

From Simulations to Theorems: A Position Paper on Research in the Field of Computational Trust (Extended Abstract)*

Karl Krukow and Mogens Nielsen

BRICS[†]
University of Aarhus,
Denmark,
(krukow, mn)@brics.dk

Abstract. Since the millennium, a quickly increasing number of research papers in the field of “computational trust and reputation” have appeared in the Computer Science literature. However, it remains hard to compare and evaluate the respective merits of proposed systems. We argue that rigorous use of formal probabilistic models enables the clear specification of the assumptions and objectives of systems, which is necessary for comparisons. To exemplify such probabilistic modeling, we present a simple probabilistic trust model in which the system assumptions as well as its objectives are clearly specified. We show how to compute (in this model) the so-called predictive probability: The probability that the next interaction with a specific principal will have a specific outcome. We sketch preliminary ideas and first theorems indicating how the use of probabilistic models could enable us to quantitatively compare proposed systems in various different environments.

1 Introduction

What are the fundamental models in the field of computational trust?

While this question is highly relevant for researches in the field of computational trust and reputation, in fact, it is hard to identify one model (or even a few) *accepted widely* by the community. One common classification of proposals is into “probabilistic” and “non-probabilistic” systems [1–3]. The non-probabilistic systems may be further classified into various different types (e.g., social networks and cognitive); in contrast, the probabilistic systems usually have a common objective and structure: Probabilistic systems *(i)* assume a particular (probabilistic) model for principal behavior; and *(ii)* propose algorithms for approximating

[†] Full Paper will be published in a special collection dedicated to Gordon Plotkin (to appear). Available online: <http://www.brics.dk/~krukow>

[†] BRICS: Basic Research in Computer Science (www.brics.dk), funded by the Danish National Research Foundation.

the behavior of principals (i.e., prediction in the model). In systems based on such models, the trust information about a principal is information about its past behavior. Probabilistic systems usually do not classify this information as ‘good,’ ‘bad,’ ‘trustworthy’ or ‘untrustworthy;’ rather, such systems attempt to approximate the probability of various outcomes in a potential next interaction, given the past behavior. The probabilistic systems, known as “game-theoretical” in the terminology of Sabater and Sierra [2], are based on the view on trust of Gambetta: “(. . .) trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both *before* he can monitor such action (or independently of his capacity ever to be able to monitor it) *and* in a context in which it affects *his* own action” [4].

The contributions of this paper relate only to this, i.e., the probabilistic or *predictive* view on trust. We restrict ourselves to this approach for two primary reasons: (i) It is founded on well-understood mathematical theory and models (i.e., probability theory);¹ and (ii) the assumptions and objectives of systems are *precise*. Lack of formal models leads to in-clarity about the exact objectives of proposed systems; as Samuel Karlin was quoted to have said in a tribute lecture to honor R.A. Fisher: “The purpose of models is not to fit the data but to sharpen the questions.” We need to sharpen *our* questions. Our position is that for *any* approach to computational trust, probabilistic or not, it should be possible to: (i) Specify precisely the assumptions about the intended environments for the proposed system, i.e., in which applications does the system do well? (ii) Specify precisely the objective of the system, i.e., exactly what does the system compute?

The purpose of this paper is to highlight some of the advantages of formal probabilistic models. We show how formal probabilistic models enable systems that satisfy our two mentioned criteria of foundation and precision. Further, we sketch ideas towards a theoretically well-founded technique for comparing probabilistic systems in various different environments.

Outline. To illustrate probabilistic models, we develop a probabilistic extension of the event structure framework [6], used previously in the SECURE project [7] to model outcomes of interactions. The probabilistic event structure model generalizes previous probabilistic models from binary outcomes, e.g., each interaction is either ‘good’ or ‘bad,’ to multiple structured outcomes (technically, we obtain probabilities on the configurations of finite confusion-free event structures).² This is developed in Section 2 and Section 3.

To further illustrate the benefits of probabilistic models, we present preliminary ideas towards solving one open problem in computational trust research: Comparison of algorithms for probabilistic trust computation. We develop a measure which ‘scores’ a probabilistic algorithm in a given probabilistic model

¹ We follow the Bayesian approach to probability theory, as advocated by Jaynes [5].

² For those familiar with Bayesian analysis, we generalize models with beta priors to multiple structured outcomes where the prior pdfs are products of Dirichlet distributions.

of principal behavior. The measure is *parametric in the model* in the sense that for *any* probabilistic model, λ , the ‘score’ quantitatively measures how well a given algorithm approximates the true principal behavior in λ . Algorithms can then be compared by comparing their scores in various models. This work is presented in Section 4.

2 Probabilistic Event Structures

We have previously proposed to use event structures to model outcomes of principal interactions in distributed systems [6]; the model was used in the SECURE project [8, 7] to formalize the notions of outcomes and observations. However, we did not present a formal probabilistic model of principal behavior; hence, although we showed how to compute “trust values” which could be interpreted as probabilities of outcomes, there was no probabilistic model to justify the computation. In the next two sections, we augment the event structure framework with a probabilistic model which generalizes the model used in systems based on the beta distribution [9–12]. We show how to compute the probabilities of outcomes given a history of observations. This could be valuable in its own right; however, we would like to emphasize that our primary reason is to *illustrate an example* of a formal probabilistic model which enables “sharp” questions; the heart of this paper is really Section 4. The system proposed in Sections 2 and 3 is well-founded on probability theory and it generalizes many existing systems; however, it not yet practical: There are many issues it does not handle, e.g., dynamic principal behavior-change, lying reputation sources, multiple contexts, etc. We believe that the probabilistic models must be properly understood before we can deal with such issues in a theoretically well-founded manner. For further examples of probabilistic systems we refer to Aberer and Despotovic [1], and to most of the systems based on Bayesian analysis with beta prior distributions [9–12].

Observations and interaction outcomes. Agents in a distributed system obtain information by observing events which are typically generated by the reception or sending of messages. The structure of these message exchanges are given in the form of protocols known to both parties before interaction begins. By *behavioral observations*, we mean observations that the parties can make about specific runs of such protocols. These include information about the contents of messages, diversion from protocols, failure to receive a message within a certain time-frame, etc.

We will use the event-structure framework that we have proposed previously for modeling observations and outcomes in the SECURE project [6, 7]. The framework is suitable for our purpose as it provides a *generic* model for observations that is independent of any specific programming language. In the framework, the information that an agent has about the behavior of another agent p , is information about a number of (possibly active) protocol-runs with p , represented as a sequence of *sets of events*, $x_1x_2 \cdots x_n$, where event-set x_i

represents information about the i th initiated protocol-instance. Note that, as opposed to many existing systems, we are not *rating* the behavior of principals, but instead, we *record* the actual behavior, i.e., which events occurred in the interaction.

Event structures. We briefly recapture the basic definitions (for more details and examples, we refer to Nielsen and Krukow [6] and Krukow et al. [13,3]). An *event structure* is a triple $(E, \leq, \#)$ consisting of a set E of *events* which are partially ordered by \leq , the *necessity relation* (or causality relation), and $\#$ is a binary, symmetric, irreflexive relation $\# \subset E \times E$, called the *conflict relation*. The relations satisfy

$$[e] \stackrel{(\text{def})}{=} \{e' \in E \mid e' \leq e\} \text{ is finite; and}$$

$$\text{if } e \# e' \text{ and } e' \leq e'' \text{ then } e \# e''$$

for all $e, e', e'' \in E$. We say that two events are *independent* if they are not in either of the two relations.

The two basic relations on event structures have an intuitive meaning in our set up. An event may *exclude* the possibility of the occurrence of a number of other events; this is what the conflict relation models. The necessity relation is also natural: Some events are *only possible* when others have already occurred. Finally, if two events are in neither of the relations, they are said to be independent.

The event structure models the set of events that can occur in a protocol; however, due to the relations on event structures, not all sets of events can occur in a particular run. The notion of configurations formalizes this: A set of events $x \subseteq E$ is a *configuration* (of ES) if it satisfies the following two properties: Conflict free, i.e., for any $e, e' \in x : e \# e'$; Causally closed, i.e., for any $e \in x, e' \in E : e' \leq e \Rightarrow e' \in x$. Write \mathcal{C}_{ES} for the set of configurations of ES . Note that the set of all maximal configurations defines a set of mutually exclusive and exhaustive outcomes of an interaction.

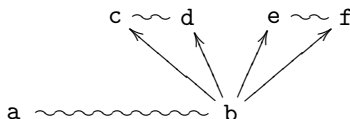
Histories. A finite configuration models information regarding *a single* interaction, i.e., a single run of a protocol. In general, the information that one principal possesses about another will consist of information about *several* protocol runs; the information about each individual run being represented by a configuration in the corresponding event structure. The concept of a (local) interaction history models this. An *interaction history* in ES is a finite ordered sequence of configurations, $h = x_1 x_2 \cdots x_n \in \mathcal{C}_{ES}^*$. The entries x_i (for $1 \leq i \leq n$) are called the *sessions* (of h).

Remarks. While the order of sessions is recorded (histories are *sequences*), in contrast, the order of *independent* events within *a single session* is not. Hence independence of events is a *choice of abstraction* one may make when designing

an event-structure model (because one is not interested in the particular order of events, or because the exact recording of the order of events is not feasible). However, note that this is not a limitation of event structures: In a scenario where this order of events is relevant (and observable), one can always use a “serialized” event structure in which this order of occurrences is recorded. A serialization of events consists of splitting the events in question into different events depending on the order of occurrence.

2.1 Confusion-Free Event Structures

We consider a special type of event structures, the *confusion free* event structures with independence, for which it is especially simple to adjoin probabilities [14]. Consider the following event structure (\sim represents conflict, and \rightarrow represents causality).



The events c and e are *independent*; as are c and f ; d and e ; and finally, d and f . However, in terms of the relations of event structures, *independent* simply means that both events can occur in the same configuration and in any order. Later we shall consider a probabilistic model where *independence* means also *probabilistic independence*. To do this we first introduce a notion of *cells* and *immediate conflict* [14]. In the following $ES = (E, \leq, \#)$ is a fixed event structure.

Write $[e]$ for $[e] \setminus \{e\}$, and say that events $e, e' \in E$ are in *immediate conflict*, writing $e \#_{\mu} e'$, if $e \# e'$ and both $[e] \cup [e']$ and $[e'] \cup [e]$ are configurations. It is easy to see that a conflict $e \# e'$ is immediate if-and-only-if there exists a configuration x where both e and e' are enabled (i.e., can occur in x). For example the conflict $a \# b$ is immediate, whereas $a \# c$ is not.

A *partial cell* is a non-empty set of events $c \subseteq E$ such that $e, e' \in c$ implies $e \#_{\mu} e'$ and $[e] = [e']$. A maximal partial cell is called a *cell*. There are three cells in the above event structure: $\{a, b\}$, $\{c, d\}$ and $\{e, f\}$. Cells represent choices; in probabilistic event structures, *probabilistic choices*. A *confusion free* event structure is an event structure where (the reflexive closure of) immediate conflict is an equivalence relation and *within cells* (i.e., that $e \#_{\mu} e'$ implies $[e] = [e']$). We suspect that most event structures for simple interaction protocols are confusion free.

In confusion-free event structures, if an event e of a cell c is enabled at configuration x , then all events $e' \in c$ are also enabled at x . If the event structure is also finite, a maximal configuration (i.e., an outcome of an interaction) is obtained by starting with the empty configuration and then repeating the following: Let C be the set of cells that are enabled in the current configuration. If C is empty then stop: The current configuration is maximal; otherwise, non-deterministically select a cell $c \in C$, and then non-deterministically select, or probabilistically sample, an event $e \in c$; finally, update the current configuration by adding e .

The following notion of cell-valuation formalizes probabilistic sampling in cells.

Definition 1 (Cell valuation, Varacca et al. [14]). *When $f : X \rightarrow [0, +\infty]$ is a function, for every $Y \subseteq X$, we define $f[Y] = \sum_{y \in Y} f(y)$. A cell valuation on a confusion free event structure $ES = (E, \leq, \#)$ is a function $p : E \rightarrow [0, 1]$ such that for every cell c , we have $p[c] = 1$.*

If cell choices are probabilistic, say given by a cell-valuation p , and if we assume independence between cells, then one can obtain the probability of any configuration x (i.e., any outcome) as the product of the probabilities of each event in x given p .

3 A probabilistic framework

We will be concerned with adjoining probabilities to the configurations of a finite confusion-free event structure ES . As mentioned in the previous section, we can do this by finding a cell valuation $p : E \rightarrow [0, 1]$, or, equivalently, for each cell c , a function $p_c : c \rightarrow [0, 1]$ with $p_c[c] = 1$. The functions p_c should be derived from the past experience obtained from interacting with an entity in ES . In the following paragraph, we state the assumptions about the behavior of entities in our model. We then proceed to (i) find abstractions that preserve sufficient information under the model; and (ii) derive equations for the predictive probabilities, i.e., answering “what is the probability of outcome x in the next interaction with entity q (in the model)?”

The model. Let us consider a finite and confusion-free event structure ES . Let us write $C(ES)$ for the set of cells (which are then the equivalence classes of immediate conflict). Write $C(ES) = \{c_1, c_2, \dots, c_k\}$, and let $N_i = |c_i|$ for each i . Let us make the following assumptions about principal behavior, and write λ_{DES} for the assumptions of this model:

Each principal’s behavior is so that there are fixed parameters such that at each interaction we have, *independently of anything we know about other interactions*, the probability $\theta_{c,e}$ for event e at cell c .

Each θ_{c_i} for $c_i \in C(ES)$ is a vector of size N_i such that $\sum_{e \in c_i} \theta_{c_i,e} = 1$. Hence, the collection $\theta = (\theta_c \mid c \in C(ES))$ defines a cell valuation on ES . For each configuration $x \in \mathcal{C}_{ES}$ the probability of obtaining x in any run of ES with a principal parametrized by θ is

$$P(x \mid \theta \lambda_{DES}) = \prod_{e \in x} \theta_e \tag{1}$$

where θ_e is defined by $\theta_{c,e}$ where c is the unique cell with $e \in c$.

The goal of our probabilistic framework is to estimate the parameters θ given a prior distribution and data regarding past interactions. In the λ_{DES} model, we

need only estimate the parameters of each cell c , i.e., θ_c , to obtain a probability distribution on configurations (Equation 1). Furthermore, it follows from λ_{DES} that given a sequence $h = x_1x_2 \cdots x_n \in \mathcal{C}_{ES}^*$ of observed data (about a fixed principal), we need only keep track of event counts of h to estimate the parameters of each θ_c (e.g., according to λ_{DES} , the order of sessions does not matter). This means that an event count, i.e., a function $\mathbf{X} : E \rightarrow \mathbb{N}$, is sufficient information to estimate θ_c for each cell c .

To estimate the parameters θ , we shall use Bayesian analysis. Hence, we need prior distributions. It turns out that the family of Dirichlet distributions are a family of conjugate prior distributions to the family of multinomial trials. A family F of distributions is a *conjugate prior for a likelihood function* L if whenever the prior distribution belongs to F then also the posterior distribution belongs to F . The use of conjugate priors represents a computational convenience common for Bayesian analysis: The distributions always maintain the same algebraic form (i.e., that of family F). As we shall see, the uniform distribution belongs to the Dirichlet family; this means that the prior, if desired, can be chosen not to bias any event over another.

Since each sampling from a cell is a multinomial trial (according to λ_{DES}), we use Dirichlet distributions as our prior distributions. Specifically, a prior Dirichlet distribution is assigned to each cell c of ES . Event counts are then used to update the Dirichlet at each cell. Hence, at any time we have, for each cell c , a Dirichlet distribution f_c on the parameters θ_c of the events of that cell; we show that the probability of an outcome $x \subseteq E$ is then the product of certain expectations of these distributions. We explain the Dirichlet distributions in the following.

3.1 The Dirichlet distribution.

The Dirichlet family \mathcal{D} of order K , where $2 \leq K \in \mathbb{N}$, is a parametrized collection of continuous probability density functions defined on $[0, 1]^K$. There are K parameters of positive reals, $\alpha = (\alpha_i)_{i=1}^K$, that select a specific Dirichlet distribution from the family. For a variable $\theta \in [0, 1]^K$, the pdf $\mathcal{D}(\theta | \alpha)$ is given by the following:

$$\mathcal{D}(\theta | \alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1}$$

(where Γ is the Gamma function, $\Gamma(z) = \int_0^\infty dt t^{z-1} e^{-t}$, for $z > 0$). Define $[\alpha] = \sum_j \alpha_j$; the expected value and variance of each parameter θ_i are given by

$$\mathbf{E}_{\mathcal{D}(\theta|\alpha)}(\theta_i) = \frac{\alpha_i}{[\alpha]}, \quad \sigma_{\mathcal{D}(\theta|\alpha)}^2(\theta_i) = \frac{\alpha_i([\alpha] - \alpha_i)}{[\alpha]^2([\alpha] + 1)}$$

A conjugate prior. Consider sequences of independent experiments with K 'ary outcomes ($K \in \mathbb{N}$), each yielding outcome i with some fixed probability θ_i ; let us call such experiments multinomial trials (in our framework, such experiments will correspond to probabilistic event-choices at a cell). Let $\lambda_{\mathcal{D}}$ denote background information encoding this. Let X_i , for $i = 1, 2, \dots, n$, represent the i th trial, i.e.,

$X_i = j$ is the statement that the i th trial has outcome $j \in \{1, 2, \dots, K\}$. Let \mathbf{X} be a conjunction of n statements, $(Z_i)_{i=1}^n$, of the form:

$$Z_i \equiv (X_i = j_i), \quad \text{where each } j_i \in \{1, \dots, K\}.$$

Suppose there are m_j statements of the form $X_i = j$, and let $\theta = (\theta_i)_{i=1}^K$. Then, by definition of multinomial trials, we have the following likelihood:

$$P(\mathbf{X} \mid \theta \lambda_{\mathcal{D}}) = \prod_{i=1}^n P(Z_i \mid \theta \lambda_{\mathcal{D}}) = \prod_{i=1}^K \theta_i^{m_i}.$$

The Dirichlet distributions constitute a family of conjugate prior distributions for this likelihood. In other words, if the prior distribution on θ , say $g(\theta \mid \lambda_{\mathcal{D}})$, is a Dirichlet $\mathcal{D}(\theta \mid \alpha)$, for $\alpha = (\alpha_i)_{i=1}^K$, then the posterior given data \mathbf{X} (obtained via Bayes' Theorem), $g(\theta \mid \mathbf{X} \lambda_{\mathcal{D}})$, is also Dirichlet. In the language equations:

$$g(\theta \mid \mathbf{X} \lambda_{\mathcal{D}}) = g(\theta \mid \lambda_{\mathcal{D}}) \frac{P(\mathbf{X} \mid \theta \lambda_{\mathcal{D}})}{P(\mathbf{X} \mid \lambda_{\mathcal{D}})}.$$

In fact, it is not hard to show that $g(\theta \mid \mathbf{X} \lambda_{\mathcal{D}}) = \mathcal{D}(\theta \mid \alpha_1 + m_1, \dots, \alpha_K + m_K)$.

Note, choosing $\alpha_i = 1$ (for all i) in the prior gives the uniform prior distribution.

The predictive probability ($\lambda_{\mathcal{D}}$). Now, let $Z_{n+1} \equiv (X_{n+1} = i)$, then one can interpret $P(Z_{n+1} \mid \mathbf{X} \lambda_{\mathcal{D}})$ as a predictive probability: Given no direct knowledge of θ , but only past evidence (\mathbf{X}) and the model ($\lambda_{\mathcal{D}}$), then $P(Z_{n+1} \mid \mathbf{X} \lambda_{\mathcal{D}})$ is the probability that the next trial will result in a type i outcome. It is easy to show that:

$$P(Z_{n+1} \mid \mathbf{X} \lambda_{\mathcal{D}}) = \mathbf{E}_{g(\theta \mid \mathbf{X} \lambda_{\mathcal{D}})}(\theta_i) = \frac{\alpha_i + m_i}{[\alpha] + n}$$

(since $g(\theta \mid \mathbf{X} \lambda_{\mathcal{D}})$ is $\mathcal{D}(\theta \mid (\alpha_1 + m_1, \alpha_2 + m_2, \dots, \alpha_K + m_K))$ and $\sum_i m_i = n$).

To summarize, in the Dirichlet model, $\lambda_{\mathcal{D}}$, one can compute the probability of outcome i in the next multinomial trial as the expectation of the i th parameter of the Dirichlet pdf $g(\theta \mid \mathbf{X} \lambda_{\mathcal{D}})$ which results via Bayesian updating given history \mathbf{X} .

3.2 Dirichlets on cells

Let us return to our probabilistic model. For each cell $c \in \mathcal{C}(ES)$ we will associate a prior distribution on the parameters θ_c determining the behavior of a fixed principal for the events of c . As we interact, we obtain data about these parameters, and the distribution on each cell is updated via Bayes' Theorem. Each cell $c \in \mathcal{C}(ES)$ presents a choice between the mutually exclusive and exhaustive events of c , and by the assumptions of $\lambda_{\mathcal{D}ES}$ a sequence of such choices from c is a sequence multinomial trials. We use Dirichlet priors on each cell so that the posterior distributions are also Dirichlets. At any time, we obtain the

predictive probability of the next interaction resulting in a particular configuration by multiplying the expectations (according to the current cell distributions) of the parameters for each event in the configuration.

Let us be precise: Let $f_c(\theta_c \mid \lambda_{\mathcal{DES}})$ denote the prior distribution on the parameters for each cell $c \in \mathcal{C}(ES)$ (when interacting with a fixed principal). Let α_c be a vector of positive real numbers of size $N_c = |c|$; we take,

$$f_c(\theta_c \mid \lambda_{\mathcal{DES}}) = \mathcal{D}(\theta_c \mid \alpha_c) = \frac{\Gamma(\sum_{i=1}^{N_c} \alpha_{c,i})}{\prod_{i=1}^{N_c} \Gamma(\alpha_{c,i})} \prod_{i=1}^{N_c} \theta_{c,i}^{\alpha_{c,i}-1}$$

For example, taking $\alpha_{c,j} = 1$ (for all j) gives the uniform distribution. Let $\mathbf{X} : E \rightarrow \mathbb{N}$ be an event count modeling data about past runs with a specific principal. Let $\mathbf{X}_c = \mathbf{X}|_c$ (i.e., the restriction of \mathbf{X} to cell c), then the posterior pdf is given by the following: Assume that $c = \{e_1, e_2, \dots, e_{N_c}\}$ then,

$$\begin{aligned} f_c(\theta_c \mid \mathbf{X}\lambda_{\mathcal{DES}}) &= \frac{\Gamma(\sum_{i=1}^{N_c} \alpha_{c,i} + \mathbf{X}(e_i))}{\prod_{i=1}^{N_c} \Gamma(\alpha_{c,i} + \mathbf{X}(e_i))} \prod_{i=1}^{N_c} \theta_{c,i}^{\alpha_{c,i} + \mathbf{X}(e_i) - 1} \\ &= \mathcal{D}(\theta_c \mid \alpha_c + \mathbf{X}_c) \end{aligned}$$

Hence, each event count $\mathbf{X} : E \rightarrow \mathbb{N}$ can be used to do Bayesian updating of the distribution at each cell.

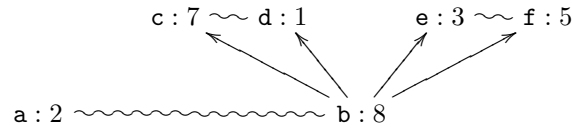
The predictive probability ($\lambda_{\mathcal{DES}}$). By Bayesian updating, we obtain a Dirichlet distribution for each cell c of ES . Let \mathbf{X} be an event count corresponding to n previously observed configurations, and let Z be the proposition that “the $(n+1)$ ’st interaction results in outcome i ” (where $1 \leq i \leq M$ and M is the number of maximal configurations in ES). Let x_i be the i ’th maximal configuration, and for $e \in x_i$ let $c(e)$ denote the unique cell c with $e \in c$. The predictive probability is the product of the expectations of each of the cell parameters.

$$P(Z \mid \mathbf{X}\lambda_{\mathcal{DES}}) = \prod_{e \in x_i} \mathbf{E}_{f_{c(e)}(\theta_{c(e)} \mid \mathbf{X}\lambda_{\mathcal{DES}})}(\theta_{c(e),e}) = \prod_{e \in x_i} \frac{\alpha_{c(e),e} + \mathbf{X}(e)}{[\alpha_{c(e)}] + \mathbf{X}[c(e)]}$$

3.3 Summary

We have presented a probabilistic model $\lambda_{\mathcal{DES}}$ based on probabilistic confusion-free event structures. The model generalizes previous work on probabilistic models using binary outcomes and beta priors. In our model, given a past history with a principal we need only remember the event counts of the past, i.e., a function $\mathbf{X} : E \rightarrow \mathbb{N}$. Given such an event count, there is a unique probability of any particular configuration occurring as the next interaction. We have derived equations for this probability and it is easily computed in real systems.

For example, suppose we have the following event count, \mathbf{X} .



With the following prior pdfs for the cells:

$$\begin{aligned} f_{\text{ab}}((\theta_a, \theta_b) \mid \lambda_{\mathcal{DES}}) &= \mathcal{D}((\theta_a, \theta_b) \mid (1, 1)), \\ f_{\text{cd}}((\theta_c, \theta_d) \mid \lambda_{\mathcal{DES}}) &= \mathcal{D}((\theta_c, \theta_d) \mid (1, 1)) \text{ and} \\ f_{\text{ef}}((\theta_e, \theta_f) \mid \lambda_{\mathcal{DES}}) &= \mathcal{D}((\theta_e, \theta_f) \mid (1, 1)); \end{aligned}$$

this count gives rise to three updated Dirichlets

$$\begin{aligned} f_{\text{ab}}((\theta_a, \theta_b) \mid \mathbf{X}\lambda_{\mathcal{DES}}) &= \mathcal{D}((\theta_a, \theta_b) \mid (1 + 2, 1 + 8)), \\ f_{\text{cd}}((\theta_c, \theta_d) \mid \mathbf{X}\lambda_{\mathcal{DES}}) &= \mathcal{D}((\theta_c, \theta_d) \mid (1 + 7, 1 + 1)) \text{ and} \\ f_{\text{ef}}((\theta_e, \theta_f) \mid \mathbf{X}\lambda_{\mathcal{DES}}) &= \mathcal{D}((\theta_e, \theta_f) \mid (1 + 3, 1 + 5)). \end{aligned}$$

As an example, the probability of configuration $\{b, c\}$ is

$$P(\{b, c\} \mid \mathbf{X}\lambda_{\mathcal{DES}}) = \frac{9}{12} \times \frac{8}{10} = \frac{3}{5}.$$

4 Advantages of Probabilistic Models: A preliminary idea

While the purpose of models may not be to fit the data but to sharpen the questions, good models must do both! Our probabilistic models must be more realistic. For example, the beta model of principal behavior (which we consider to be state-of-the-art) assumes that for each principal p there is a single fixed parameter θ_p so at each interaction, *independently of anything else we know*, there is probability θ_p for a ‘good’ outcome and probability $1 - \theta_p$ for a ‘bad’ outcome. For *some* applications, one might argue that this is unrealistic, e.g.: (i) The parameter θ_p is fixed, independent of time, i.e., no dynamic behavior; and (ii) principal p ’s behavior when interacting with us is likely to depend on our behavior when interacting with p ; let us call this property ‘recursive behavior.’ (Note, the same issues are present in the Dirichlet model $\lambda_{\mathcal{D}}$ and the Dirichlet-Event-Structure model $\lambda_{\mathcal{DES}}$ that we propose). Some beta-based reputation systems attempt to deal with the first problem by introducing so-called “forgetting factors;” essentially this amounts to choosing a number $0 \leq \delta \leq 1$, and then each time the parameters (α, β) of the pdf for θ_p are updated, they are also scaled with δ , e.g., when observing a single ‘good’ interaction, (α, β) become $(\alpha\delta + 1, \beta\delta)$. In effect, this performs a form of exponential “decay” on the parameters. The idea is that information about old interactions should weigh less than information about new ones; however, this represent a departure from the probabilistic beta model, where all interactions “weigh the same.” Since a new model is *not* introduced, i.e., to formalize this preference towards newer information, it is not clear what the exact benefits of forgetting factors are, e.g., why exponential decay as opposed to linear? As far as we know, no-one has considered the ‘recursive behavior’ problem before.

The notion of context is also relevant for computational trust models, as have been recognized by many. Given a single-context model, one can obtain a

multi-context model by instantiating the single-context model in each context. However, as Sierra and Sabater [2] argue, this is too naive: The goal of a true multi-context model is not just to *model* multiple contexts, but to provide the basis for transferring information from one context to another related context. To our knowledge, there are no techniques that deal with this problem within the field of trust and reputation.

Finally, we believe (as do Sierra and Sabater [2]) that our field is lacking a way of comparing the qualities of the many proposed trust-based systems. Sierra and Sabater propose that our field develop “(...) test-beds and frameworks to evaluate and compare the models under a set of representative and common conditions” [2]. We agree with Sierra and Sabater (note that “a set of representative and common conditions” could be a formal probabilistic model).

In the following, we sketch ideas towards solving this last problem: We develop what one might call “a theoretical test-bed” for comparing systems for probabilistic trust computation.

4.1 Towards Comparing Probabilistic Trust-based Systems

We shall propose a generic measure to “score” specific probabilistic trust-based systems in a particular environment (i.e., “a set of representative and common conditions”). The score, which is based on the so-called Kullback-Leibler divergence, is a measure of how well an algorithm approximates the “true” probabilistic behavior of principals.

Consider a probabilistic model of principal behavior, say λ . We consider only the behavior of a single fixed principal p , and we consider only algorithms that attempt to solve the following problem: Suppose we are given an interaction history $\mathbf{X} = [(x_1, t_1), (x_2, t_2), \dots, (x_n, t_n)]$ obtained by interacting n times with principal p , observing outcome x_i at time t_i . Suppose also that there are m possible outcomes (y_1, \dots, y_m) for each interaction. The goal of a probabilistic trust-based algorithm, say \mathcal{A} , is to approximate a distribution on the outcomes (y_1, \dots, y_m) given this history \mathbf{X} . That is, \mathcal{A} satisfies:

$$\mathcal{A}(y_i | \mathbf{X}) \in [0, 1] \text{ (for all } i), \quad \sum_{i=1}^m \mathcal{A}(y_i | \mathbf{X}) = 1.$$

We assume that the probabilistic model, λ , defines the following probabilities: $P(y_i | \mathbf{X}\lambda)$, i.e., the probability of “ y_i in the next interaction given a past history of \mathbf{X} ” and $P(\mathbf{X} | \lambda)$, i.e., the “*a priori* probability of observing sequence \mathbf{X} in the model.”³

Now, $(P(y_i | \mathbf{X}\lambda) | i = 1, 2, \dots, m)$ defines the true distribution on outcomes for the next interaction (according to the model); in contrast, $(\mathcal{A}(y_i | \mathbf{X}) | i = 1, 2, \dots, m)$ attempts to approximate this distribution. The Kullback-Leibler

³ In a way, this model takes into account also the ‘recursive behavior’ problem: The probabilities $P(y_i | \mathbf{X}\lambda)$ and $P(y_i | \lambda)$ are distinguished. We have not yet given this further thoughts.

divergence [15], which is closely related to Shannon entropy, is a measure of the distance from a true distribution to an approximation of that distribution. The Kullback-Leibler divergence from distribution $\hat{p} = (p_1, p_2, \dots, p_m)$ to distribution $\hat{q} = (q_1, q_2, \dots, q_m)$ on a finite set of m outcomes, is given by

$$D_{\text{KL}}(\hat{p} \parallel \hat{q}) = \sum_{i=1}^m p_i \log_2 \left(\frac{p_i}{q_i} \right)$$

(any log-base could be used). The Kullback-Leibler divergence is almost a distance (in the mathematical sense), but the symmetry property fails. That is D_{KL} satisfies $D_{\text{KL}}(\hat{p} \parallel \hat{q}) \geq 0$ and $D_{\text{KL}}(\hat{p} \parallel \hat{q}) = 0$ only if $\hat{p} = \hat{q}$. The asymmetry comes from considering one distribution as “true” and the other as approximating.

For each n let \mathbf{O}^n denote the set of interaction histories of length n . Let us define, for each n , the n 'th *expected Kullback-Leibler divergence from λ to \mathcal{A}* :

$$D_{\text{KL}}^n(\lambda \parallel \mathcal{A}) \stackrel{\text{(def)}}{=} \sum_{\mathbf{X} \in \mathbf{O}^n} P(\mathbf{X} \mid \lambda) D_{\text{KL}}(P(\cdot \mid \mathbf{X}\lambda) \parallel \mathcal{A}(\cdot \mid \mathbf{X})),$$

that is,

$$D_{\text{KL}}^n(\lambda \parallel \mathcal{A}) = \sum_{\mathbf{X} \in \mathbf{O}^n} P(\mathbf{X} \mid \lambda) \left(\sum_{i=1}^m P(y_i \mid \mathbf{X}\lambda) \log_2 \left(\frac{P(y_i \mid \mathbf{X}\lambda)}{\mathcal{A}(y_i \mid \mathbf{X})} \right) \right).$$

Note that, for each input sequence $\mathbf{X} \in \mathbf{O}^n$ to the algorithm, we evaluate its performance as $D_{\text{KL}}(P(\cdot \mid \mathbf{X}\lambda) \parallel \mathcal{A}(\cdot \mid \mathbf{X}))$; however, we accept that some algorithms may perform poorly on very unlikely training sequences, \mathbf{X} . Hence, we weigh the penalty on input \mathbf{X} , i.e., $D_{\text{KL}}(P(\cdot \mid \mathbf{X}\lambda) \parallel \mathcal{A}(\cdot \mid \mathbf{X}))$, with the intrinsic probability of sequence \mathbf{X} ; that is, we compute the *expected* Kullback-Leibler divergence.

The Kullback-Leibler divergence is a well-established measure in statistic; however, to our knowledge, the measure D_{KL}^n on probabilistic algorithms is new. Due to the relation to Shannon's Information Theory, one can interpret $D_{\text{KL}}^n(\lambda \parallel \mathcal{A})$ quantitatively as the expected number of bits of information one would gain if one would know the true distribution instead of \mathcal{A} 's approximation on n -length training sequences.

An example. For an example of our measure, we compare the beta-based algorithm of Mui et al. [10] with the maximum-likelihood algorithm of Aberer and Despotovic [16]. We can compare these because they both deploy the same fundamental assumptions:

Assume that the behavior of each principal is so that there is a fixed parameter such that at each interaction we have, independently of anything we know about other interactions, the probability θ for a ‘success’ and therefore probability $1 - \theta$ for ‘failure.’

This gives us the *beta model*, $\lambda_{\mathbf{B}}$. Let s stand for ‘success’ and f stand for ‘failure,’ and let $\mathbf{X} \in \{s, f\}^n$ for some $n > 0$.

We have the following likelihood for any $\mathbf{X} \in \{s, f\}^n$:

$$P(\mathbf{X} \mid \lambda_{\mathbf{B}}\theta) = \theta^{N_s(\mathbf{X})}(1 - \theta)^{N_f(\mathbf{X})}$$

(where $N_x(\mathbf{X})$ denotes the number of x occurrences in \mathbf{X}).

Let \mathcal{A} denote the algorithm of Mui et al., and let \mathcal{B} denote the algorithm of Aberer and Despotovic. Then,

$$\mathcal{A}(s \mid \mathbf{X}) = \frac{N_s(\mathbf{X}) + 1}{n + 2} \text{ and } \mathcal{A}(f \mid \mathbf{X}) = \frac{N_f(\mathbf{X}) + 1}{n + 2},$$

and it is easy to show that

$$\mathcal{B}(s \mid \mathbf{X}) = \frac{N_s(\mathbf{X})}{n} \text{ and } \mathcal{B}(f \mid \mathbf{X}) = \frac{N_f(\mathbf{X})}{n}.$$

For each choice of $\theta \in [0, 1]$, and each choice of training-sequence length, we can compare the two algorithms by computing and comparing $D_{\text{KL}}^n(\lambda_{\mathbf{B}}\theta \parallel \mathcal{A})$ and $D_{\text{KL}}^n(\lambda_{\mathbf{B}}\theta \parallel \mathcal{B})$. For example:

Theorem 1. *If $\theta = 0$ or $\theta = 1$ then the algorithm \mathcal{B} of Aberer and Despotovic [16] computes a better approximation of principal behavior than the algorithm \mathcal{A} of Mui et al. [10]. In fact, \mathcal{B} always computes the exact probability of success on any possible training sequence.*

Proof. Assume that $\theta = 0$, and let $n > 0$. The only sequence of length n which has non-zero probability is f^n , and we have $\mathcal{B}(f \mid f^n) = 1$; in contrast, $\mathcal{A}(f \mid f^n) = \frac{n+1}{n+2}$, and $\mathcal{A}(s \mid f^n) = \frac{1}{n+2}$. Since $P(s \mid f^n \lambda_{\mathbf{B}}\theta) = \theta = 0 = \mathcal{B}(s \mid f^n)$ and $P(f \mid f^n \lambda_{\mathbf{B}}\theta) = 1 - \theta = 1 = \mathcal{B}(f \mid f^n)$, we have

$$D_{\text{KL}}^n(\lambda_{\mathbf{B}}\theta \parallel \mathcal{B}) = 0.$$

Since $D_{\text{KL}}^n(\lambda_{\mathbf{B}}\theta \parallel \mathcal{A}) > 0$ we are done (the argument for $\theta = 1$ is similar). \square

Now let us compare \mathcal{A} and \mathcal{B} with $0 < \theta < 1$. Since \mathcal{B} assigns probability 0 to s on input f^k (for all $k \geq 1$) which results in $D_{\text{KL}}^n(\lambda \parallel \mathcal{B}) = \infty$, then according to our measure D_{KL}^n , algorithm \mathcal{A} is always better than \mathcal{B} . However, this results from a property of the Kullback-Leibler measure: Given two distribution $\hat{p} = (p_1, \dots, p_n)$ and $\hat{q} = (q_1, \dots, q_n)$, if one of the ‘‘real’’ probabilities, p_i is non-zero and the corresponding ‘‘approximating’’ probability q_i is zero, then $D_{\text{KL}}(\hat{p} \parallel \hat{q}) = \infty$. To obtain a stronger and more informative result, we shall consider a continuum of algorithms, denoted \mathcal{A}_ϵ for a real number $0 < \epsilon < 1$, defined as

$$\mathcal{A}_\epsilon(s \mid \mathbf{X}) = \frac{N_s(\mathbf{X}) + \epsilon}{n + 2\epsilon} \text{ and } \mathcal{A}_\epsilon(f \mid \mathbf{X}) = \frac{N_f(\mathbf{X}) + \epsilon}{n + 2\epsilon}.$$

One can think of \mathcal{A}_ϵ as approximating \mathcal{B} ($= \mathcal{A}_0$) for small epsilon.

We have the following theorem which compares \mathcal{A}_ϵ and the algorithm of Mui et al., \mathcal{A} , in a continuum of different environments.

Theorem 2. Let $\lambda_{\mathbf{B}}$ be the beta model with parameter $\theta \in [\frac{1}{2} - \frac{1}{\sqrt{12}}, \frac{1}{2} + \frac{1}{\sqrt{12}}]$. For any $n \geq 0$ we have

$$D_{\text{KL}}^n(\lambda_{\mathbf{B}}\theta \parallel \mathcal{A}) < D_{\text{KL}}^n(\lambda_{\mathbf{B}}\theta \parallel \mathcal{A}_\epsilon),$$

for all $\epsilon \in (0, 1)$.

Proof. See the full paper, to appear soon. □

What does this mean? Another way to put it is that if θ is in the interval $[\frac{1}{2} - \frac{1}{\sqrt{12}}, \frac{1}{2} + \frac{1}{\sqrt{12}}]$ (approximately), then *independently* of training-sequence length (n), then the algorithm of Mui et al. is better (on average) than *any* algorithm \mathcal{A}_ϵ (for $0 < \epsilon < 1$). To our knowledge, this is the first *theorem* which compares two algorithms for trust computation: All previous comparisons have been via computer simulations. In fact, it is not so much the concrete comparison of algorithms \mathcal{A} and \mathcal{B} that interests us; rather, our message is that using probabilistic models enables the *possibility* of such theoretical comparisons. Notice that without formal probabilistic models we would be unable to even *state* precisely such theorems.

5 Conclusion

Our “position” on computational trust research is that any proposed system should be able to answer two fundamental questions precisely: What are the assumptions about the intended environments for the system? And, what is the objective of the system? An advantage of formal probabilistic models is that they enable rigorous answers to these questions. To illustrate this point, we have presented an example of a formal probabilistic model, λ_{DES} . There are other examples: The beta model specifies the assumption of the computational trust model of Jøsang et al. [9], and under these assumptions their algorithm computes the probability of a principal well-behaving in the next interaction.

There are further benefits of formal probabilistic models: As we have illustrated, it is possible to compare two algorithms, say \mathcal{X} and \mathcal{Y} , under the same type of principal behavior, say model λ , by examining which algorithm best approximates the true principal behavior (as specified by λ). For example, we propose to compute and compare:

$$D_{\text{KL}}^n(\lambda \parallel \mathcal{X}) \text{ and } D_{\text{KL}}^n(\lambda \parallel \mathcal{Y}).$$

Note, no simulations of algorithms \mathcal{X} and \mathcal{Y} are necessary; the numbers give a theoretical justification, e.g., stating that “in environment λ , on the average, algorithm \mathcal{X} outperforms algorithm \mathcal{Y} on training sequences of length n .” If one can further show that this holds for all n , or for all n greater than some number, this gives a way of saying that \mathcal{X} is better than \mathcal{Y} . Another type of property one might desire is the following: Suppose \mathcal{X} satisfies for each $\epsilon > 0$ there exists an $N > 0$ so that for all $n \geq N$ we have $D_{\text{KL}}^n(\lambda \parallel \mathcal{X}) < \epsilon$. This means that given

a long enough training sequence, algorithm \mathcal{X} approximates the true principal behavior to an arbitrary precision.

We have further results which will be published in the full paper. We consider \mathcal{A}_ϵ for *all* epsilon, not just $[0, 1]$. We show that for each choice of θ there is an optimal ϵ_θ for which $\mathcal{A}_{\epsilon_\theta}$ is best among algorithms \mathcal{A}_ϵ . Recall that the results of Section 4 were all based on the simple beta model, $\lambda_{\mathbf{B}}$. We illustrate how our measure is parametric by considering a probabilistic model of dynamic principal behavior based on Hidden Markov Models. We show how one can use our measure “out-of-the-box” to compare algorithms working in this model.

Acknowledgments. We thank the anonymous reviewers for pointing to several places where clarification of our position and intention was necessary.

References

1. Despotovic, Z., Aberer, K.: P2P reputation management: Probabilistic estimation vs. social networks. *Computer Networks* **50**(4) (2006) 485–500
2. Sabater, J., Sierra, C.: Review on computational trust and reputation models. *Artificial Intelligence Review* **24**(1) (2005) 33–60
3. Krukow, K.: Towards a Theory of Trust for the Global Ubiquitous Computer. PhD thesis, University of Aarhus, Denmark (2006) ; available online (submitted): <http://www.brics.dk/~krukow>.
4. Gambetta, D.: Can we trust trust? In Gambetta, D., ed.: *Trust: Making and Breaking Cooperative Relations*. University of Oxford, Department of Sociology (2000) 213–237 Chapter 13. Electronic edition <http://www.sociology.ox.ac.uk/papers/gambetta213-237.pdf>.
5. Jaynes, E.T.: *Probability Theory: The Logic of Science*. Cambridge University Press, The Edinburgh Building, Cambridge, CB2 2RU, United Kingdom (2003)
6. Nielsen, M., Krukow, K.: On the formal modelling of trust in reputation-based systems. In Karhumäki, J., Maurer, H., Paun, G., Rozenberg, G., eds.: *Theory Is Forever: Essays Dedicated to Arto Salomaa on the Occasion of His 70th Birthday*. Volume 3113 of *Lecture Notes in Computer Science*. Springer Verlag (2004) 192–204
7. Cahill, V., Seigneur, J.M.: The SECURE website. <http://secure.dsg.cs.tcd.ie> (2004)
8. Cahill, V., Gray *et al.*, E.: Using trust for secure collaboration in uncertain environments. *IEEE Pervasive Computing* **2**(3) (2003) 52–61
9. Jøsang, A., Ismail, R.: The beta reputation system. In: *Proceedings from the 15th Bled Conference on Electronic Commerce, Bled*. (2002)
10. Mui, L., Mohtashemi, M., Halberstadt, A.: A computational model of trust and reputation (for ebusinesses). In: *Proceedings from 5th Annual Hawaii International Conference on System Sciences (HICSS'02)*, IEEE (2002) 188
11. Buchegger, S., Le Boudec, J.Y.: A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In: *P2PEcon 2004*. (2004)
12. Teacy, W.T.L., Patel, J., Jennings, N.R., Luck, M.: Coping with inaccurate reputation sources: experimental analysis of a probabilistic trust model. In: *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, New York, NY, USA, ACM Press (2005) 997–1004

13. Krukow, K., Nielsen, M., Sassone, V.: A logical framework for reputation systems. Submitted. Available online www.brics.dk/~krukow (2006)
14. Varacca, D., Völzer, H., Winskel, G.: Probabilistic event structures and domains. In Gardner, P., Yoshida, N., eds.: Proceedings from 15th International Conference on Concurrency Theory (CONCUR'04). Volume 3170 of Lecture Notes in Computer Science., Springer (2004) 481–496
15. Kullback, S., Leibler, R.A.: On information and sufficiency. *Annals of Mathematical Statistics* **22**(1) (1951) 79–86
16. Despotovic, Z., Aberer, K.: A probabilistic approach to predict peers' performance in P2P networks. In: Proceedings from the Eighth International Workshop on Cooperative Information Agents (CIA 2004). Volume 3191 of Springer Lecture Notes in Computer Science., Springer (2004) 62–76