([0, 1]^2, \leq_k)\) are both lattices and the negation \(\neg\) serves to impose a relationship between them:

\[
(t_1, d_1) \leq_{td} (t_2, d_2) \Rightarrow \neg(t_1, d_1) \geq_{td} \neg(t_2, d_2)
\]

\[
(t_1, d_1) \leq_k (t_2, d_2) \Rightarrow \neg(t_1, d_1) \leq_k \neg(t_2, d_2),
\]

such that \(\neg\neg(t_1, d_1) = (t_1, d_1)\). In other words, \(\neg\) is an involution that reverses the \(\leq_{td}\)-order and preserves the \(\leq_k\)-order.

Note that Ginsberg’s bilattice is a generalization of *FOUR*, the logic introduced by Belnap in [10, 11], in which he advocated the use of four truth values (‘true’, ‘false’, ‘unknown’ and ‘contradiction’).

Figure 2.1 shows \(\mathcal{B} \mathcal{L}\), along with some examples of trust scores. These scores are interpreted as epistemic values: compared to Jøsang’s subjective logic, the trust and distrust degrees are not complementary, but they reflect the imperfect knowledge we have about the actual trust and distrust values (which are complementary). The lattice \(([0, 1]^2, \leq_{td})\) orders the trust scores going from complete distrust \((0, 1)\) to complete trust \((1, 0)\). The lattice \(([0, 1]^2, \leq_k)\) evaluates the amount of available trust evidence, ranging from a “shortage of evidence”, \(t_1 + d_1 < 1\), to an “excess of evidence”, viz. \(t_1 + d_1 > 1\).

The boundary values of the \(\leq_k\) ordering, \((0, 0)\) and \((1, 1)\), reflect ignorance, resp. contradiction. We call trust scores \((t, d)\) with \(t + d < 1\) incomplete, while those with \(t + d > 1\) are called inconsistent. In both cases, there is a knowledge defect, which can be quantified by the following \([0, 1]\)-valued measure:
Definition 2.3 (Knowledge defect, Knowledge defective trust score). We define the knowledge defect of a trust score \((t, d)\) as \(kd(t, d) = |1 - t - d|\). We say that trust scores \((t, d)\) for which \(kd(t, d) = 0\), i.e., \(t + d = 1\), have perfect knowledge (i.e., there is no uncertainty about the trust value), while all others are called knowledge defective.

Definition 2.4 (Consistent, Inconsistent trust score). We call a trust score \((t, d)\) consistent iff \(t + d \leq 1\), and inconsistent otherwise.

The bottom part of the bilattice (or lower triangle, under the \(kd(t, d) = 0\) line) contains the trust scores for which there is some doubt (uncertainty) about the trust degree. The information contained in such a trust score \((t, d)\) can be represented as an interval \([t, 1 - d]\), denoting that the agent should be trusted at least to the degree \(t\), but not more than \(1 - d\); note the similarities with De Cock et al.’s [23] and Prade’s [115] approaches. In such an interval representation, complete ignorance is represented as \([0, 1]\).

We call agents that issue consistent trust scores consistent agents. In this work, we assume that every agent is consistent. However, although we start from consistent agents, modelling inconsistent information is still needed when we want to accurately represent the result of a trust score aggregation process. This is illustrated by Ex. 2.3; we will elaborate upon this in Chap. 4. The upper part of the bilattice (higher triangle, above the \(kd(t, d) = 0\) line) contains such inconsistent trust scores denoting conflicting information, i.e., trust scores with \(t + d > 1\). Note that trust scores in the higher triangle cannot be represented as
intervals, since they contain too much information instead of a lack.

The trust scores in $\mathcal{BL} = ([0, 1]^2, \leq_{td}, \leq_k, \neg)$ can also be considered within the alternative space $([0, 1]^2, \leq_t, \leq_d, \neg)$, with $\neg$ defined in Def. 2.2, and $\leq_t$ and $\leq_d$ as in Def. 2.5. Note that $\leq_t$ and $\leq_d$ are quasi-orderings, since they are not antisymmetric.

**Definition 2.5 (Trust ordering, Distrust ordering).** The trust ordering $\leq_t$ and distrust ordering $\leq_d$ are defined by

\[(t_1, d_1) \leq_t (t_2, d_2) \text{ iff } t_1 \leq t_2\]
\[(t_1, d_1) \leq_d (t_2, d_2) \text{ iff } d_1 \leq d_2\]

The trust and distrust orderings can also be seen as two extra orderings on $\mathcal{BL}$, which separately evaluate the amount of trust and distrust information respectively. The negation $\neg$ serves to impose a relationship between them:

\[(t_1, d_1) \leq_t (t_2, d_2) \iff \neg(t_1, d_1) \leq_d \neg(t_2, d_2).\]

**Proposition 2.1.** The orderings from Definitions 2.2 and 2.5 are related to each other by $(t_1, d_1) \leq_k (t_2, d_2) \iff (t_1, d_1) \leq_t (t_2, d_2) \land (t_1, d_1) \leq_d (t_2, d_2)$, and $(t_1, d_1) \leq_{td} (t_2, d_2) \iff (t_1, d_1) \leq_t (t_2, d_2) \land (t_1, d_1) \geq_d (t_2, d_2)$.

**Proof.** By definition of $\leq_k$ it holds that $(t_1, d_1) \leq_k (t_2, d_2) \iff t_1 \leq t_2 \land d_1 \leq d_2$, and hence by definition of $\leq_t$ and $\leq_d$ that $t_1 \leq t_2 \land d_1 \leq d_2 \iff (t_1, d_1) \leq_t (t_2, d_2) \land (t_1, d_1) \leq_d (t_2, d_2)$. Analogously for $\leq_{td}$. □

The mapping is illustrated in Fig. 2.2. The dotted line denotes the trust scores $(t, d)$ with perfect knowledge, i.e., $kd(t, d) = 0$. The triangles underneath (in the gray area) contain the consistent trust scores; inconsistent trust scores reside in the upper triangles.

The trust score space allows for a widely applicable lightweight trust model that is nevertheless able to preserve a lot of provenance information by simultaneously representing partial trust, partial distrust, partial ignorance and partial inconsistency, and treating them as different, related concepts. Moreover, by using a bilattice model the aforementioned problems disappear:

1. By using trust scores we can now distinguish full distrust $(0, 1)$ from ignorance $(0, 0)$ and analogously, full trust $(1, 0)$ from inconsistency $(1, 1)$. This is an improvement of, among others, [1, 153].
(2) We can deal with both incomplete information and inconsistency (improvement of [23, 60]).

(3) We do not lose important information (improvement of [50]), because, as will become clear in the next chapters, we keep the trust and distrust degree separated throughout the whole trust process (propagation and aggregation).

### 2.4 Trust Networks

A trust network is a network in which every couple of agents is connected through a trust statement. This statement can denote full trust, partial trust, complete distrust, ..., or ignorance (when two agents do not know each other at all). In other words, the trust relationship between every agent couple can be represented by a trust score. Remark that trust statements are not necessarily reciprocal; think e.g. of a trust network between teachers and students: students may highly trust their math professor for statistical problems, but this certainly does not imply that the teacher will trust every single pupil to the same degree.

It is easy to see that a trust network can be modeled as a directed, fully connected graph. The agents in the trust network can then be represented by nodes in the graph, the relations between the agents by directed edges, and the corresponding levels of trust (trust scores) as weights on the edges.

As has become clear throughout this chapter, we do not work in a binary setting where
agents are either trusted or distrusted, but in an environment where agents can express partial and gradual trust, distrust and ignorance. This brings us to the domain of fuzzy set theory \[69, 152\], an extension of the classical set theory. In the latter, an element either completely belongs to a set, or not at all. Fuzzy sets, however, allow elements to partially belong to a set, and consequently also to belong to several sets at the same time. As a simple example, consider the concept of age: a baby certainly fully belongs to the set of ‘young people’, whereas everyone will agree that an elderly man does not. We say that the baby belongs to the young people set with a membership degree of 1 (on a scale from 0 to 1), while the elderly man has an associated membership degree of 0. On the other hand, what can we say about a teenager? Obviously, a teenager is still a young person, but not as young as a baby; hence, the membership degree will be somewhere in between 0 and 1. Just like ‘young’ and ‘old’, ‘trust’ and ‘distrust’ are also clearly gradual phenomena; hence, fuzzy sets are the pre- eminent tools for modeling trust networks.

Our formal definition of a trust network relies on the notion of a fuzzy relation. A fuzzy relation is characterized by the same two items as a fuzzy set, i.e., its elements and associated membership degrees. This time however, the elements of a fuzzy relation are couples. In our setting, the trust relation between agents in the trust network can be defined by the set of agent couples \((a, b)\) where \(a\) trusts \(b\) to a certain degree (which we call the ‘trust set’). E.g., if \(a\) completely trusts \(b\), then the membership degree of \((a, b)\) in the trust set is 1, and if \(c\) and \(d\) do not know each other, the membership degree of \((c, d)\) in the trust set is 0. Analogously, \((c, d)\)’s membership degree in the set of agents who distrust each other (the distrust set) is 0, and for \((a, b)\) as well. In other words, the trust relation is a fuzzy mapping from the set of couples of agents into \([0, 1]\), and the same definition holds for the distrust relation as well.

However, as we have argued in the previous sections, it is not wise to consider trust and distrust as separate concepts. Hence, it is better to replace the two fuzzy relations (denoting trust and distrust) by one bilattice-fuzzy relation, a mapping from the set of agent into \([0, 1]^2\). This brings us to our final definition of a trust network:

**Definition 2.6 (Trust Network).** A trust network is a couple \((A, R)\) in which \(A\) is the set of agents and \(R\) is an \(A \times A \to [0, 1]^2\) mapping that associates with each couple \((a, b)\) of agents in \(A\) a trust score \(R(a, b) = (R^+(a, b), R^-(a, b))\) in \([0, 1]^2\), in which \(R^+(a, b)\) and \(R^-(a, b)\) denote resp. the trust and distrust degree of \(a\) in \(b\).

In other words, the available trust information is modeled as a \(BL\)-fuzzy relation in the
set of agents that associates a score drawn from the trust score space with each ordered pair of agents. It should be thought of as a snapshot taken at a certain moment, since trust scores can be updated.

2.5 Conclusions

Seldom, very seldom, we have just enough information to make a perfect assessment of someone’s character, tastes or intentions. Instead, we often have too little information or too much information for a good estimation. This is certainly the case in large agent networks where (partial) ignorance and conflicting opinions are the rule rather than the exception. The trust models that had been proposed so far could not cope with such knowledge defects, since they do not preserve vital trust provenance information. Representing trust estimations as elements of a bilattice as we have proposed in this chapter resolves these issues and enables agents to accurately describe their own or computed (via propagation and aggregation) opinions, so that the requiring agents can safely act upon them. The ability to handle ignorance and inconsistency becomes extremely meaningful in an agent network where the trustworthiness of many agents is initially unknown to an agent, which does not imply that he distrusts all of them, but that he may eventually gather evidence to trust or distrust some agents and still ignore others.