Computational Trust

SFM'11
Bertinoro June 2011

Mogens Nielsen
University of Aarhus DK

Plan of talk

1) The grand challenge of Ubiquitous Computing
2) The role of Computational Trust in Ubiquitous Computing - a brief survey
3) Some results towards rigorously defined models of Computational Trust

Joint work with Sassone, Palamidessi, Krukow, Carbone, Cahill,….

Wave of Grand Challenge Initiatives

- Grand Challenges in Computer Science and Engineering
  - Computing Research Association, USA
- Fundamentals of Computer Science - Challenges and Opportunities
  - National Science Foundation, USA
- Short papers on Grand Challenges in Computer Science
  - Journal of ACM 50 (1) 2003
- 2020 Future of Computing
  - Nature, 2006
- UK Grand Challenges for Computing Research
  - EPSRC and others, currently

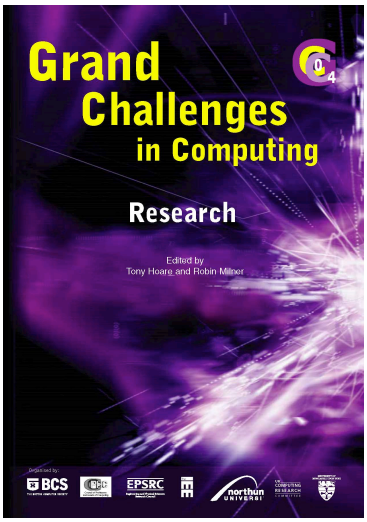
UK Grand Challenge

Engineering and Physical Sciences Research Council

British Computer Society

Institution of Electrical Engineers

ukcrc.org.uk/grand-challenge/index.cfm



UK Grand Challenges in Computing Research

1. In Vivo <=> In Silico
2. Ubiquitous Computing: UbiComp

   www-dse.doc.ic.ac.uk/Projects/UbiNet/GC
3. Memories for Life
4. The Architecture of Brain and Mind
5. Dependable Systems Evolution
6. Non-Classical Computation
7. Learning for Life
8. Bringing the Past to Life for the Citizen

Visions of UbiComp

- Billions of autonomous mobile networked entities
  - Mobile users
  - Mobile software agents
  - Mobile networked devices:
    - Mobile communication devices (phones, pagers, …)
    - Mobile computing devices (laptops, palmtops, …)
    - Commodity products (embedded devices)
- Entities will collaborate with each other
  - Resource sharing
    - Ad hoc networks, computational grids, …
  - Information sharing
    - Collaborative applications, recommendation systems, …

Data Security in UbiComp

- Data Security related properties of UbiComp
  - Large number of autonomous entities
  - Large number of administrative domains
  - No common trusted computing base
  - Virtual anonymity
- - excluding the use of traditional security mechanisms used in distributed systems – e.g. passwords, certificates, keys,...!
- ONE alternative approach:
  Trust based data security

Computational Trust - UbiComp

- Decisions related to communication made autonomously based on
  - entities' behaviour, reputation, credentials,..
  - other entities' recommendations,..
  - incomplete information, contexts, mobility,…
- Decisions related to communication made autonomously based on
  - a suitable *computational* notion of trust in order to achieve some required properties of communication between entities

Plan of talk

1) The grand challenge of Ubiquitous Computing
2) The role of Computational Trust in Ubiquitous Computing - a brief survey
3) Some results towards rigorously defined models of Computational Trust

Trust Surveys

- Trust in the Social Sciences
  - D. H. McKnight, N.L. Chervany: *The Meaning of Trust,* Trust in Cyber-societies, Springer LNAI 2246, 2001

McKnight and Chervany

- TRUST
  - Disposition
  - Structural
  - Affect/Attitude
  - Belief/Expectancy
  - Intention
  - Behaviour
- TRUSTEE
  - Competence
  - Benevolence
  - Integrity
  - Predictability
  - Openness, carefulness,..
  - People, Institutions,…

Computational Trust Surveys

- Computational Trust in UbiComp
  - T. Grandison, M. Sloman: *A Survey of Trust in Internet Applications*, IEEE Communications Surveys & Tutorials, 3(4), 2000
  - J. Sabater, C. Sierra: Review on Computational Trust and Reputation *Models*, Artificial Intelligence Review, 24, 33-60, 2005
  - A. Jøsang, R. Ismail, C. Boyd: *A Survey of Trust and Reputation for Online Service Provision*, Decision Support Systems, 43(2), 2006

Jøsang et al: Computational Trust

- Find adequate online substitutes for the traditional cues to trust and reputation from the physical world and identify information elements (specific to a particular online application) which are suitable for deriving measures of trust and reputation
- Take advantage of IT and the internet to create efficient systems for collecting that information, and for deriving measures of trust and reputation, in order to support decision making and to improve the quality of online markets

Jøsang et al: Trust semantics

- Trust values:
  - Discrete trust values
  - Summation or average of ratings
  - Probabilistic systems
  - Belief models
  - Fuzzy models
  - Flow models

Jøsang et al: Commercial systems

- Specific versus General
- Subjective versus Objective

- eBay's Feedback Forum
- Amazon
- Google Page Ranking

Computational Trust Applications

- Information provider applying trust in requesters
  - e.g. should I allow requester *R* to access my resource *r*?
    - Data security, Access control,..

- Information requester applying trust in providers
  - e.g. which of providers *P, Q, R,..* will provide the best service *s* for me?
    - Quality of services,..

Computational Trust Systems

- *Credential based*
  - the *KeyNote* System of Blaze et al
  - the *Delegation Logic* of Li et al
  - .....
- *Reputation based*
  - the *Beta Reputation System* of Jøsang et al
  - the *Eigentrust Syste*m of Kamvar et al
  - .....

Computational Trust

- Trust formation
  - Individual experience
  - Recommendation from known (trusted) third parties
  - Reputation (recommendation from many strangers)
- Trust evolution
  - Incorporating new trust formation data
  - Expiration of old trust values
- Trust exploitation
  - Risk analysis
  - Feedback based on experience
  - Context dependence

UbiComp Challenges

- Science Goal
  - to develop a coherent informatics science whose concepts, calculi, models, theories and tools allow descriptive, explanatory and predictive analysis of ubiquitous computing at many levels of abstraction
  - to employ these theories to derive all its systems and software, including languages
  - to analyse and justify all its constructions by these theories and tools

UbiComp: *Computational* Trust

- On *trust:*

  "..*trust* between autonomous agents will be an important ingredient..... A discipline of trust will only be effective if it is *rigorously defined*..."

- On *rigorously defined:*

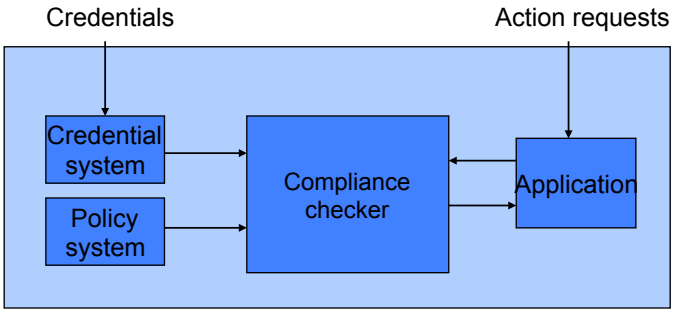
  "...tools for formalization, specification, validation, analysis, diagnosis, evaluation, ..."

Plan of talk

1) The grand challenge of Ubiquitous Computing
2) The role of Computational Trust in Ubiquitous Computing - a brief survey
3) Some results towards rigorously defined models of Computational Trust
   a) Trust in requesters – based on credentials
   b) Trust in providers – based on reputation

Trust in Requesters – Based on Credentials

Trust Management - Blaze, Feigenbaum et al



Trust management elements

- Language for Actions
- Naming scheme for Principals
- Language for Trust-Policies
- Language for Credentials
- Compliance checker and interface

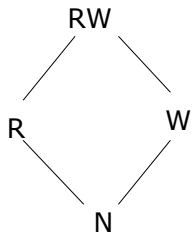
- Blaze, Feigenbaum, Ioannidis, Keromytis: *The Role of Trust Management ion Distributed Systems Security*, Springer LNCS 1603, 185-210, 1999
- Li, Mitchell: *A Role-based Trust-management Framework*, DISCEX III, IEEE Computer Society Press, 2003

Trust policies

- Each principal defines a trust policy which declares how it computes its trust in every other principal
- A small policy language could have constructs like
  - Refer to the information registered locally
  - Refer to information registered by other principals
  - Refer to the information P would obtain if it were to compute its trust
  - Other operations…

Example: A simple trust setting

- Let $\mathcal{T}$ be {N, R, W, RW}



Example trust policies

b: $\lambda x.\ (x{=}c \Rightarrow \mathrm{W},\ldots.)$ abstraction

a: $\lambda x.\ (\lceil b \rceil x \vee \mathrm{R})$ referencing

a: $\lambda x.\ (\ (\lceil a \rceil b \wedge \lceil b \rceil x)\ \vee\ \mathrm{R})$ discounting

a: $\lambda x.\ (\lceil b \rceil x)$
b: $\lambda x.\ (\lceil a \rceil x)$ cyclic delegation

Modeling Trust

- Scenario with
  - A set $\mathcal{P}$ of principals (ranged over by *a,b,c*)
  - A set $\mathcal{T}$ of trust values
- Trust information of a system represented by
  - `trust-state`: P → P → T
  - `trust-state`(A)(B): represents A's trust in B

Modeling the web of Trust

Each Principal specifies a *policy*
which is a local contribution to the global trust

Given principals *a* with policies π_a:

$$\pi_a : [\ \mathcal{P} \to \mathcal{P} \to \mathcal{T}\ ] \to [\ \mathcal{P} \to \mathcal{T}\ ]$$

The collection of π_a's induces a global trust function:

$$\Pi : [\ \mathcal{P} \to \mathcal{P} \to \mathcal{T}\ ] \to [\mathcal{P} \to \mathcal{P} \to \mathcal{T}\ ]$$

Definition of Trust

Assume T is a lattice/cpo, given a $\leq$-continuous global trust function

$$\Pi : [\ \mathcal{P} \rightarrow \mathcal{P} \rightarrow \mathcal{T}\ ] \rightarrow [\mathcal{P} \rightarrow \mathcal{P} \rightarrow \mathcal{T}\ ]$$

TRUST is defined as the *least fixed-point* of Π

Weeks: *Understanding Trust Management Systems,* IEEE Symposium on Security and Privacy, 2001

Lattices and continuity

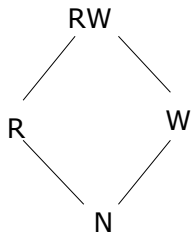
In a *complete lattice* T = (D, $\leq$) all subsets X of D have a least upper bound $\cup$X and a greatest lower bound $\cap$X

F : D $\rightarrow$ D is $\leq$-continuous iff $F(\cup X) = \cup F(X)$
implying that F is $\leq$-monotone
F : D $\rightarrow$ D is $\leq$-monotone iff $x \leq y \Rightarrow F(x) \leq F(y)$

For F : D $\rightarrow$ D $\leq$ continuous, the least fixed point of F exists and is equal to $\cup F^i(\perp)$

Example: A simple trust setting

- Let $\mathcal{T}$ be {N, R, W, RW}



Example (1)

- Suppose we have the following policies:

	a	*b*	*c*
d	⌈*f*⌉ ∨ W	⌈*e*⌉ ∧ W	N
e	R	R	⌈*f*⌉
f	⌈*e*⌉	N	⌈*e*⌉

Example (2)

- The computation:

	a	*b*	*c*
d	⌈*f*⌉ ∨ W	⌈*e*⌉ ∧ W	N
e	R	R	⌈*f*⌉
f	⌈*e*⌉	N	⌈*e*⌉

	a	*b*	*c*
d	[N,RW]	[N,RW]	[N,RW]
e	[N,RW]	[N,RW]	[N,RW]
f	[N,RW]	[N,RW]	[N,RW]

Example (3)

- The computation:

	a	*b*	*c*
d	⌈*f*⌉ ∨ W	⌈*e*⌉ ∧ W	N
e	R	R	⌈*f*⌉
f	⌈*e*⌉	N	⌈*e*⌉

	a	*b*	*c*
d	[W,RW]	[N,W]	[N,N]
e	[R,R]	[R,R]	[N,RW]
f	[N,RW]	[N,N]	[N,RW]

Example (4)

- The computation:

	a	*b*	*c*
d	⌈*f*⌉ ∨ W	⌈*e*⌉ ∧ W	N
e	R	R	⌈*f*⌉
f	⌈*e*⌉	N	⌈*e*⌉

	a	*b*	*c*
d	[W,RW]	[N,N]	[N,N]
e	[R,R]	[R,R]	[N,RW]
f	[R,R]	[N,N]	[N,RW]

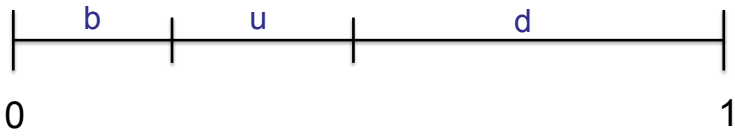
Example (5)

- The computation:

	a	*b*	*c*
d	⌈*f*⌉ ∨ W	⌈*e*⌉ ∧ W	N
e	R	R	⌈*f*⌉
f	⌈*e*⌉	N	⌈*e*⌉

	a	*b*	*c*
d	[RW,RW]	[N,N]	[N,N]
e	[R,R]	[R,R]	[N,RW]
f	[R,R]	[N,N]	[N,RW]

Belief Models



b – belief
u – uncertainty
d - disbelief

Trust domain

- T is equipped with two orderings $\leq$ and $\leq$ where
- $\leq$ represents information ordering
- $\leq$ represents trust ordering

Example: Proof carrying requests

- Idea: Assume *r* sending a request to *a,* requiring *high* trust

  a: $\lambda x.\ (\lceil b \rceil x \vee .....)$
  b: $\lambda x.\ (x{=}r \Rightarrow high, ...)$

Example: Proof carrying request

Theorem
Assume that $\leq$ is $\leq$-continuous
and that Π is $\leq$-monotone

Given m: $\mathcal{P} \to \mathcal{P} \to \mathcal{T}$, if

- $m \leq \bot_{\leq}$
- $m \leq \Pi\,(m)$

then $m \leq \mathrm{lfp}_{\leq}\ \Pi$

Example: Proof carrying request

- *Idea:* Requester provides m along with his request (sufficient for the request to be met) as an argument for $m \leq \mathrm{lfp}_{\leq} \Pi$

- Send m to all principals a for which $m(a)$ is different from $\lambda p.\bot_{\leq}$, and ask a to check that $m \leq \pi_a(m)$
- If this is the case for all such principals, conclude that $m \leq \Pi(m)$, and hence $m \leq \mathrm{lfp}_{\leq} \Pi$

Example: Proof carrying requests

- Idea: Assume r sending a request to a, requiring *high* trust

  a: $\lambda x.$ ($\lceil b \rceil x \vee$ .....)
  b: $\lambda x.$ ($x{=}r \Rightarrow high$,....)

Trust in Providers – Based on Reputations
EigenTrust Algorithm - Kamvar et al

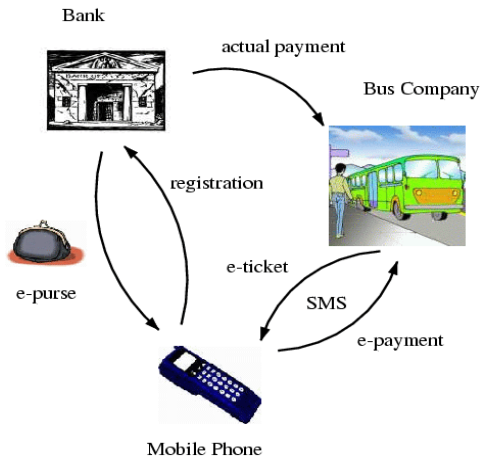
- Peers (i,j,..) interact and mutually rate interactions as being either *satisfactory* or *unsatisfactory*:
  - $s_{ij} = \max (N_{sat}(i,j) - N_{unsat}(i,j), 0)$
- These ratings are normalised
  - $c_{ij} = s_{ij} / \Sigma_j s_{ij}$
- $[c_{ij}]$ is a Markov chain with stationary distribution $[t_j]$ - interpreted as the global trust in peer j

EigenTrust Algorithm for P2P Networks

- System simulations show that EigenTrust can significantly reduce the number of non-authentic file downloads in a P2P filesharing system, even when up to 70% of the peers maliciously cooperate
- But what is Eigentrust computing, - e.g. what does it mean that the trust in some peer is .75?
- Kamwar, Schlosser, Garcia-Molina: The Eigentrust Algorithm for Reputation Management in P2P Networks, proceedings of WWW'03, ACM Press, 640-651, 2003

Trust in Providers – Based on Reputations

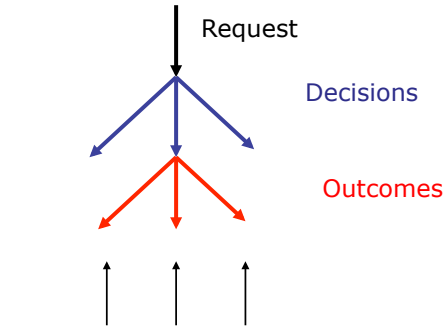
Beta Reputation - Jøsang et al



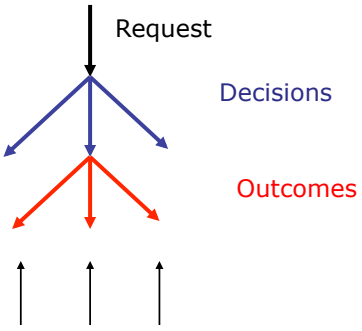
E-Purse Scenario

- Consider a situation where a user is considering requesting an amount *m* of e-cash from a bank
- Seen from the point of view of the user, an "untrusted " bank may
  - deny the request, e.g. because the bank server is down for maintenance
  - grant the request, but withdraw an amount different from *m* from users account
  - grant the request, but the transferred e-cash may be forged

Trust/Risk Based Decisions

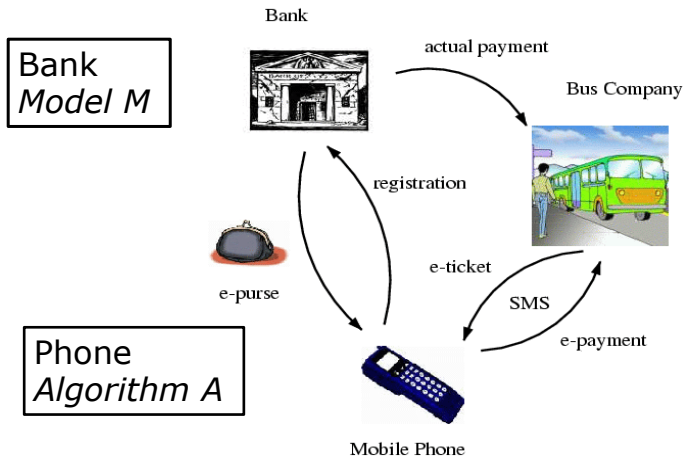


Probabilistic Computational Trust



$$\exp = \sum_i \text{cost}(o_i) * \text{likelihood}(o_i)$$

Models and Algorithms



Probabilistic Models for Computational Trust

- Given a (finite) set of outcomes of interactions
  - $O = \{o_1, o_2, \ldots, o_m\}$
- A probabilistic model M of principal behaviour defines for $h \in O^*$ and $o_i \in O$
  - $P(h \mid M)$ - the probability of observing h in M
  - $P(o_i \mid h\ M)$ - the probability of o_i in the next interaction given observation h in M

Probabilistic Computational Trust Algorithms

- Given a (finite) set of outcomes of interactions
  - $O = \{o_1, o_2,...,o_m\}$
- A probabilistic computational trust algorithm A
  - takes as input a history $h \in O^*$ and
  - outputs a probability distribution on O

  $A(o_i \mid h) \in [0,1]$ for $i = 1,2,..,m$

The Goal for Probabilistic Trust Algorithms

- Algorithm A producing $A(o_i \mid h)$ should approximate Model M probabilities $P(o_i \mid h\ M)$ as well as possible!
- Notice that this gives rise to rigid versions of soft correctness question:
  - how well does a particular algorithm approximate the model?
  - how robust is it - wrt. the model and its parameters?

A Concrete Simple Probabilistic Model

- The Bernoulli Model – $M_B(\theta)$
  - Assume that the behaviour of a particular principal, *p*, has only two outcomes, with a probability θ for *success* (and $1-\theta$ for *failure*)
- Algorithm *A*
  - Output: a probability distribution $\{s, f\} \rightarrow [0, 1]$
- The Goal
  - *A* should approximate $(\theta, 1-\theta)$ as well as possible

Probabilistic Trust Algorithms

- Focus on two example algorithms:
  - P2P Reputation Management of Despotocvic et al
  - Computational Model for eBusiness of Mui et al

Despotovic et al 2004: *Algorithm A_D*

- The Specification (of trust computation algorithm *A*)
  - Input: a sequence of observations $h = x_1x_2..x_n \in \{s, f\}^*$
  - Output: a probability distribution $\{s, f\} \rightarrow [0, 1]$
- The algorithm A_D for $M_B(\theta)$
  - $A_D(s \mid h) = N_s(h) / |h|$
  - $A_D(f \mid h) = N_f(h) / |h|$
- Despotovic, Aberer: A Probabilistic Approach to Predict Peers' Performance in P2P Networks, CIA'04, Springer LNCS 3192, 62-76, 2004

Mui et al 2002: *Algorithm A_M*

- The Specification (of trust computation algorithm A)
  - Input: a sequence of observations $h = x_1x_2..x_n \in \{s, f\}^*$
  - Output: a probability distribution $\{s, f\} \rightarrow [0, 1]$
- The algorithm A_M:
  - $A_M(s \mid h) = (N_s(h) + 1) / (|h| + 2)$
  - $A_M(f \mid h) = (N_f(h) + 1) / (|h| + 2)$
- Mui, Motashemi, Halberstadt: A Computatinal Model of Trust and Reputation for eBusinesses, HICSS'02, IEEE Press, 2002

A Question: how to choose

- The Goal
  - Algorithm A should approximate $(\theta, 1-\theta)$ as well as possible
- Which of the two algorithms A_D and A_M performs best relative to this goal?
  - *Experimental approach*: answers given based on experiments in simulation environments
  - *Theoretical approach*: answer given in terms of mathematical results in our probability model

How to measure "approximate"?

- The "distance from a true distribution $\boldsymbol{p}$ to an approximation $\boldsymbol{q}$" (here on $O = \{o_1,o_2,...,o_m\}$) can be measured as e.g
  - the Relative Entropy (also called the Kullback-Leibler divergence):

    $D(\boldsymbol{p}\,||\boldsymbol{q}) = \Sigma_i \; \boldsymbol{p}(o_i) \times log_2(\boldsymbol{p}(o_i) / \boldsymbol{q}(o_i))$
  - $D(\boldsymbol{p}\,||\boldsymbol{q}) = \Sigma_i \; (\boldsymbol{p}(o_i) - \boldsymbol{q}(o_i))^2$
- Results holds for e.g. both these choices

The Goal of a Probabilistic Algorithm: Formally

- The Goal
  - Algorithm A producing $A(o_i \mid h)$ should approximate $P(o_i \mid h\, M)$ as well as possible
- We choose to interpret "as well as possible" in terms of the expected distance between the two distributions:

$$ED^n(\boldsymbol{M} \,||\, \boldsymbol{A}) = \sum_{h \in O^n} p(h \mid \boldsymbol{M}) \times D(P(\cdot \mid h\boldsymbol{M}) || \boldsymbol{A}(\cdot \mid h))$$

How to choose: Formally

- Comparing A_D and A_M against M_B:

If $\theta = 0$ or $\theta = 1$ then for all n

$$ED^n(M_B(\theta), A_D) = 0 < ED^n(M_B(\theta), A_M)$$

If $0 < \theta < 1$ then for all n

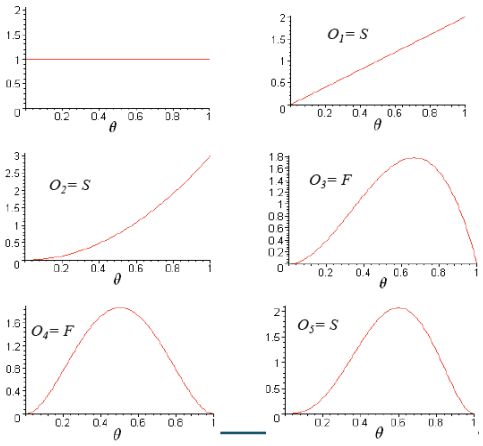
$$ED^n(M_B(\theta), A_M) < ED^n(M_B(\theta), A_D) = \infty$$

Bayesian Approach

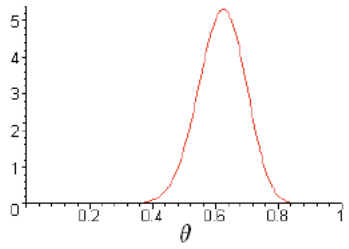
- Bayes' theorem:
  $P(\theta \mid h,M) = P(\theta \mid M) \times (P(h \mid \theta,M) / P(h \mid M))$

- For the model M_B choosing
  - $P(\theta \mid M_B) = Beta(a, \beta)(\theta)$
  - $Beta(a, \beta)(\theta) = \theta^{a-1} (1-\theta)^{\beta-1} \Gamma(a+\beta) / \Gamma(a) \Gamma(\beta)$
- Allows the following simple "algorithms" computing the a posteriori information
  - $P(\theta \mid h,M_B) = Beta(a + N_s(h), \beta + N_f(h))$
  - $E(Beta(a, \beta)) = a / (a + \beta)$

Beta (a, β)



Beta (a, β) – after 25 "S" and 15 "F"



Two Examples Generalised

- A_D the P2P Reputation Management of Despotocvic et al
  - an example of the Bayesian approach with $a=\beta=0$
- A_M the Computational Model for eBusiness of Mui et al
  - an example of the Bayesian approach with $a=\beta=1$

- Generalize to all symmetric Beta priors, i.e. for arbitrary real numbers $\varepsilon \geq 0$:
  - $A_\varepsilon(s \mid h) = (N_s(h) + \varepsilon) / (|h| + 2\varepsilon)$
  - $A_\varepsilon(f \mid h) = (N_f(h) + \varepsilon) / (|h| + 2\varepsilon)$
- What is a good choice of ε - and how does this choice depend on θ and n?

Some Theoretical Answers: how to choose

For any $\theta \in [0,1]$, $\theta \neq 1/2$, there exists an ε_θ which for any n minimizes $ED^n(M_B(\theta), A_\varepsilon)$. Furthermore, ε_θ is defined as the following function of θ

$$\varepsilon_\theta \quad = \quad 2\theta(1-\theta) \;/\; (2\theta-1)^2$$

Meaning: unless behaviour is completely random, there is a unique best algorithm (choosing $\varepsilon := \varepsilon_\theta$) outperforming all other A_ε algorithms, $\varepsilon \geq 0$

Some Theoretical Answers: *Robustness*

Furthermore, $ED^n(M_B(\theta), A_\varepsilon)$ is continuous (as a function of ε) – decreasing on the interval $(0, \varepsilon_\theta)$ and increasing on $(\varepsilon_\theta, \infty)$

Meaning: The closer ε is to ε_θ the better performance of A_ε

Some Theoretical Answers: how to choose

Given a particular ε, the algorithm A_ε is an optimal choice (for all n, and amongst all the A_ε algorithms) for

$$\theta = {}^1/_2 \ +/- \ {}^1/_{2\ sqrt(2\varepsilon+1)}$$

Example: A_M is optimal for $\theta = {}^1/_2 \ +/- \ {}^1/_{sqrt(12)}$

Non-symmetric priors

- Using the prior *Beta(a, β)* yields the following algorithm computing the mean of the posterior distribution:
  - $A_{\alpha,\beta}(s \mid h) = (N_s(h) + \alpha) / (|h| + a + \beta)$
  - $A_{\alpha,\beta}(f \mid h) = (N_f(h) + \beta) / (|h| + a + \beta)$
- How to choose the parameters α and *β?*

Non-symmetric priors

- Assume the true behaviour (M_B) to be $Beta(a_t, \beta_t)$, define the "risk" of an algorithm $A_{\alpha,\beta}$
- $R^n(A_{\alpha,\beta}) = \int_{[0;1]} Beta(a_t, \beta_t)\, ED^n(M_B(\theta), A_{\alpha,\beta})\, d\theta$
- *Theorem*
  For all n, $R^n(A_{\alpha,\beta})$ is minimum for $a = a_t$ and $\beta = \beta_t$

Non-symmetric priors

- Assume no knowledge of the true behaviour (θ in M_B), define the "risk" of an algorithm $A_{\alpha,\beta}$
- $R^n(A_{\alpha,\beta}) = \int_{[0;1]} ED^n(M_B(\theta), A_{\alpha,\beta})\, d\theta$
- *Theorem*
  For all n, $R^n(A_{\alpha,\beta})$ is minimum for $\alpha = \beta = 1$

Many More Issues to be Modelled....

- Trust formation
  - Individual experience
  - Recommendation from known (trusted) third parties
  - Reputation (recommendation from many strangers)
- Trust evolution
  - Incorporating new trust formation data
  - Expiration of old trust values
- Trust exploitation
  - Risk analysis
  - Feedback based on experience
  - Context dependence

Some Publications

- ElSalamouny, Nielsen, Sassone, HMM-based Trust Model, FAST'09, Springer LNCS 5893, 21-35, 2010
- Krukow, Nielsen, Sassone: Probabilistic Computational Trust, Perspectives in Concurrency Theory, Universities Press, 295-316, 2009
- Nielsen, Krukow, Sassone: *Trust Models in Ubiquitous Computing*, Phil. Trans. of the Royal Society, Volume 366, Number 1881, 3781-3793, 2008
- Nielsen, Krukow, Sassone: *A Bayesian Model for Event-based Trust*, Electronic Notes in Theoretical Computer Science, vol. 172, 499-521, 2007
- Krukow, Nielsen: *From Simulations to Theorems*, FAST'06, Springer LNCS, 96-111, 2007

Some more Publications

- Nielsen, Krukow, Sassone: *A Logical Framework for Reputation Systems*, Journal of Computer Security, vol. 16 nr. 1, 63-101, 2007
- Nielsen, Krukow, Sassone: *Towards a Formal Framework for Computational Trust*, 5th International Symposium on Formal Methods for Components and Objects, Springer, 175-184, 2007
- Nielsen, Krukow, 2007, *Trust Structures*, International Journal of Information Security, vol. 6 nr. 2-3, 153-181, 2007
- Krukow, Nielsen, Sassone: *A Framework for Concrete Reputation-Systems with Applications to History-Based Access Control,* CCS'05, ACM Press, 2005

Some more Publications

- Carbone, Nielsen, Sassone: *A Calculus for Trust Management*, FSTTCS'04, Springer LNCS 3328, 2004
- Nielsen, Krukow: *On the Formal Modeling of Trust in Reputation-Based Systems*, Springer LNCS 3113, 2004
- Nielsen, Krukow: *Towards a Formal Notion of Trust*, PPDP'03, IEEE, 2003
- Carbone, Nielsen, Sassone: *A Formal Model for Trust in Dynamic Networks*, SEFM, IEEE, 2003
- Cahill, Shand, Gray, Dimmock, Twigg, Bacon, English, Wagaella, Terzis, Nixon, Bryce, Seigneur, Carbone, Krukow, Jensen, Chen, Nielsen: *Using trust for Secure Collaboration in Uncertain Environments*, IEEE Pervasive Computing, 2003

References – Reputation Based Trust

- Despotovic, Aberer: A Probabilistic Approach to Predict Peers' Performance in P2P Networks, proceedings of CIA'04, Springer LNCS 3192, pp 62-76, 2004
- Mui, Motashemi, Halberstadt: A Computatinal Model of Trust and Reputation for eBusinesses, proceedings of HICSS'02, IEEE Press, 2002
- Kamwar, Schlosser, Garcia-Molina: The Eigentrust Algorithm for Reputation Management in P2P Networks, proceedings of WWW'03, ACM Press, pp 640-651, 2003
- Jøsang, Ismail: *The Beta Reputation System*, 15th Conference on Electronic Commerce, 2002
- Shmatikov, Talcott: *Reputation-Based Trust Management*, Journal of Computer Security, 2005

Thank you for your attention!