A Bayesian model for event-based trust

Mogens Nielsen [1] Karl Krukow [2]

BRICS
University of Aarhus, Denmark [3]

Vladimiro Sassone

ECS, University of Southampton

Abstract

The application scenarios envisioned for '*global ubiquitous computing*' have unique requirements that are often incompatible with traditional security paradigms. One alternative currently being investigated is to support security decision-making by explicit representation of principals' trusting relationships, i.e., via systems for *computational trust*. We focus here on systems where trust in a computational entity is interpreted as the expectation of certain future behaviour based on behavioural patterns of the past, and concern ourselves with the foundations of such probabilistic systems. In particular, we aim at establishing formal probabilistic models for computational trust and their fundamental properties. In the paper we define a *mathematical measure* for quantitatively comparing the effectiveness of probabilistic computational trust systems in various environments. Using it, we compare some of the systems from the computational trust literature; the comparison is derived formally, rather than obtained via experimental simulation as traditionally done. With this foundation in place, we formalise a general notion of information about past behaviour, based on *event structures*. This yields a flexible trust model where the probability of complex protocol outcomes can be assessed.

1 Introduction

Part of the Grand Challenge of a science for *global ubiquitous computing* (GUC) [5] is to find alternatives to existing approaches to access control, and, more gener-

[1] Email: mn@brics.dk
[2] Email: krukow@brics.dk

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

ally, security sensitive decision making. Many features of GUC (virtual anonymity, scalability, mobility, autonomy, ubiquity, incomplete information, global connectivity, ...) will affect our notion of security requirements. For example, mobility implies that a GUC entity might find itself in a hostile environment, disconnected from its preferred security infrastructure, e.g., its usual certification authorities. Further, the autonomy requirement means that even in this scenario, it must be able to assign privileges to other GUC entities; privileges that are meaningful based on usually incomplete information the assigning entity has about the assigned entity. These properties of GUC imply that traditional security mechanisms are no longer applicable (see e.g., Blaze, Feigenbaum *et al* [1]). One of the alternatives currently being investigated is an approach based on the notion of trust that, in some ways, resembles the concept of trust as it exists among human beings. We refer to this line of research as *computational trust*. In fact, computational trust deals not merely with access control, but more generally with decision making by computational agents in the presence of unknown, uncontrollable and possibly harmful entities. This is the case for e.g. the autonomous selection of (apparently similar) providers of particular services.

In the area of computational trust it is hard to identify one model (or even a few) accepted widely by the community. The GUC feature of incomplete information naturally leads to probabilistic decision making; hence, one common classification distinguishes between 'probabilistic' and 'non-probabilistic' models [8,21,13]. The non-probabilistic systems may be further classified into different types (e.g., social networks or cognitive); in contrast, the probabilistic systems usually have a common objective and structure: they *(i)* assume a particular (probabilistic) model for principal behaviour; and *(ii)* put forward algorithms for approximating the behaviour of principals (i.e., for making predictions in the model). In such models the trust information about a principal is information about its past behaviour, its history. Such histories do not immediately classify principals as 'trustworthy' or 'untrustworthy,' as 'good' or 'bad;' rather, they are used to estimate the probability of potential outcomes arising in a next interaction with an entity. Probabilistic systems, called 'game-theoretical' by Sabater and Sierra [21], are based on Gambetta's view of trust [9]: *"... 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."*

The contribution of this paper is inspired by such a *predictive* view of trust. Any probabilistic model is predicated on an underlying model of interaction amongst computing entities, and for this purpose we use the event structures of Plotkin *et al* [18], as previously argued for in [17] and [14]. We equip event structures with probabilities as in Varacca *et al* [23], and we follow the Bayesian approach to probability theory as advocated in e.g. [10]. In fact, we develop a probabilistic extension of

the event structure framework we previously used in the SECURE project [4], and we use it to model outcomes of interactions and make predictions using Bayesian learning on their configurations. In this sense, our framework generalises previous probabilistic models with only 'binary' outcomes [16,8,22] – i.e., where each interaction is perceived as either 'good' or 'bad' – to multiple, structured outcomes. Such outcomes may in simple cases represent different degrees of satisfaction on the 'good'–'bad' scale, or in more complex cases exploit the full expressive power of event structures' causation mechanisms.

Bayesian analysis consists of formulating hypotheses on some real-world phenomenon, running experiments to test such hypothesis, and thereafter updating the hypotheses –if necessary– to provide a better explanation of the experimental observations, a better fit of the hypotheses to the observed behaviours. By formulating it in terms of conditional probabilities on the space of interest, this procedure is expressed succinctly in formulae by Bayes' Theorem:

$$Prob(\boldsymbol{\Theta} \mid \boldsymbol{X}) \propto Prob(\boldsymbol{X} \mid \boldsymbol{\Theta}) \cdot Prob(\boldsymbol{\Theta}).$$

Reading from left to right, the formula is interpreted as saying: the probability of the hypotheses $\boldsymbol{\Theta}$ *posterior* to the outcome of experiment $\boldsymbol{X}$ is *proportional* to the *likelihood* of such outcome under the hypotheses multiplied by the probability of the hypotheses *prior* to the experiment.[4] In the present context, the prior $\boldsymbol{\Theta}$ will be an estimate of the probability of each potential outcome in our next interaction with principal p, whilst the posterior will be our amended estimate after one such interaction took place with outcome $\boldsymbol{X}$.

It is important to observe here that $Prob(\boldsymbol{\Theta} \mid \boldsymbol{X})$ is in a sense a second order notion, and we are not interested in computing it for any particular value of $\boldsymbol{\Theta}$. Indeed, as $\boldsymbol{\Theta}$ is the unknown in our problem, we are interested in deriving the entire distribution in order to compute its expected value, and use it as our next estimate for $\boldsymbol{\Theta}$.

In order to make this discussion more concrete, let us first focus on a model of binary outcomes. Here $\boldsymbol{\Theta}$ can be represented by a single probability Θ_p, the probability that principal p will behave benevolently, i.e., that an interaction with p will be successful. In this case, a sequence of n experiments $\boldsymbol{X} = X_1 \cdots X_n$ is a sequence of binomial (Bernoulli) trials, and is modelled by a binomial distribution

$$Prob(\boldsymbol{X} \text{ consists of } k \text{ successes}) = \Theta_p^k (1 - \Theta_p)^{n-k}.$$

It turns out that if the prior $\boldsymbol{\Theta}$ follows a $\boldsymbol{\beta}$-distribution, say $B(\alpha, \beta) \propto \Theta_p^{\alpha-1}(1-\Theta_p)^{\beta-1}$ of parameters α and β, then so does the posterior: viz., if $\boldsymbol{X}$ is an n-sequence of k successes, $Prob(\boldsymbol{\Theta} \mid \boldsymbol{X})$ is $B(\alpha + k, \beta + n - k)$, the $\boldsymbol{\beta}$-distribution of parameters $\alpha + k$ and $\beta + n - k$. This is a particularly happy circumstance when it comes to

[4] We shall often omit the proportionality factor, as that is uniquely determined as the constant that makes the right-hand side term a probability distribution. In fact, it equals $Prob(\boldsymbol{X})^{-1}$.

apply Bayes' Theorem, because it makes it straightforward to compute the posterior distribution and its expected value from the prior and the observations; it is known in the literature as the condition that the $\boldsymbol{\beta}$-distribution family is a *conjugate prior* for the binomial trials.

As described above, our model of choice departs from the view of interaction as a sequence of events with binary (success/failure) outcomes. Technically, we see configurations of finite, confusion-free event structures as arising from sequences of *independent, multiple* probabilistic choices. Mathematically, this entails passing from the binomial distributions typical of binomial trials to multinomial distributions $\Theta_1^{n_1} \ldots \Theta_k^{n_k}$ (with $\sum \Theta_i = 1$) typical of n-sequences of trials ($n = \sum n_i$) with k distinct outcomes. In this new framework, our Bayesian analysis relies on observing sequences of event structure configurations –one event at the time– to 'learn' (i.e., estimate) the probability of each configuration occurring as the outcome of the next complex (sequence of elementary) interactions. Here of course Θ_i represents our current estimation of the probability that the ith event in the k-way choice. Correspondingly, we need to identify a suitable *conjugate prior* to multinomial trials, to replace the $\boldsymbol{\beta}$ distribution in the application of Bayes' Theorem. As we explain in §4, we identify it in the family of *Dirichlet* distribution

$$D(\alpha_1, \ldots, \alpha_k) \propto \Theta_1^{\alpha_1 - 1} \cdots \Theta_k^{\alpha_k - 1}.$$

In complete analogy with the binary case, and thus determining a smooth and uniform lifting of the theory, if the prior follows a Dirichlet distribution $D(\alpha_1, \ldots, \alpha_k)$, then the posterior $Prob(\boldsymbol{\Theta} \mid \boldsymbol{X})$ follows the Dirichlet distribution

$$D(\alpha_1 + \#_1(\boldsymbol{X}), \ldots, \alpha_k + \#_k(\boldsymbol{X})),$$

where $\#_i(\boldsymbol{X})$ counts the occurrences of event i in the sequence $\boldsymbol{X}$. We remark that a similar observation was independently made in [11,20].

Our second contribution in this paper is the definition of a formal measure expressing the quality of probabilistic computational trust systems in various application environments. The measure is based on the so-called Kullback-Leibler divergence [15], also known as *information divergence* or *relative entropy*, used in the information theory literature to measure the 'distance' from an approximation to a known target probability distribution. Here we shall adapt it to measure how well an computational trust algorithm approximates the 'true' probabilistic behaviours of computing entities and, therefore, to provide a formal benchmark for the comparison of such algorithms. As an illustration of the applicability of the theory, we present theoretical results within the field, regarding a whole class of existing probabilistic trust algorithms. To our knowledge, no such approach has been proposed previously (but cf. [6] for an application of similar concepts to anonymity). Indeed, we consider this the main result of the paper, in that it presents the first formal results ever in way of comparison of computational trust algorithms.

Structure of the paper. The paper is organised as follows. In §2 we make precise the scenario illustrated somehow informally in the Introduction, and prove our results on the formal of computational trust algorithms. For simplicity, we present our arguments in the case where experiments are sequences of unstructured outcomes; indeed, we expect all of them to go through *mutatis mutandis* to the case where outcomes are event structure configurations. The rest of the paper is dedicated to lifting the binary model to structured, distributed, complex outcomes afforded by event structures. In §3 we introduce the model of probabilistic event structures; readers acquainted with [23] may safely omit this section. In §4 we equip event structures with Dirichlet distributions, and illustrate our event-based framework for Bayesian analysis. Finally, §5 reflects on some of the basic hypotheses of the probabilistic models illustrated in the paper, and points forward to future research aimed at relaxing them.

2 Bayesian models for trust

At the outset, Bayesian trust models are based on the assumption that principals behave in a way that can profitably be approximated by fixed probabilities. Accordingly, while interacting with principal p one will constantly experience outcomes as following an immutable probability distribution $\boldsymbol{\Theta}_p$. Such assumption may of course be unrealistic in several real-world scenarios, and we shall discuss in §5 a research programme aimed to lift it; for the moment however, we proceed to explore where such an assumption leads us.

Our overall goal is to obtain an estimate of $\boldsymbol{\Theta}_p$ in order to inform our future policy of interaction with p. Computational trust algorithms attempt to do this using Bayesian analysis on the history of past interactions with p. Let us fix a probabilistic model of principal behaviour, that is a set of basic assumptions on the way principals behave, say $\boldsymbol{\lambda}$, and then consider the behaviour of a single, fixed principal p. We shall focus on algorithms for the following problem: let $\boldsymbol{X}$ be an interaction history $x_1, x_2, \ldots, x_n$ obtained by interacting n times with p and observing in sequence outcomes x_i out of a set $\{y_1, \ldots, y_k\}$ of possible outcomes. A probabilistic computational trust algorithm, say $\mathcal{A}$, outputs on input $\boldsymbol{X}$ a probability distribution on the outcomes $\{y_1, \ldots, y_k\}$. That is, $\mathcal{A}$ satisfies:

$$\mathcal{A}(y_i \mid \boldsymbol{X}) \in [0,1] \quad {\scriptstyle (i=1,\ldots,k)} \qquad\qquad \sum_{i=1}^{k} \mathcal{A}(y_i \mid \boldsymbol{X}) = 1.$$

Such distribution is meant to to approximate a $\boldsymbol{\Theta}_p$ under the hypotheses $\boldsymbol{\lambda}$. To make this precise, let us assume that the probabilistic model $\boldsymbol{\lambda}$, defines the following

probabilities:

$Prob(y_i \mid \boldsymbol{X}\,\boldsymbol{\lambda})$: the probability of "observing y_i in the next interaction, given the past history $\boldsymbol{X}$;"

$Prob(\boldsymbol{X} \mid \boldsymbol{\lambda})$: the *a priori* probability of "observing $\boldsymbol{X}$ in the model $\boldsymbol{\lambda}$."

Now, $Prob(\cdot \mid \boldsymbol{X}\,\boldsymbol{\lambda})$ defines the 'true' distribution on outcomes for the next interaction (according to the model); in contrast, $\mathcal{A}(\cdot \mid \boldsymbol{X})$ aims at approximating it. We shall now propose a generic measure to 'score' specific algorithms $\mathcal{A}$ against given probability distributions. The score, based on the so-called Kullback-Leibler divergence, is a measure of how well the algorithm approximates the 'true' probabilistic behaviour of principals.

2.1 Towards Comparing Probabilistic Trust-based Systems

Closely related to Shannon's notion of entropy, Kullback and Leibler's information divergence [15] is a measure of the distance between two probability distributions. For $p = (p_1, \ldots, p_k)$ and $q = (q_1, q_2, \ldots, q_m)$ distributions on a finite set of events, the Kullback-Leibler divergence from p to q is defined by

$$\mathrm{D}_{\mathrm{KL}}(p \parallel q) = \sum_{i=1}^{k} p_i \log_2(p_i/q_i),$$

where the log-base used is immaterial. Information divergence resembles a distance in the mathematical sense: it can be proved that D_{KL} satisfies $\mathrm{D}_{\mathrm{KL}}(p \parallel q) \geq 0$ and that equality is obtained if and only if $p = q$; however, it fails to be symmetric. We adapt D_{KL} to score the distance between algorithms by taking the its average over possible input sequences, as illustrated below.

For each $n \in \mathbb{N}$, let $\mathbf{O}^n$ denote the set of interaction histories of length n. Define $\mathrm{D}^n_{\mathrm{KL}}$, the nth *expected Kullback-Leibler divergence* from $\boldsymbol{\lambda}$ to $\mathcal{A}$ as:

$$\mathrm{D}^n_{\mathrm{KL}}(\boldsymbol{\lambda} \parallel \mathcal{A}) = \sum_{\boldsymbol{X} \in \mathbf{O}^n} Prob(\boldsymbol{X} \mid \boldsymbol{\lambda}) \cdot \mathrm{D}_{\mathrm{KL}}(Prob(\cdot \mid \boldsymbol{X}\,\boldsymbol{\lambda}) \parallel \mathcal{A}(\cdot \mid \boldsymbol{X})),$$

That is,

$$\mathrm{D}^n_{\mathrm{KL}}(\boldsymbol{\lambda} \parallel \mathcal{A}) = \sum_{\boldsymbol{X} \in \mathbf{O}^n} Prob(\boldsymbol{X} \mid \boldsymbol{\lambda}) \cdot \sum_{i=1}^{k} Prob(y_i \mid \boldsymbol{X}\lambda) \log_2\left(\frac{P(y_i \mid \boldsymbol{X}\,\boldsymbol{\lambda})}{\mathcal{A}(y_i \mid \boldsymbol{X})}\right).$$

Note that, for each possible input sequence $\boldsymbol{X} \in \boldsymbol{O}^n$, we evaluate the algorithm's performance as $\mathrm{D}_{\mathrm{KL}}(Prob(\cdot \mid \boldsymbol{X}\,\boldsymbol{\lambda}) \parallel \mathcal{A}(\cdot \mid \boldsymbol{X}))$, i.e. we accept that some algorithms may perform poorly on very unlikely training sequences $\boldsymbol{X}$, whilst providing excellent results frequent inputs. Hence, we weigh the performance on each input $\boldsymbol{X}$ by

the intrinsic probability of sequence $\boldsymbol{X}$. In other terms, we compute the *expected* information divergence for inputs of size n.

While Kullback and Leibler's information divergence is a well-established measure in statistics, to our knowledge measuring probabilistic algorithms via $\mathrm{D}^n_{\mathrm{KL}}$ is new. Due to the relation to Shannon's information theory, one can interpret $\mathrm{D}^n_{\mathrm{KL}}(\boldsymbol{\lambda} \parallel \mathcal{A})$ quantitatively as the expected number of *bits of information* one would gain by knowing the 'true' distribution $\mathit{Prob}(\cdot \mid \boldsymbol{X}\, \boldsymbol{\lambda})$ on all training sequences of length n, rather than its approximation $\mathcal{A}(\cdot \mid \boldsymbol{X})$.

2.1.1 An example.

In order to exemplify our measure, we compare the $\boldsymbol{\beta}$-based algorithm of Mui *et al* [16] with the maximum-likelihood algorithm of Aberer and Despotovic [7]. The comparison is possible as the algorithms share the same fundamental assumptions that:

> each principal's behaviour is so that there is a fixed parameter Θ that at each interaction we have, *independently of anything we know about other interactions*, probability Θ of 'success' and, therefore, probability $1 - \Theta$ of 'failure.'

We refer to these as the $\boldsymbol{\beta}$*-model* $\boldsymbol{\lambda}_{\mathbf{B}}$. With s and f standing respectively for 'success' and 'failure,' an n-fold experiment is a sequence $\boldsymbol{X} \in \{s, f\}^n$, for some $n > 0$. The likelihood of $\boldsymbol{X} \in \{s, f\}^n$ is given by

$$\mathit{Prob}(\boldsymbol{X} \mid \Theta\, \boldsymbol{\lambda}_{\mathbf{B}}) = \Theta^{\#_s(\boldsymbol{X})}(1 - \Theta)^{\#_f(\boldsymbol{X})},$$

where $\#_x(\boldsymbol{X})$ denotes the number of occurrences x in $\boldsymbol{X}$. Using $\mathcal{A}$ and $\mathcal{B}$ to denote respectively the algorithm of Mui *et al*, and of Aberer and Despotovic, we have that:

$$\mathcal{A}(s \mid \boldsymbol{X}) = \frac{\#_s(\boldsymbol{X}) + 1}{n + 2} \qquad \text{and} \qquad \mathcal{A}(f \mid \boldsymbol{X}) = \frac{\#_f(\boldsymbol{X}) + 1}{n + 2},$$

$$\mathcal{B}(s \mid \boldsymbol{X}) = \frac{\#_s(\boldsymbol{X})}{n} \qquad \text{and} \qquad \mathcal{B}(f \mid \boldsymbol{X}) = \frac{\#_f(\boldsymbol{X})}{n}.$$

For each choice of $\Theta \in [0, 1]$ and each choice of training-sequence length n, we can compare the two algorithms by computing and comparing $\mathrm{D}^n_{\mathrm{KL}}(\Theta\, \boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A})$ and $\mathrm{D}^n_{\mathrm{KL}}(\Theta\, \boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{B})$.

Theorem 2.1 *If $\Theta = 0$ or $\Theta = 1$, Aberer and Despotovic's algorithm $\mathcal{B}$ from [7] computes a better approximation of the principal's behaviour than Mui et al's algorithm $\mathcal{A}$ from [16]. In fact, under the assumptions, $\mathcal{B}$ always computes the exact probability of success on any possible training sequence.*

Proof. Assume that $\Theta = 0$, and let $n > 0$. The only n-sequence with non-zero probability is f^n, and we have $\mathcal{B}(f \mid f^n) = 1$; in contrast, $\mathcal{A}(f \mid f^n) = (n+1)/(n+2)$,

while $\mathcal{A}(s \mid f^n) = 1/(n+2)$). Since $Prob(s \mid f^n \Theta \boldsymbol{\lambda_B}) = \Theta = 0 = \mathcal{B}(s \mid f^n)$ and $Prob(f \mid f^n \Theta \boldsymbol{\lambda_B}) = 1 - \Theta = 1 = \mathcal{B}(f \mid f^n)$, we can conclude that

$$\mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{B}) = 0.$$

Since $\mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{A}) > 0$ we are done. (The argument for $\Theta = 1$ is similar). □

Let us now compare $\mathcal{A}$ and $\mathcal{B}$ for $0 < \Theta < 1$. Observe that $\mathcal{B}$ assigns probability 0 to s on input f^k for all $k \geq 1$; this results in $\mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{B}) = \infty$. It follows necessarily that in this case $\mathcal{A}$ provides a better approximation.

In order to explore the space of $\boldsymbol{\beta}$-based algorithms further, we define a parametric algorithm $\mathcal{A}_\epsilon$, for $\epsilon \geq 0$, that encompasses both $\mathcal{A}$ and $\mathcal{B}$:

$$\mathcal{A}_\epsilon(s \mid h) = \frac{\#_s(h) + \epsilon}{|h| + 2\epsilon} \qquad \text{and} \qquad \mathcal{A}_\epsilon(s \mid \boldsymbol{X}) = \frac{\#_f(h) + \epsilon}{|h| + 2\epsilon}.$$

Observe that $\mathcal{A}_0 = \mathcal{B}$ and $\mathcal{A}_1 = \mathcal{A}$.

Let us now study the expression $\mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{A}_\epsilon)$ as a function of ϵ. We shall prove that for each $\Theta \neq 1/2$ and independently of n there is a unique $\overline{\epsilon}$ which minimises the distance $\mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{A}_\epsilon)$. Furthermore, $\mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{A}_\epsilon)$ is decreasing on the interval $(0, \overline{\epsilon}]$ and increasing on the interval $[\overline{\epsilon}, \infty)$. (Notice of course that $\mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{A}_\epsilon) \to \infty$ when $\epsilon \to 0$.) By definition, we have:

$$\mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{A}_\epsilon) = \sum_{i=0}^{n} \binom{n}{i} \Theta^i (1-\Theta)^{n-i} \left[\Theta \log \frac{\Theta(n+2\epsilon)}{i+\epsilon} + (1-\Theta) \log \frac{(1-\Theta)(n+2\epsilon)}{n-i+\epsilon} \right].$$

Isolating the terms that contain ϵ, we obtain

$$\sum_{i=0}^{n} \binom{n}{i} \Theta^i (1-\Theta)^{n-i} \Big[\Theta \log \Theta + (1-\Theta) \log(1-\Theta) \Big] + \log(n+2\epsilon)$$
$$- \sum_{i=0}^{n} \binom{n}{i} \Theta^i (1-\Theta)^{n-i} \Big[\Theta \log(i+\epsilon) + (1-\Theta) \log(n-i+\epsilon) \Big].$$

By differentiating $\mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{A}_\epsilon)$ with respect to epsilon, we obtain

$$\frac{d}{d\epsilon} \mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{A}_\epsilon) = \frac{2\alpha}{n+2\epsilon} - \sum_{i=0}^{n} \binom{n}{i} \Theta^i (1-\Theta)^{n-i} \left[\frac{\Theta\alpha}{i+\epsilon} + \frac{(1-\Theta)\alpha}{n-i+\epsilon} \right],$$

where $\alpha = \log e$ is a positive constant obtained when differentiating the function log. In order to find a minimal point for the information diverge, let us examine which ϵ nullify the derivative $d\,\mathrm{D}^n_{\mathrm{KL}}(\Theta \boldsymbol{\lambda_B} \parallel \mathcal{A}_\epsilon)/d\epsilon$. The bulk of the calculation is illustrated in Fig. 1. Observe that since $\sum_{i=0}^{n} i \binom{n}{i} \Theta^i (1-\Theta)^{n-i}$ is the expected number of successes in a Bernoulli trial of length n, it equals Θn. Similarly, one can show

$$
\begin{aligned}
&\frac{d}{d\epsilon}\mathrm{D}^{n}_{\mathrm{KL}}(\Theta\,\boldsymbol{\lambda_{\mathbf{B}}}\parallel\mathcal{A}_{\epsilon}) \quad = \quad 0\\
&\Updownarrow\\
&\sum_{i=0}^{n}\binom{n}{i}\Theta^{i}(1-\Theta)^{n-i}\left[\frac{1}{n/2+\epsilon}-\frac{\Theta}{i+\epsilon}-\frac{(1-\Theta)}{n-i+\epsilon}\right] \quad = \quad 0\\
&\Updownarrow\\
&\sum_{i=0}^{n}\binom{n}{i}\Theta^{i}(1-\Theta)^{n-i}\Big[(i+\epsilon)(n-i+\epsilon)-\Theta(n/2+\epsilon)(n-i+\epsilon)\\
&\qquad\qquad -\ (1-\Theta)(n/2+\epsilon)(i+\epsilon)\Big] \quad = \quad 0\\
&\Updownarrow\\
&\epsilon^{2}\sum_{i=0}^{n}\binom{n}{i}\Theta^{i}(1-\Theta)^{n-i}\Big[1-\Theta-(1-\Theta)\Big] \quad +\\
&\epsilon\sum_{i=0}^{n}\binom{n}{i}\Theta^{i}(1-\Theta)^{n-i}\Big[i+n-i-\Theta(3n/2-i)-(1-\Theta)(n/2+i)\Big] \quad +\\
&\sum_{i=0}^{n}\binom{n}{i}\Theta^{i}(1-\Theta)^{n-i}\Big[i(n-i)-\theta(n^{2}/2-ni/2)-ni(1-\Theta)/2\Big] \;=\; 0\\
&\Updownarrow\\
&\epsilon\sum_{i=0}^{n}\binom{n}{i}\Theta^{i}(1-\Theta)^{n-i}\,[(2\Theta-1)(i-n/2)] \quad +\\
&\qquad\qquad\sum_{i=0}^{n}\binom{n}{i}\Theta^{i}(1-\Theta)^{n-i}\,[(\Theta n-i)(i-n/2)] \quad = \quad 0\\
&\Updownarrow\\
&\epsilon\sum_{i=0}^{n}\binom{n}{i}\theta^{i}(1-\theta)^{n-i}\left[(2\theta-1)(i-\frac{n}{2})\right]=\sum_{i=0}^{n}\binom{n}{i}\theta^{i}(1-\theta)^{n-i}\left[(i-\theta n)(i-\frac{n}{2})\right] \quad (1)
\end{aligned}
$$

Fig. 1. Solving the equation $d\,\mathrm{D}^{n}_{\mathrm{KL}}(\Theta\,\boldsymbol{\lambda_{\mathbf{B}}}\parallel\mathcal{A}_{\epsilon})/d\epsilon=0$.

that

$$\sum_{i=0}^{n} i^{2}\binom{n}{i}\Theta^{i}(1-\Theta)^{n-i}=n(n-1)\Theta^{2}+\Theta n.$$

These equalities lets us write equation (1) in a simpler form:

$$\epsilon(2\Theta-1)(n\Theta-n/2)=n(n-1)\Theta^{2}+\Theta n-\Theta n(n/2+\Theta n)+\Theta n^{2}/2.$$

We therefore have $\epsilon(2\Theta-1)(n\Theta-n/2) = \epsilon(2\Theta-1)n(\Theta-1) = \epsilon(2\Theta-1)^2\, n/2$, and

$$\begin{aligned} n(n-1)\Theta^2 + \Theta n - \Theta n(n/2+\Theta n) + \Theta n^2/2 &= \\ n^2(\Theta^2 - \Theta/2 - \Theta^2 + \Theta/2) + n(\Theta - \Theta^2) &= n\Theta(1-\Theta). \end{aligned}$$

Since $(2\Theta-1)^2$ is non-zero when $\Theta \neq 1/2$, we obtain that $d\,\mathrm{D}^n_{\mathrm{KL}}(\Theta\,\boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A}_\epsilon)/d\epsilon$ is nullified if and only if $\Theta \neq 1/2$ and

$$\epsilon = \frac{2\Theta(1-\Theta)}{(2\Theta-1)^2}.$$

Remarkably, this is independent of n. Also, from the same derivation we immediately obtain that

$$\frac{d}{d\epsilon}\mathrm{D}^n_{\mathrm{KL}}(\Theta\,\boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A}_\epsilon) < 0 \quad \Longleftrightarrow \quad \epsilon < \frac{2\Theta(1-\Theta)}{(2\Theta-1)^2}$$

and

$$\frac{d}{d\epsilon}\mathrm{D}^n_{\mathrm{KL}}(\Theta\,\boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A}_\epsilon) > 0 \quad \Longleftrightarrow \quad \epsilon > \frac{2\Theta(1-\Theta)}{(2\Theta-1)^2}$$

We have therefore proved the following.

Theorem 2.2 *For any $\Theta \in [0,1/2) \cup (1/2,1]$ there exists $\overline{\epsilon} \in [0,\infty)$ that minimises $\mathrm{D}^n_{\mathrm{KL}}(\Theta\,\boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A}_\epsilon)$ simultaneously for all n; viz., $\overline{\epsilon} = 2\Theta(1-\Theta)/(2\Theta-1)^2$.*
Furthermore, $\mathrm{D}^n_{\mathrm{KL}}(\Theta\,\boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A}_\epsilon)$ is a decreasing function of ϵ in the interval $(0,\overline{\epsilon}\,)$ and increasing in $(\,\overline{\epsilon},\infty)$.

This means that unless the principal's behaviour is completely unbiased, then there exists a unique best $\mathcal{A}_{\overline{\epsilon}}$ algorithm that outperforms all the others, for all n. If instead $\Theta = 1/2$, then the larger the ϵ, the better the algorithm. Regarding $\mathcal{A}$ and $\mathcal{B}$, an application of Theorem 2.2 tells us that the former is optimal for $\Theta = 1/2 \pm 1/\sqrt{12}$, whilst –as anticipated by Theorem 2.1– the latter is such for $\Theta = 0$ and $\Theta = 1$.

Concluding this section, it is useful to remark that it is not so much the comparison of algorithms $\mathcal{A}$ and $\mathcal{B}$ that interests us; rather, the message is that using formal probabilistic models enables such mathematical comparisons and, more in general, to investigate properties of models and algorithms.

3 Probabilistic Event Structures

Agents in a distributed system obtain information by effecting behavioural observations, typically triggered by exchanging messages. The structure of such message exchanges is usually given in the form of protocols known to both parties before

the interaction begins. By behavioural 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, and more. Here as in previous work (cf. [17,14,13]) we use *event structures* to formalise the concepts of protocols, observations, and outcomes.

Event structures are well suited to our present purposes, as they provide a *generic* model for events (i.e., basic observables) and causation that is independent of any specific programming language and higher-level model. In our model, the information that an agent holds about another agent's behaviour, is information about a number of protocol-runs with it, organised as a sequence of *sets of events*, or *configurations*, $x_1 x_2 \cdots x_n$. Configuration x_i represents the ith run of the protocol (e.g., ordered chronologically by starting times), and collects all the events happened up to that point in that instance of the protocol; x_i may represent a completed protocol-run, in which case it records the complete outcome of an interaction, or an running one, in which case more events will be added to it as the computation proceeds. Note that, as opposed to many existing systems, here we are not *rating* the behaviour of principals; we are instead *recording* their actual behaviour, i.e., the precise events occurred in the interaction. We will later equip the model with probability measures so as to rate interaction outcomes and therefore assess the likelihood of future interactions.

Although event structures were the model of choice for computing 'trust values' of distributed interactions in the SECURE project [3,4], we did not use in that context a formal probabilistic model of principal behaviour. In the next two sections, we amend that: we augment event structure framework with a probabilistic model which generalises the one used in systems based on the beta-distribution [12,16,2,22], and we show how to compute the probabilities of outcomes given a history of observations. While this could be valuable in its own right, we remark that our primary reason is to illustrate an example of a formal probabilistic model which enables formal questions to be asked (and answered). The proposed system is not yet practical: there are in fact many issues it does not handle, as e.g., changes of principal behaviours, lying reputation sources, and multiple execution contexts. We believe that the basic probabilistic model must be better understood before we can deal with such issues successfully.

3.1 Event structures

We briefly recaptulate the basic definitions, whilst referring the reader to [17,14,13] for more details and examples. An *event structure* is a triple $ES = (E, \leq, \#)$ consisting of a set E of *events* which are partially ordered by $\leq$, the *necessity* (or *causality*) *relation*; the *conflict relation* $\#$ is a binary, symmetric, irreflexive relation on events.

They satisfy the following properties for all $e, e', e'' \in E$.

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

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

The intention behind all this should be intuitive. An event may *exclude* the possibility of the occurrence of other events; this is what the conflict relation models. The necessity relation represents the idea that events are *only possible* when others, their causes, have already occurred. Finally, if two events are in neither of the relations, they are said to be *independent*. The two conditions above are therefore that events must be finitely-caused and that conflict extents along with causation.

An event structure models the set of events that can occur in a particular protocol; the control flow is provided by $\leq$ and $\#$, that guarantee that not all sets of events can occur in a particular run. The notion of configurations formalises this as follows. A set of events $x \subseteq E$ is a *configuration* of ES if it is

$$\textit{Conflict free:} \text{ for any } e, e' \in x : \text{ not } e \# e'; \quad \text{and}$$

$$\textit{Causally closed:} \text{ for any } e \in x, e' \in E : e' \leq e \text{ implies } e' \in x.$$

We write $\mathcal{C}_{ES}$ for the set of configurations of ES. The set of all maximal configurations, i.e., configuration that cannot be extended, defines the set outcomes of an interaction exhaustively. Such configurations are of course mutually exclusive.

3.2 Histories

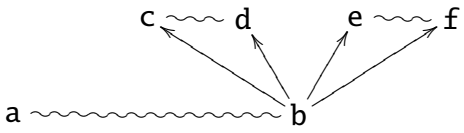
A finite configuration models information regarding *a single* interaction, i.e., a single run of a protocol. In general, the information that one principal possesses about another consists of information about *several* protocol runs; the information about each individual run being represented by a 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 are called *sessions* of h.

Remarks. While the order of sessions is recorded (that is, histories are *sequences*), in contrast, the order of *independent* events within *a single session* is not. Independence of events is in fact 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). This is of course a feature and not a limitation of event structures: in a scenario where ordering events is both relevant and observable, one can always use a suitably 'serialised' event structure to record it.

3.3 Confusion-Free Event Structures

In the following we consider a special type of event structures, so-called *confusion free*, to which it is especially simple to adjoin probabilities [23]. As we shall see, the key for that is to assure that all 'choice points' –here called cells– are independent of each other. This amounts to requiring that the occurrence of an event does not affect the relative probabilities inside cells, even though it may of course rule out entire cells. This will be achieved by guaranteeing that each event belongs to at most one cell and that conflict behaves uniformly on cells.

Consider the following event structure as an aid to fix ideas in the following definitions ($\sim$ represents conflict, and $\rightarrow$ represents causality).



Events c and e are *independent*, as are the following pairs: c and f; d and e; and d and f. In event structures, this simply means that both events in independent pairs can occur in any order in the same configuration. We aim at defining a probabilistic model where independence also means *probabilistic* independence. To such end we present the concepts of *cell* and *immediate conflict* [23].

Let $ES = (E, \leq, \#)$ be a fixed event structure. Write $[e)$ for $[e] \setminus \{e\}$, and say that events $e, e' \in E$ are in *immediate conflict*, in symbol $e \#_\mu e'$, if

$$e \# e' \quad \text{and} \quad \text{both } [e) \cup [e'] \text{ and } [e] \cup [e') \text{ are configurations.}$$

Clearly, a conflict $e \# e'$ is immediate if-and-only-if there exists a configuration x where both e and e' are enabled. This means that they can occur at the same time 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*. This entails that in order to complete from $[e)$, the computation will have to 'pick' exactly one event from the cell. Cells represent choices. There are three cells in the above event structure: $\{a, b\}, \{c, d\}$ and $\{e, f\}$.

A *confusion free* event structure is an event structure where immediate conflict is a transitive relation and is within cells, i.e., $e \#_\mu e'$ implies $[e) = [e')$. In confusion-free event structures, if an event of a cell c is enabled at configuration x, then all events of c are enabled at x. This is because if e is in conflict with some $e' \in c$, then e is in conflict with all $e' \in c$. 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, as the current configuration is maximal; otherwise, non-deterministically select a cell

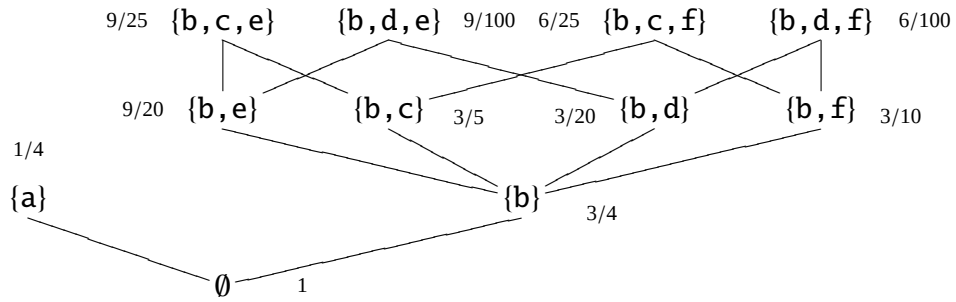


Fig. 2. An example of cell valuation and the probabilities of configurations

$c \in C$, and then non-deterministically select (or probabilistically sample) an event $e \in c$. Update the current configuration by adding e.

The concept of cell-valuation formalises probabilistic sampling in cells. Here and in the following, for $f : X \to [0, +\infty]$ a function and $Y \subseteq X$ a set, we use $f[Y]$ to denote $\sum_{y \in Y} f(y)$.

Definition 3.1 (**Cell valuation**, Varacca *et al* [23]) A *cell valuation* on a confusion-free event structure $ES = (E, \leq, \#)$ is a function $p : E \to [0, 1]$ such that for every cell c, we have $p[c] = 1$.

If we assume (probabilistic) independence between events in cells, then we can compute the probability of any configuration x occurring simply as the product of the probabilities of the constituting events.

Proposition 3.2 ([23]) *Let p be a cell valuation, and write $p(x)$ for $\prod_{e \in x} p(e)$; then*

- $p(\emptyset) = 1$;
- $p(x) \geq p(x')$, *if* $x \subseteq x'$;
- $p(x) = p[C]$, *if C is a maximal set of configurations covering x;*
- *p is a probability distribution on maximal configurations.*

In the formulation above, we say y covers x to mean $y = x \cup \{e\}$ for some e. Observe that the implication of Proposition 3.2 is that $p(x)$ must be interpreted as the probability that the (partial) configuration x is contained in the outcome of the computation. On maximal configurations, p yields a probability distribution. With reference to the event structure of our running example, the assignment

$$\{a \mapsto 1/4;\ \ b \mapsto 3/4;\ \ c \mapsto 4/5;\ \ d \mapsto 1/5;\ \ e \mapsto 3/5;\ \ f \mapsto 2/5\}$$

is easily seen to represent a cell valuation. The entire structure of configurations and their probabilities is given in Fig 2.

4 A Bayesian framework for event-based models

As illustrated in the previous section, finding a cell valuation $p : E \to [0,1]$ is the key step to assign probabilities to the configurations of a finite, confusion-free event structure *ES*. Observe that to give one such p is to give for each cell c a function $p_c : c \to [0,1]$ with $p_c[c] = 1$, i.e., a probability distribution. Our assumption is, typical of the Bayesian approach, that the distributions p_c exists independently and immutably; our intention is, equally typical, for them to be 'learned' via experiments, that in our case means derived from the past history of interactions with an external entity. Under the following heading, we state explicitly the assumptions about the behaviour of entities in our model. We then proceed to *(i)* find abstractions that preserve sufficient information under the model; and *(ii)* derive equations for predictive probabilities, i.e., formulae to answer questions such as "what is the probability of outcome x in the next interaction with entity q?"

4.1 The model

Let *ES* be a finite, confusion-free event structure *ES* and $\mathrm{C}(ES) = \{c_1, c_2, \ldots, c_M\}$ its set of its cells, where $c_i = \{e^i_1, \ldots, e^i_{K_i}\}$. We write $\boldsymbol{\lambda}_{\mathbf{DES}}$ for the following assumptions of our model:

> each principal's behaviour is so that there are fixed parameters Θ_{c_i} such that at each interaction there is, *independently of anything we know about other interactions*, probability *distribution* Θ_{c_i} for the events of cell c_i to occur, if c_i is enabled.

Such basic data are equivalent to give a probability Θ_e to each event e of *ES*, so that $\sum_{k=1}^{K_i} \Theta_{e^i_k} = 1$. The collection $\boldsymbol{\Theta} = (\Theta_{c_1}, \ldots, \Theta_{c_M})$ determines a cell valuation on *ES*. It follows from our assumption that for each configuration $x \in C_{ES}$ the probability of obtaining x in any run of *ES* with a principal parametrised by $\boldsymbol{\Theta}$ is

$$Prob(x \mid \boldsymbol{\Theta}\, \boldsymbol{\lambda}_{\mathbf{DES}}) = \prod_{e \in x} \Theta_e. \tag{2}$$

In way of comparison with model $\boldsymbol{\lambda}_{\mathbf{B}}$ described in §2, $\boldsymbol{\lambda}_{\mathbf{DES}}$ assigns probabilities to (maximal) configurations to the same effect as $\boldsymbol{\lambda}_{\mathbf{B}}$ does for the binary outcomes $\{s, f\}$. In this case however, the assignment is not 'atomic,' but obtained via a cell evaluation, i.e., an assignment of probability distributions to cells and, ultimately, to basic events. While the occurrence of an x from $\{s, f\}$ is a *binomial (Bernoulli) trial*, the occurrence of an event from c_i is a random process with K_i outcomes. That is, a *multinomial* trial on Θ_{c_i}. To exploit this analogy, we therefore only need to lift the framework of §2 to one based on multinomial experiments. In particular, we shall need to identify a family of distributions that can play here the same role as the $\boldsymbol{\beta}$-distribution does there.

Firstly, we observe that in order to estimate the parameters $\boldsymbol{\Theta}$ given a prior distribution, we only need a simple event count, i.e., a function $\boldsymbol{X} : E \to \mathbb{N}$. In fact,

in force of Eq. (2), it is sufficient to estimate the parameters Θ_c for each cell c. It then follows from the assumptions of $\boldsymbol{\lambda}_{\mathbf{DES}}$ that a count $\boldsymbol{X}$ of the event occurrences in h is the only significant information for any sequence $h \in C^*_{ES}$ of data observed about a fixed principal.

Secondly, in order to apply Bayesian analysis, we need *prior* distributions. As we intend to use our estimates to determine expected values for entire distributions, it is fundamental that we are able to compute them in symbolic form. This is one of the roles of conjugate priors. We phrase the following definition with terminology used in the Introduction to illustrate Bayes' Theorem.

Definition 4.1 A family F of probability 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.

Indeed, the use of conjugate priors generally affords a significant computational convenience in Bayesian analysis, in that the distributions always maintain the same algebraic form. As we shall see below, it turns out that the family of *Dirichlet distributions* is a family of conjugate prior distributions for multinomial trials.

The use of Dirichlet distributions as priors completes the picture of our application of Bayes' Theorem to event structures. Specifically, a prior Dirichlet distribution is assigned to each cell c of ES. Event counts $\boldsymbol{X}$ are then used to update the Dirichlet at each cell. Hence, at any time we have for each cell c a Dirichlet distribution on the parameters Θ_c of that cell. We will show that the probability of an outcome $x \subseteq E$ is then the product of certain expectations of these distributions.

4.2 The Dirichlet distribution

The Dirichlet family D of order K, for $2 \leq K \in \mathbb{N}$, is a parametrised collection of continuous probability density functions (pdf) defined on $[0,1]^K$; K parameters of positive reals, $\alpha = (\alpha_1, \ldots, \alpha_K)$, select a specific Dirichlet distribution from the family. For a variable $\Theta = (\Theta_1, \ldots, \Theta_K) \in [0,1]^K$, the pdf is given by:

$$D(\Theta \mid \alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \cdot \prod \Theta_1^{\alpha_1 - 1} \cdots \Theta_K^{\alpha_K - 1},$$

where the Gamma function, $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt$, for $z > 0$, is used to define the normalisation constant. Luckily, we shall not be explicitly concerned with such a constant. The main values of interests in our application are the expected value and the variance of variables distributed according to $D(\cdot \mid \alpha)$ which, importantly, depend only on α. Namely, using $[\alpha]$ as a shorthand for $\sum_j \alpha_j$, we have:

$$\boldsymbol{E}_{D(\Theta|\alpha)}(\Theta_i) = \frac{\alpha_i}{[\alpha]} \qquad \boldsymbol{\sigma}^2_{D(\Theta|\alpha)}(\Theta_i) = \frac{\alpha_i([\alpha] - \alpha_i)}{[\alpha]^2([\alpha] + 1)} \tag{3}$$

4.3 A conjugate prior

Consider sequences of independent experiments with K-ary outcomes, each yielding outcome i with some fixed probability Θ_i; such experiments are *multinomial trials* (and in our framework correspond to the probabilistic choice of one event at a cell). Let $\boldsymbol{\lambda_D}$ denote a model collecting such hypothesis, and let X_i represent the ith trial ($i = 1, \ldots, n$). In other words, $Z_i \equiv (X_i = j_i)$ is the statement that the ith trial has outcome $j_i \in \{1, 2, \ldots, K\}$. Let $\boldsymbol{Z} = (Z_1, \ldots, Z_n)$ be the conjunction of n such statements. Then, by definition of multinomial trials, the sequence of independent experiments has the following likelihood:

$$Prob(\boldsymbol{Z} \mid \Theta\, \boldsymbol{\lambda_D}) = \prod_{i=1}^{n} Prob(Z_i \mid \Theta\, \boldsymbol{\lambda_D}) = \prod_{i=1}^{K} \Theta_i^{\#_i(\boldsymbol{Z})},$$

where $\#_i(\boldsymbol{Z})$ is the number of occurrences of i in $\boldsymbol{Z}$.

The Dirichlet distributions constitute a family of conjugate prior distributions for this likelihood. In order to illustrate this fact, let us recall that according to Bayes' Theorem one can derive from the prior distribution on Θ, say $f(\Theta \mid \boldsymbol{\lambda_D})$, and the experiments $\boldsymbol{Z}$, a posterior distribution $f(\Theta \mid \boldsymbol{Z}\, \boldsymbol{\lambda_D})$ as:

$$f(\Theta \mid \boldsymbol{Z}\, \boldsymbol{\lambda_D}) = f(\Theta \mid \boldsymbol{\lambda_D}) \frac{Prob(\boldsymbol{Z} \mid \Theta\, \boldsymbol{\lambda_D})}{Prob(\boldsymbol{Z} \mid \boldsymbol{\lambda_D})}.$$

In fact, it is not hard to show that when $f(\Theta \mid \boldsymbol{\lambda_D})$ is a Dirichlet distribution, say $D(\Theta \mid \alpha_1, \ldots, \alpha_K)$, then $f(\Theta \mid \boldsymbol{Z}\, \boldsymbol{\lambda_D})$ is a Dirichlet distribution too; viz.,

$$f(\Theta \mid \boldsymbol{Z}\, \boldsymbol{\lambda_D}) = D(\Theta \mid \alpha_1 + \#_1(\boldsymbol{Z}), \ldots, \alpha_K + \#_K(\boldsymbol{Z})),$$

which is what we wanted. Note that by choosing $\alpha_i = 1$ for all i, the Dirichlet distribution degenerates to the uniform distribution on $[0, 1]^K$. This is very useful, as it provides a convenient unbiased initial prior for those cases, relatively frequent in our application domain, where we have no prior information on principals.

4.4 Predictive probability in the Dirichlet model $\boldsymbol{\lambda_D}$

Let us now consider the statement $Z_{n+1} \equiv (X_{n+1} = i)$ before performing the $n + 1$ experiment. We can then can interpret $Prob(Z_{n+1} \mid \boldsymbol{Z}\, \boldsymbol{\lambda_D})$ as a predictive probability: given no direct knowledge of Θ, but only past evidence (viz., $\boldsymbol{Z}$) and the model (viz., $\boldsymbol{\lambda_D}$), then $Prob(Z_{n+1} \mid \boldsymbol{Z}\, \boldsymbol{\lambda_D})$ is the probability that the next trial will result in outcome i. It is easy to show that:

$$Prob(Z_{n+1} \mid \boldsymbol{Z}\, \boldsymbol{\lambda_D}) = \boldsymbol{E}_{f(\Theta \mid \boldsymbol{Z} \boldsymbol{\lambda_D})}(\Theta_i) = \frac{\alpha_i + \#_i(\boldsymbol{Z})}{[\alpha] + n}$$

In fact, the predictive probability that the $n + 1$th outcome is i is obviously the expectation of the ith parameter of the posterior computed after the experiments

$\boldsymbol{Z}$. Then, given the Dirichlet expression above for $f(\Theta \mid \boldsymbol{Z}\,\boldsymbol{\lambda}_{\mathbf{D}})$ and the fact that $\sum_i \#_i(\boldsymbol{Z}) = n$, the results follows from the expectation formula (3). We remark that the variance formula can be used at any time to evaluate the accuracy of our prediction: the lower the variance, the more likely the prediction.

4.5 Dirichlet distributions on cells

Returning to our event structure model, we will associate to each cell $c \in \mathrm{C}(ES)$ a Dirichlet prior distribution on the parameters Θ_c determining the behaviour of a fixed principal for the events of c. As we interact with the principal, we use Bayes' Theorem and the formulae derived above to tighten the parameters to the observation and therefore sharpen our ability to predict outcomes via the predictive probability. Each cell $c \in \mathrm{C}(ES)$ presents a choice between the mutually exclusive and exhaustive events of c, and by the assumptions of $\boldsymbol{\lambda}_{\mathbf{DES}}$ such choices are multinomial trials. At any time, we obtain the predictive probability of the next interaction resulting in a particular configuration by multiplying the expectations of the parameters for each event in the configuration.

Let us be precise. Let $f_c(\Theta_c \mid \boldsymbol{\lambda}_{\mathbf{DES}})$ denote the prior distribution for $c \in \mathrm{C}(ES)$, and let there be a positive real number α_e associated to each $e \in E$. We use α_{c_i} as a shorthand for the vector of parameters associated to c_i, i.e., $(\alpha_{e^i_1}, \ldots, \alpha_{e^i_{K_i}})$. We are then just left with the task of adapting the formulae of §4.3. We have:

$$f_{c_i}(\Theta_{c_i} \mid \boldsymbol{\lambda}_{\mathbf{DES}}) = D(\Theta_{c_i} \mid \alpha_{c_i}) = \frac{\Gamma([\alpha_{c_i}])}{\prod_{k=1}^{K_i} \Gamma(\alpha_{e^i_k})} \cdot \prod_{k=1}^{K_i} \Theta_{e^i_k}^{\alpha_{e^i_k} - 1}\,.$$

Let $\boldsymbol{X} : E \to \mathbb{N}$ be an event count that models the observations about past runs with a specific principal. The posterior pdf is given below, where + denotes both scalar and vector sum, and $\boldsymbol{X}(c_i) = (\boldsymbol{X}(e^i_1), \ldots, \boldsymbol{X}(e^i_{K_i}))$.

$$f_{c_i}(\Theta_{c_i} \mid \boldsymbol{X}\,\boldsymbol{\lambda}_{\mathbf{DES}}) = D(\Theta_{c_i} \mid \alpha_{c_i} + \boldsymbol{X}(c_i)) = \frac{\Gamma([\alpha_{c_i} + \boldsymbol{X}(c_i)])}{\prod_{k=1}^{K_i} \Gamma(\alpha_{e^i_k} + \boldsymbol{X}(e^i_k))} \cdot \prod_{k=1}^{K_i} \Theta_{e^i_k}^{\alpha_{e^i_k} + \boldsymbol{X}(e^i_k) - 1}\,.$$

Such fierce-looking formula is in reality very simple: it just states that each event count $\boldsymbol{X} : E \to \mathbb{N}$ can be used to do Bayesian updating of our cell valuation simply by adding $\boldsymbol{X}(e)$ to α_e, for each $e \in E$.

4.6 Predictive probability in the model $\boldsymbol{\lambda}_{\mathbf{DES}}$

Also in this case, we merely need to adapt the formulae derived in §4.4. Let $\boldsymbol{X}$ be an event count corresponding to the observation of n previous configurations $x_1 \cdots x_n$ in the interaction with a fixed principal. Let Z be the proposition that "the $(n{+}1)$'st interaction results in outcome x." According to the independence hypotheses made

in $\boldsymbol{\lambda}_{\mathbf{DES}}$, the predictive probability $Prob(Z \mid \boldsymbol{X}\, \boldsymbol{\lambda}_{\mathbf{DES}})$ is the product of the probabilities of occurrence of each $e \in x$. But the probability of e^i_j occurring, *provided* its cell c_i is enabled, is exactly the expected value of $\Theta_{e^i_j}$, and since we know the pdfs $f_c(\Theta_c \mid \boldsymbol{X}\, \boldsymbol{\lambda}_{\mathbf{DES}})$ for all $c \in CES$, we can use the expectation formulae (3).

$$\boldsymbol{E}(\Theta_{e^i_j} \mid \boldsymbol{X}\, \boldsymbol{\lambda}_{\mathbf{DES}}) = \frac{\alpha_{e^i_j} + \boldsymbol{X}(e^i_j)}{[\alpha_{c_i} + \boldsymbol{X}(c_i)]}\,.$$

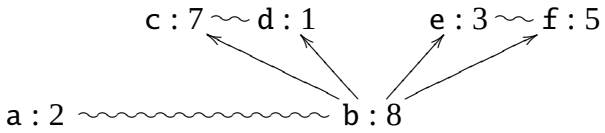
The predictive probability is therefore the product of the expectations of each of the cell parameters.

$$\begin{aligned} Prob(\text{next outcome is } x \mid \boldsymbol{X}\, \boldsymbol{\lambda}_{\mathbf{DES}}) &= Prob(Z \mid \boldsymbol{X}\, \boldsymbol{\lambda}_{\mathbf{DES}}) \\ &= \prod_{e \in x} \boldsymbol{E}(\Theta_e \mid \boldsymbol{X}\, \boldsymbol{\lambda}_{\mathbf{DES}}) = \prod_{e^i_j \in x} \frac{\alpha_{e^i_j} + \boldsymbol{X}(e^i_j)}{[\alpha_{c_i} + \boldsymbol{X}(c_i)]}\,. \end{aligned}$$

4.7 Summary

We have presented a probabilistic model $\boldsymbol{\lambda}_{\mathbf{DES}}$ based on probabilistic confusion-free event structures. The model generalises previous work on probabilistic models using binary outcomes and $\boldsymbol{\beta}$ prior distributions. In our model, given a past history with a principal we need only remember the event counts of the past, i.e., a function $\boldsymbol{X} : E \to \mathbb{N}$. 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.

With reference to the event structure of our running example, suppose we have the following event count $\boldsymbol{X}$.



If we assume to start with uniform priors, i.e., $\alpha_e = 1$ for $e \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$, then $\boldsymbol{X}$ gives rise to the following updated Dirichlet distributions.

$$\begin{aligned} f_{\{\mathtt{a},\mathtt{b}\}}(\Theta_{\mathtt{a}}, \Theta_{\mathtt{b}} \mid \boldsymbol{X}\, \boldsymbol{\lambda}_{\mathbf{DES}}) &= D(\Theta_{\mathtt{a}}, \Theta_{\mathtt{b}} \mid 3, 9), \\ f_{\{\mathtt{c},\mathtt{d}\}}(\Theta_{\mathtt{c}}, \Theta_{\mathtt{d}} \mid \boldsymbol{X}\, \boldsymbol{\lambda}_{\mathbf{DES}}) &= D(\Theta_{\mathtt{c}}, \Theta_{\mathtt{d}} \mid 8, 2), \\ f_{\{\mathtt{e},\mathtt{f}\}}(\Theta_{\mathtt{e}}, \Theta_{\mathtt{f}} \mid \boldsymbol{X}\, \boldsymbol{\lambda}_{\mathbf{DES}}) &= D(\Theta_{\mathtt{e}}, \Theta_{\mathtt{f}} \mid 4, 6). \end{aligned}$$

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

$$Prob(\{b, c\} \mid \boldsymbol{X}\, \boldsymbol{\lambda}_{\mathbf{DES}}) = \frac{9}{12} \times \frac{8}{10} = \frac{3}{5}.$$

In fact, the cell valuation arising from this is the one illustrated in Fig 2.

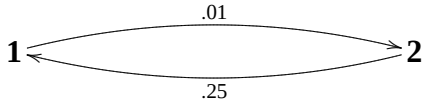
5 Towards a formal model of dynamic behaviour

In this section we reflect on what has been achieved in this paper, but mainly on what has not. Our main motivation when we started this investigation was to put on formal grounds what we had been seeing in the literature, so as to be able to ask sharp questions of our data. We succeeded in this to a comforting extent, both by presenting the first ever formal framework for the comparisons of computational trust algorithms and by extending a well-known formal concurrency model with a framework for Bayesian analysis.

However, 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 β-model of principal behaviour (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, the probability of a 'good' outcome is Θ_p of the one of 'bad' outcome is $1 - \Theta_p$. One might argue that this is unrealistic for some applications. In particular, the model allows for no dynamic behaviour, while in reality not only the p is likely to change its behaviour in time, as its environmental conditions change, but p's behaviour in interactions with q is likely to depend on q's behaviour in interactions with p. The same criticisms apply of course to the Dirichlet model we presented here.

Some beta-based reputation systems attempt to deal with the first problem by introducing so-called 'forgetting factors.' Essentially this amounts to choosing a factor $0 \leq \delta \leq 1$, and then each time the parameters (α, β) of the pdf for Θ_p are updated, they are also scaled by δ. In particular, when observing a single 'good' interaction, (α, β) becomes $(\alpha\delta + 1, \beta\delta)$ rather than (α, β). Effectively, this performs a form of exponential 'decay' on parameters. The idea is that information about old interactions is less relevant than new information, as it is more likely to be outdated. This approach represents a departure from the probabilistic beta model, where all interactions 'weigh' equally, and in the absence of any mathematical it is not clear what the exact benefits of this bias towards newer information is. Regarding the second problem, to our knowledge it has not yet been considered in the literature.

Let us point out some ideas towards refining such hypothesis and embracing the fact that the behaviour of p depends on its internal state, which is likely to change over time. Suppose we model p as a kind of Markov chain, a probabilistic finite-state system with n states $S = \{1, 2, \ldots, n\}$ and n^2 transition probabilities $t_{ij} \in [0, 1]$, with $\sum_{j=1}^{n} t_{ij} = 1$. After each interaction, p changes state according to t: it takes a transition from state i to state j with probability t_{ij}. Such state-changes are likely in our context to be unobservable: a principal q does not know for certain which state principal p is in. All that q can observe, now as before, is the outcome of



$\pi_{\mathbf{1}} = 1$ $\qquad$ $\pi_{\mathbf{2}} = 0$

$B_{\mathbf{1}}(a) = .95$ $\qquad O = \{a, b\}$ $\qquad B_{\mathbf{2}}(a) = .05$

$B_{\mathbf{1}}(b) = .05$ $\qquad\qquad$ $B_{\mathbf{2}}(b) = .95$

Fig. 3. Example Hidden Markov Model.

its interactions with p; based on that, it must make inferences on p's likely state and future actions. If we accept the finite state assumption and the Markovian transition probabilities, we can then incorporate unobservable states in the model by using so-called Hidden Markov Models [19].

A discrete *Hidden Markov Model* (HMM) is a tuple $\lambda = (S, \pi, t, O, s)$ where S is a finite set of *states*; π is a distribution on S, the *initial distribution*; $t : S \times S \to [0, 1]$ is the *transition matrix*, with $\sum_{j \in S} t_{ij} = 1$; finite set O is the set of possible *observations*; and where $s : S \times O \to [0, 1]$, the *signal*, assigns to each state $j \in S$, a distribution s_j on observations, i.e., $\sum_{o \in O} s_j(o) = 1$.

An example. Consider the HMM in Figure 3. This models a simple two-state process with two possible observable outputs a and b. For example, this could model a channel which can forward a packet or drop it. State **1** models the normal mode of operation, whereas state **2** models operation under high load. Suppose that output a means 'packet forwarded' and output b means 'packet dropped.' Most of the time, the channel is in state **1**, and packets are forwarded with probability .95; occasionally the channel will transit to state **2** where packets are dropped with probability .95. Although this example is just meant to illustrate a simple HMM, we expect that by tuning their parameters Hidden Markov Models can provide an interesting model many of the dynamic behaviours needed for probabilistic trust-based systems.

Consider now an observation sequence, $h = a^{10}b^2$ (that is ten a's followed by two b's), which is reasonably probable in our model on Figure 3. The final fragment consisting of two consecutive occurrences of b's makes it likely that a state-change from **1** to **2** has occurred. Nevertheless, a simple counting algorithm, say $\mathcal{H}$, would probably assign high probability to the event that a will happen next:

$$\mathcal{H}(a \mid h) = \frac{\#_a(a^{10}b^2) + 1}{|h| + 2} = 11/14 \sim .80$$

However, if a state-change has indeed occurred, that probability would be as low as .05.

Suppose now exponential decay is used, e.g., as in the Beta reputation system [12], with a factor of $\delta = .5$. This means that the last observation weighs approximately the same as the rest of the history; in such a case, the algorithm would adapt

quickly, and assign probability $\mathcal{H}(a \mid h) \sim .25$, which is a much better estimate. However, suppose that we now observe *bb* and then another *a*. Again this would be reasonably likely in state **2**, and would make a state-change to **1** probable in the model. The exponential forgetting would assign a high weight to *a*, but also a high weight to *b*, because the last four observations were *b*'s. In a sense, perhaps the algorithm adapts 'too quickly,' it is too sensitive to new observations. So, no matter what δ is, it appears easy to describe situations where it does not reach its intended objective; our main point here is the same as for our comparisons of computational trust algorithms in §2: that the underlying assumptions behind a computational idea (e.g., the exponential decay) need to be specified, and that formal models for principal's behaviour (e.g., HMMs) may serve the purpose, allowing precise questions on the applicability of the computational idea.

6 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 the point, we have presented an example of a formal probabilistic model, $\boldsymbol{\lambda}_{\mathbf{DES}}$. The central technical contribution here has been to recast one of the best known and most popular models of concurrency, the event structures, in a framework for Bayesian analysis. This allows 'learning' and 'prediction' of composite, multi-event structured outcomes (viz., event structure configurations) in complex interaction protocols (viz., event structures). We anticipate the model will be useful in several applications, even though in the paper we discussed some of its shortcomings and hinted at future developments.

Among the several benefits of formal probabilistic models, we have focussed on the possibility to compare algorithms, say $\mathcal{X}$ and $\mathcal{Y}$, that work under the same assumption on principal behaviours. The comparison technique we proposed relies on Kullback and Liebler's information diverge, and consists of measuring which algorithm best approximates the 'true' principal behaviour postulated by the model. For example, in order to compare $\mathcal{X}$ and $\mathcal{Y}$ in the model $\boldsymbol{\lambda}$, we propose to compute and compare

$$\mathrm{D}^n_{\mathrm{KL}}(\boldsymbol{\lambda} \,\|\, \mathcal{X}) \qquad \text{and} \qquad \mathrm{D}^n_{\mathrm{KL}}(\boldsymbol{\lambda} \,\|\, \mathcal{Y}).$$

Note that no simulations of algorithms $\mathcal{X}$ and $\mathcal{Y}$ are necessary; the mathematics provide a theoretical justification –rooted in concepts from Information Theory– stating e.g. that "in environment $\boldsymbol{\lambda}$, on average, algorithm $\mathcal{X}$ outperforms algorithm $\mathcal{Y}$ on training sequences of length n." Using our method in this paper we have been successful in showing a novel theoretical comparison between two $\boldsymbol{\beta}$-based algorithms well-known in the literature. Moreover, we explored the entire space of $\boldsymbol{\beta}$-based algorithms and proved constructively that for each principal behaviour

$\boldsymbol{\Theta}$, there exists a best approximating algorithm. Remarkably, this does not depend on n, the length of the training sequence. We regard this as the main result of the paper. More generally, another type of property one might desire to prove using the notion of information diverge is that $\lim_{n\to\infty} \mathrm{D}^n_{\mathrm{KL}}(\boldsymbol{\lambda} \parallel \mathcal{X}) = 0$, meaning that algorithm $\mathcal{X}$ approximates the true principal behaviour to an arbitrary precision, given a sufficiently long training sequence.

References

[1] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis. The role of trust management in distributed systems security. In Jan Vitek and Christian D. Jensen, editors, *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, volume 1603 of *Lecture Notes in Computer Science*, pages 185–210. Springer, 1999.

[2] S. Buchegger and J-Y. Le Boudec. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. In *P2PEcon 2004*, 2004.

[3] V. Cahill and E. Gray *et al.* Using trust for secure collaboration in uncertain environments. *IEEE Pervasive Computing*, 2(3):52–61, 2003.

[4] V. Cahill and J-M. Seigneur. The SECURE website. `http://secure.dsg.cs.tcd.ie`, 2004.

[5] D. Chalmers, M. Chalmers, J. Crowcroft, M. Kwiatkowska, R. Milner, E. ONeill, T. Rodden, V. Sassone, and M. Sloman. Ubiquitous computing: Experience, design and science. Version 4. Available from: `http://www-dse.doc.ic.ac.uk/Projects/UbiNet/GC/Manifesto/manifesto.pdf`, 2006.

[6] K. Chatzikokolakis, C. Palamidessi, and P. Panangaden. Anonymity protocols as noisy channels. In *Proceedings of TGC'06*, LNCS. Springer, 2007. To appear.

[7] Z. Despotovic and K. Aberer. 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*, pages 62–76. Springer, 2004.

[8] Z. Despotovic and K. Aberer. P2P reputation management: Probabilistic estimation vs. social networks. *Computer Networks*, 50(4):485–500, Mar. 2006.

[9] D. Gambetta. Can we trust trust? In Diego Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 213–237. University of Oxford, Department of Sociology, 2000. Chapter 13. Electronic edition `http://www.sociology.ox.ac.uk/papers/gambetta213-237.pdf`.

[10] E.T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, The Edinburgh Building, Cambridge, CB2 2RU, United Kingdom, 2003.

[11] A. Jøsang and J. Haller. Dirichlet reputation systems. In *Proceedings of the 2nd Intl Conference on Availability, Reliability and Security, ARES 2007*. 2007. To appear.

[12] A. Jøsang and R. Ismail. The beta reputation system. In *Proceedings from the 15th Bled Conference on Electronic Commerce, Bled*, 2002.

[13] K. Krukow. *Towards a Theory of Trust for the Global Ubiquitous Computer*. PhD thesis, University of Aarhus, Denmark, August 2006. ; available online (submitted): `http://www.brics.dk/~krukow`.

[14] K. Krukow, M. Nielsen, and V. Sassone. A logical framework for reputation systems. *Journal of Computer Security*, 2007. To appear. Available online `www.brics.dk/~krukow`.

[15] S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, March 1951.

[16] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation (for ebusinesses). In *Proceedings from 5th Annual Hawaii International Conference on System Sciences (HICSS'02)*, page 188. IEEE, 2002.

[17] M. Nielsen and K. Krukow. On the formal modelling of trust in reputation-based systems. In J. Karhumäki, H. Maurer, G. Paun, and G. Rozenberg, editors, *Theory Is Forever: Essays Dedicated to Arto Salomaa on the Occasion of His 70th Birthday*, volume 3113 of *Lecture Notes in Computer Science*, pages 192–204. Springer Verlag, 2004.

[18] M. Nielsen, G. Plotkin, and G. Winskel. Petri nets, event structures and domains. *Theoretical Computer Science*, 13:85–108, 1981.

[19] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[20] S. Reece, A. Rogers, S. Roberts, and N. Jennings. Rumours and reputation: Evaluating multi-dimensional trust within a decentralised reputation system. In *Proceedings of the 6th Intl Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-07*. 2007. Available at `http://eprints.ecs.soton.ac.uk/13260`.

[21] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, 2005.

[22] W.T.L. Teacy, J. Patel, N.R. Jennings, and M. Luck. 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*, pages 997–1004, New York, NY, USA, 2005. ACM Press.

[23] D. Varacca, H. Völzer, and G. Winskel. Probabilistic event structures and domains. In Philippa Gardner and Nobuko Yoshida, editors, *Proceedings from 15th International Conference on Concurrency Theory (CONCUR'04)*, volume 3170 of *Lecture Notes in Computer Science*, pages 481–496. Springer, 2004.