Towards a Formal Framework for Computational Trust
(Extended Abstract)

Vladimiro Sassone[1], Karl Krukow[2], and Mogens Nielsen[2]

[1] ECS, University of Southampton
[2] BRICS*, University of Aarhus

Abstract. We define a mathematical measure for the quantitative comparison of probabilistic computational trust systems, and use it to compare a well-known class of algorithms based on the so-called beta model. The main novelty is that our approach is formal, rather than based on experimental simulation.

1 Introduction

Computational trust is an abstraction inspired by the human concept of trust which aims at supporting decision-making by computational agents in the presence of unknown, uncontrollable and possibly harmful entities and in contexts where the lack of reliable information makes classical techniques useless. Such is for instance the case of open networks and ubiquitous computing, where it is entirely unrealistic to assume a priori level of understanding of the environment. Although it would be reductive to think of computational trust as a technique limited to just security, the latter certainly provides an important class of applications where, in general, access to resources is predicated on control policies that depend on the trust relationships in act between their managers and consumers.

As expected of an ineffable idea deeply linked with human emotions and experience, trust appears in computing in several very different forms, from description and specification languages to middleware, from social networks and management of credential to human-computer interaction. These rely in different degrees on a variety of underpinning mathematical theories, including e.g. logics, game theory, semantics, algorithmics, statistics, and probability theory. 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 area of computational trust one common classification distinguishes between 'probabilistic' and 'non-probabilistic' models (cf. e.g. [1, 2, 3, 11] for the latter and [6, 16, 10, 14] for the former). The non-probabilistic systems vary considerably and need further classification (e.g., as social networks or cognitive); in contrast, the probabilistic systems usually have common objectives and structure: they assume a particular

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

F.S. de Boer et al. (Eds.): FMCO 2006, LNCS 4709, pp. 175–184, 2007.

(probabilistic) model for principal behaviour at the outset, and then put forward algorithms for approximating such behaviour and thus making predictions. In such models the trust information about a principal is typically information about its past behaviour, its history. 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 models (called 'game-theoretical' by Sabater and Sierra [16]) rely on Gambetta's view of trust [7]:

> *"... trust 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, and follows the Bayesian approach to probability theory as advocated in e.g. [8] and exploited in works such as [13, 6, 17]. In particular, we borrow ideas from information theory to measure the quality of the behaviour-approximation algorithms and, therefore, suggest a formal framework for the comparison of probabilistic models.

Bayesian analysis consists of formulating hypotheses on real-world phenomena of interest, running experiments to test such hypotheses, 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.[1] 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 concrete, let us focus on a model of binary outcomes, which is very often used in practice. 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

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

[1] 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 $Prob(\boldsymbol{X})^{-1}$.

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 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.

In [14] we extend the framework from events with binary (success/failure) outcomes to complex, structured outcomes: namely, the configurations of finite, confusion-free event structures. In the 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.

In this paper we illustrate our main technical results from [14], viz., 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 [12], 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 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. [4] for an application of similar concepts to anonymity), and these 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 informally in the Introduction, while in §3 we illustrate our results on the formal of computational trust algorithms. We remand the reader to [14] for the formal proofs. Finally, §4 reflects on some of the basic hypotheses of the probabilistic models considered 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 §4 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 $x_1 x_2 \cdots x_n$ 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 $\mathcal{A}(\cdot \mid \boldsymbol{X})$ on the outcomes $\{y_1, \ldots, y_k\}$. That is, $\mathcal{A}$ satisfies:

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

Such distribution is meant 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 in the model $\boldsymbol{\lambda}$, 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.

3 Towards Comparing Probabilistic Trust-Based Systems

Closely related to Shannon's notion of entropy, Kullback and Leibler's information divergence [12] is a measure of the distance between two probability distributions. For $p = (p_1, \ldots, p_k)$ and $q = (q_1, \ldots, q_k)$ distributions on a set of k events, the Kullback-Leibler divergence from p to q is defined by

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

Information divergence resembles a distance in the mathematical sense: it can be proved that D_{KL} satisfies $\mathrm{D}_{\mathrm{KL}}(p \parallel q) \geq 0$, where the equality holds if and only if $p = q$; however, D_{KL} 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 $X_1 \cdots X_n$ of length n. Define $\mathrm{D}^n_{\mathrm{KL}}$, the nth *expected Kullback-Leibler divergence* from $\boldsymbol{\lambda}$ to $\mathcal{A}$ as:

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

Note that, for each possible input sequence $\boldsymbol{X} \in \boldsymbol{O}^n$, we evaluate the algorithm's performance as $\mathrm{D}_{\mathrm{KL}}(\mathit{Prob}(\cdot \mid \boldsymbol{X}\, \boldsymbol{\lambda}) \parallel \mathcal{A}(\cdot \mid \boldsymbol{X}))$, 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 $\mathcal{A}(\cdot \mid \boldsymbol{X})$.

An example. In order to exemplify our measure, we compare the $\boldsymbol{\beta}$-based algorithm of Mui *et al* [13] with the maximum-likelihood algorithm of Aberer and Despotovic [5]. 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 1. *If $\Theta = 0$ or $\Theta = 1$, Aberer and Despotovic's algorithm $\mathcal{B}$ from [5] computes a better approximation of the principal's behaviour than Mui et al's algorithm $\mathcal{A}$ from [13]. In fact, under the assumptions, $\mathcal{B}$ always computes the exact probability of success on any possible training sequence.*

The proof follows easily after observing that under the hypothesis on Θ there is only one n-sequence with non-zero probability, viz., either f^n or s^n.

Concerning the same comparison when $0 < \Theta < 1$, it suffices to observe that $\mathcal{B}$ assigns probability 0 to s on input f^k for all $k \geq 1$; this results in $\mathrm{D}^n_{\mathrm{KL}}(\Theta\, \boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{B}) = \infty$. It follows that $\mathcal{A}$ 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 $\mathrm{D}^n_{\mathrm{KL}}(\Theta\, \boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A}_\epsilon)$ as a function of ϵ. It turns out that for each $\Theta \neq 1/2$ and independently of n there is a unique $\overline{\epsilon}$ which minimises the distance $\mathrm{D}^n_{\mathrm{KL}}(\Theta\, \boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A}_\epsilon)$. Furthermore, $\mathrm{D}^n_{\mathrm{KL}}(\Theta\, \boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A}_\epsilon)$ is decreasing on the interval $(0, \bar{\epsilon}]$ and increasing on the interval $[\bar{\epsilon}, \infty)$. (Note of course that $\mathrm{D}^n_{\mathrm{KL}}(\Theta\, \boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A}_\epsilon) \to \infty$ when $\epsilon \to 0$.) By definition, we have:

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

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

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

where $\alpha = \log e$ is a positive constant obtained when differentiating the function log. It is proved in [14] that ϵ nullifies the derivative $d\, \mathrm{D}^n_{\mathrm{KL}}(\Theta\, \boldsymbol{\lambda}_{\mathbf{B}} \parallel \mathcal{A}_\epsilon)/d\epsilon$ if and only if

$$\Theta \neq 1/2 \qquad \text{and} \qquad \epsilon = \frac{2\Theta(1-\Theta)}{(2\Theta-1)^2}.$$

In addition to that, one can prove that in fact

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

and

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

Remarkably, these formulae are independent of n. We have thus the following result.

Theorem 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 $\overline{\epsilon}$, the better the algorithm. In fact, $\overline{\epsilon}$ tends to ∞ as Θ tends to $1/2$. Regarding $\mathcal{A}$ and $\mathcal{B}$, an application of Theorem 2 tells us that the former is optimal for $\Theta = 1/2 \pm 1/\sqrt{12}$, whilst –as anticipated by Theorem 1– the latter is such for $\Theta = 0$ and $\Theta = 1$.

We remark here 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.

4 Towards a Formal Model of Dynamic Behaviour

Our main motivation for this investigation is to put on formal grounds what we have been seeing in the literature, with the ultimate aim to exploit a sharpened understanding on systems and models. In our view, we succeeded in this to a comforting extent, by presenting the first ever formal framework for the comparisons of computational trust algorithms.

However, our probabilistic models must become 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 several 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.

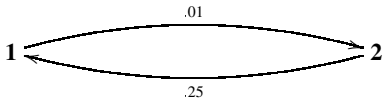
Some beta-based reputation systems attempt to deal with the first problem by introducing so-called 'forgetting factors.' Essentially this amounts to choosing a factor $0 \leq \delta \leq 1$, and then each time the parameters (α, β) of the pdf for Θ_p are updated, they are also scaled by δ. In particular, 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 explanation 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 an 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 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 [15].

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 1. 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.



$\pi_{\mathbf{1}} = 1$		$\pi_{\mathbf{2}} = 0$
$B_{\mathbf{1}}(a) = .95$	$O = \{a, b\}$	$B_{\mathbf{2}}(a) = .05$
$B_{\mathbf{1}}(b) = .05$		$B_{\mathbf{2}}(b) = .95$

Fig. 1. Example Hidden Markov Model

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 1. 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 [9], 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 §3: 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.

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.

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 λ, we propose to compute and compare

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

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 λ, on average, algorithm $\mathcal{X}$ outperforms algorithm $\mathcal{Y}$ on training sequences of length n." Using our method we have successfully in shown a theoretical comparison between two β-based algorithms well-known in the literature. Moreover, we explored the entire space of β-based algorithms and illustrated 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. More generally, another type of property one might desire to prove using the notion of information diverge is that $\lim_{n\to\infty} \mathrm{D}^n_{\mathrm{KL}}(\lambda \parallel \mathcal{X}) = 0$, meaning that algorithm $\mathcal{X}$ approximates the true principal behaviour to an arbitrary precision, given a sufficiently long training sequence.

References

1. Blaze, M., Feigenbaum, J., Ioannidis, J., Keromytis, A.D.: The role of trust management in distributed systems security. In: Vitek, J. (ed.) Secure Internet Programming. LNCS, vol. 1603, pp. 185–210. Springer, Heidelberg (1999)
2. Carbone, M., Nielsen, M., Sassone, V.: A formal model for trust in dynamic networks. In: Proceedings from Software Engineering and Formal Methods (SEFM'03), IEEE Computer Society Press, Los Alamitos (2003)

3. Carbone, M., Nielsen, M., Sassone, V.: A calculus for trust management. In: Lodaya, K., Mahajan, M. (eds.) FSTTCS 2004. LNCS, vol. 3328, pp. 161–173. Springer, Heidelberg (2004)
4. Chatzikokolakis, K., Palamidessi, C., Panangaden, P.: Anonymity protocols as noisy channels. In: Proceedings of TGC'06. LNCS, Springer, Heidelberg (to appear, 2007)
5. Despotovic, Z., Aberer, K.: A probabilistic approach to predict peers' performance in P2P networks. In: Klusch, M., Ossowski, S., Kashyap, V., Unland, R. (eds.) CIA 2004. LNCS (LNAI), vol. 3191, pp. 62–76. Springer, Heidelberg (2004)
6. Despotovic, Z., Aberer, K.: P2P reputation management: Probabilistic estimation vs. social networks. Computer Networks 50(4), 485–500 (2006)
7. Gambetta, D.: Can we trust trust. In: Gambetta, D. (ed.) Trust: Making and Breaking Cooperative Relations, pp. 213–237. University of Oxford, Department of Sociology, Ch. 13. Electronic edition (2000), http://www.sociology.ox.ac.uk/papers/gambetta213-237.pdf
8. Jaynes, E.T.: Probability Theory: The Logic of Science. Cambridge University Press, Cambridge (2003)
9. Jøsang, A., Ismail, R.: The beta reputation system. In: Proceedings from the 15th Bled Conference on Electronic Commerce, Bled (2002)
10. Krukow, K.: Towards a Theory of Trust for the Global Ubiquitous Computer. PhD thesis, University of Aarhus, Denmark (August 2006), Available at http://www.brics.dk/~krukow
11. Krukow, K., Nielsen, M., Sassone, V.: A logical framework for reputation systems. Journal of Computer Security (to appear, 2007), Available online http://eprints.ecs.soton.ac.uk/13656/
12. Kullback, S., Leibler, R.A.: On information and sufficiency. Annals of Mathematical Statistics 22(1), 79–86 (1951)
13. 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), p. 188. IEEE, Los Alamitos (2002)
14. Nielsen, M., Krukow, K., Sassone, V.: A bayesian model for event-based trust. In: Festschrift for Gordon D. Plotkin. ENTCS, Elsevier, Amsterdam (to appear, 2007)
15. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
16. Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artificial Intelligence Review 24(1), 33–60 (2005)
17. 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, pp. 997–1004. ACM Press, New York (2005)