A Bayesian model for event-based trust

Elements of a foundation for computational trust

Vladimiro Sassone

ECS, University of Southampton

joint work K. Krukow and M. Nielsen

Computational trust

Trust is an ineffable notion that permeates very many things.

What trust are we going to have in this talk?

Computer idealisation of “trust” to support decision-making in open networks. No human emotion, nor philosophical/sociological concept.

Gathering prominence in open applications involving safety guarantees in a wide sense

- credential-based trust: e.g., public-key infrastructures, authentication and resource access control, network security.
- reputation-based trust: e.g., social networks, P2P, trust metrics, probabilistic approaches.
- trust models: e.g., security policies, languages, game theory.
- trust in information sources: e.g., information filtering and provenance, content trust, user interaction, social concerns.

Computational trust

Trust is an ineffable notion that permeates very many things.

What trust are we going to have in this talk?

Computer idealisation of “trust” to support decision-making in open networks. No human emotion, nor philosophical/sociological concept.

Gathering prominence in open applications involving safety guarantees in a wide sense

- credential-based trust: e.g., public-key infrastructures, authentication and resource access control, network security.
- reputation-based trust: e.g., social networks, P2P, trust metrics, probabilistic approaches.
- trust models: e.g., security policies, languages, game theory.
- trust in information sources: e.g., information filtering and provenance, content trust, user interaction, social concerns.

Computational trust

Trust is an ineffable notion that permeates very many things.

What trust are we going to have in this talk?

Computer idealisation of “trust” to support decision-making in open networks. No human emotion, nor philosophical/sociological concept.

Gathering prominence in open applications involving safety guarantees in a wide sense

- credential-based trust: e.g., public-key infrastructures, authentication and resource access control, network security.
- reputation-based trust: e.g., social networks, P2P, trust metrics, probabilistic approaches.
- trust models: e.g., security policies, languages, game theory.
- trust in information sources: e.g., information filtering and provenance, content trust, user interaction, social concerns.

Computational trust

Trust is an ineffable notion that permeates very many things.

What trust are we going to have in this talk?

Computer idealisation of “trust” to support decision-making in open networks. No human emotion, nor philosophical/sociological concept.

Gathering prominence in open applications involving safety guarantees in a wide sense

- credential-based trust: e.g., public-key infrastructures, authentication and resource access control, network security.
- reputation-based trust: e.g., social networks, P2P, trust metrics, probabilistic approaches.
- trust models: e.g., security policies, languages, game theory.
- trust in information sources: e.g., information filtering and provenance, content trust, user interaction, social concerns.

Computational trust

Trust is an ineffable notion that permeates very many things.

What trust are we going to have in this talk?

Computer idealisation of “trust” to support decision-making in open networks. No human emotion, nor philosophical/sociological concept.

Gathering prominence in open applications involving safety guarantees in a wide sense

- credential-based trust: e.g., public-key infrastructures, authentication and resource access control, network security.
- reputation-based trust: e.g., social networks, P2P, trust metrics, probabilistic approaches.
- trust models: e.g., security policies, languages, game theory.
- trust in information sources: e.g., information filtering and provenance, content trust, user interaction, social concerns.

Trust and reputation systems

Reputation

- behavioural: perception that an agent creates through past actions about its intentions and norms of behaviour.
- social: calculated on the basis of observations made by others.

An agent's reputation may affect the trust that others have toward it.

Trust

- subjective: a level of the subjective expectation an agent has about another's future behaviour based on the history of their encounters and of hearsay.

Confidence in the trust assessment is also a parameter of importance.

Trust and reputation systems

Reputation

- behavioural: perception that an agent creates through past actions about its intentions and norms of behaviour.
- social: calculated on the basis of observations made by others.

An agent's reputation may affect the trust that others have toward it.

Trust

- subjective: a level of the subjective expectation an agent has about another's future behaviour based on the history of their encounters and of hearsay.

Confidence in the trust assessment is also a parameter of importance.

Trust and security

E.g.: Reputation-based access control

p's 'trust' in q's actions at time t, is determined by p's observations of q's behaviour up *until* time t according to a given policy ψ.

Example

You download what claims to be a new cool browser from some unknown site. Your trust policy may be:

- *allow the program to connect to a remote site if and only if it has neither tried to open a local file that it has not created, nor to modify a file it has created, nor to create a sub-process.*

Trust and security

E.g.: Reputation-based access control

p's 'trust' in q's actions at time t, is determined by p's observations of q's behaviour up *until* time t according to a given policy ψ.

Example

You download what claims to be a new cool browser from some unknown site. Your trust policy may be:

- *allow the program to connect to a remote site if and only if it has neither tried to open a local file that it has not created, nor to modify a file it has created, nor to create a sub-process.*

Outline

1. Some computational trust systems
2. Towards model comparison
3. Modelling behavioural information
   - Event structures as a trust model
4. Probabilistic event structures
5. A Bayesian event model

Outline

1. Some computational trust systems
2. Towards model comparison
3. Modelling behavioural information
   - Event structures as a trust model
4. Probabilistic event structures
5. A Bayesian event model

Simple Probabilistic Systems

The model λ_θ:

- Each principal p behaves in each interaction according to a fixed and independent probability θ_p of 'success' (and therefore $1 - \theta_p$ of 'failure').

The framework:

- Interface (Trust computation algorithm, $\mathcal{A}$):
  - Input: A sequence $h = x_1 x_2 \cdots x_n$ for $n \geq 0$ and $x_i \in \{\mathbf{s}, \mathbf{f}\}$.
  - Output: A probability distribution $\pi : \{\mathbf{s}, \mathbf{f}\} \to [0, 1]$.
- Goal:
  - Output π approximates $(\theta_p, 1 - \theta_p)$ as well as possible, under the hypothesis that input h is the outcome of interactions with p.

Simple Probabilistic Systems

The model λ_θ:

- Each principal p behaves in each interaction according to a fixed and independent probability θ_p of 'success' (and therefore $1 - \theta_p$ of 'failure').

The framework:

- Interface (Trust computation algorithm, $\mathcal{A}$):
  - Input: A sequence $h = x_1 x_2 \cdots x_n$ for $n \geq 0$ and $x_i \in \{\mathbf{s}, \mathbf{f}\}$.
  - Output: A probability distribution $\pi : \{\mathbf{s}, \mathbf{f}\} \to [0, 1]$.
- Goal:
  - Output π approximates $(\theta_p, 1 - \theta_p)$ as well as possible, under the hypothesis that input h is the outcome of interactions with p.

Maximum likelihood (Despotovic and Aberer)

Trust computation $\mathcal{A}_0$

$$\mathcal{A}_0(\mathbf{s} \mid h) = \frac{\mathrm{N}_{\mathbf{s}}(h)}{|h|} \qquad \mathcal{A}_0(\mathbf{f} \mid h) = \frac{\mathrm{N}_{\mathbf{f}}(h)}{|h|}$$

$$\mathrm{N}_x(h) = \text{"number of } x\text{'s in } h\text{"}$$

Bayesian analysis inspired by $\boldsymbol{\lambda}_\beta$ model: $f(\theta \mid \alpha\, \beta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$

Properties:

- Well defined semantics: $\mathcal{A}_0(\mathbf{s} \mid h)$ is interpreted as a *probability* of success in the next interaction.
- Solidly based on probability theory and Bayesian analysis.
- Formal result: $\mathcal{A}_0(\mathbf{s} \mid h) \to \theta_p$ as $|h| \to \infty$.

Maximum likelihood (Despotovic and Aberer)

Trust computation $\mathcal{A}_0$

$$\mathcal{A}_0(\mathbf{s} \mid h) = \frac{\mathrm{N}_{\mathbf{s}}(h)}{|h|} \qquad \mathcal{A}_0(\mathbf{f} \mid h) = \frac{\mathrm{N}_{\mathbf{f}}(h)}{|h|}$$

$\mathrm{N}_x(h) =$ "number of x's in h"

Bayesian analysis inspired by $\boldsymbol{\lambda}_\beta$ model: $f(\theta \mid \alpha\, \beta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$

Properties:

- Well defined semantics: $\mathcal{A}_0(\mathbf{s} \mid h)$ is interpreted as a *probability* of success in the next interaction.
- Solidly based on probability theory and Bayesian analysis.
- Formal result: $\mathcal{A}_0(\mathbf{s} \mid h) \to \theta_p$ as $|h| \to \infty$.

Beta models (Mui et al)

Even more tightly inspired by Bayesian analysis and by λ_β

Trust computation $\mathcal{A}_1$

$$\mathcal{A}_1(\mathbf{s} \mid h) = \frac{\mathrm{N}_{\mathbf{s}}(h) + 1}{|h| + 2} \qquad \mathcal{A}_1(\mathbf{f} \mid h) = \frac{\mathrm{N}_{\mathbf{f}}(h) + 1}{|h| + 2}$$

$\mathrm{N}_x(h) =$ "number of x's in h"

Properties:

- Well defined semantics: $\mathcal{A}_1(\mathbf{s} \mid h)$ is interpreted as a *probability* of success in the next interaction.
- Solidly based on probability theory and Bayesian analysis.
- Formal result: Chernoff bound $Prob[error \geq \epsilon] \leq 2e^{-2m\epsilon^2}$, where m is the number of trials.

Beta models (Mui et al)

Even more tightly inspired by Bayesian analysis and by $\boldsymbol{\lambda_\beta}$

Trust computation $\mathcal{A}_1$

$$\mathcal{A}_1(\mathbf{s} \mid h) = \frac{\mathrm{N}_{\mathbf{s}}(h) + 1}{|h| + 2} \qquad \mathcal{A}_1(\mathbf{f} \mid h) = \frac{\mathrm{N}_{\mathbf{f}}(h) + 1}{|h| + 2}$$

$$\mathrm{N}_x(h) = \text{“number of } x\text{'s in } h\text{”}$$

Properties:

- Well defined semantics: $\mathcal{A}_1(\mathbf{s} \mid h)$ is interpreted as a *probability* of success in the next interaction.
- Solidly based on probability theory and Bayesian analysis.
- Formal result: Chernoff bound $Prob[error \geq \epsilon] \leq 2e^{-2m\epsilon^2}$, where m is the number of trials.

Our elements of foundation

We tried to consolidate in two directions:

1. model comparison
2. complex event model

Outline

Cross entropy

An information-theoretic "distance" on distributions

Cross entropy of distributions $\mathbf{p}, \mathbf{q} : \{o_1, \ldots, o_m\} \rightarrow [0, 1]$.

$$\mathrm{D}(\mathbf{p} \,||\, \mathbf{q}) = \sum_{i=1}^{m} \mathbf{p}(o_i) \cdot \log\big(\mathbf{p}(o_i)/\mathbf{q}(o_i)\big)$$

It holds $0 \leq \mathrm{D}(\mathbf{p} \,||\, \mathbf{q}) \leq \infty$, and $\mathrm{D}(\mathbf{p} \,||\, \mathbf{q}) = 0$ iff $\mathbf{p} = \mathbf{q}$.

- Established measure in statistics for comparing distributions.
- Information-theoretic: the average amount of information discriminating $\mathbf{p}$ from $\mathbf{q}$.

Cross entropy

An information-theoretic “distance” on distributions

Cross entropy of distributions $\mathbf{p}, \mathbf{q} : \{o_1, \ldots, o_m\} \to [0, 1]$.

$$\mathrm{D}(\mathbf{p} \mid\mid \mathbf{q}) = \sum_{i=1}^{m} \mathbf{p}(o_i) \cdot \log\big(\mathbf{p}(o_i)/\mathbf{q}(o_i)\big)$$

It holds $0 \leq \mathrm{D}(\mathbf{p} \mid\mid \mathbf{q}) \leq \infty$, and $\mathrm{D}(\mathbf{p} \mid\mid \mathbf{q}) = 0$ iff $\mathbf{p} = \mathbf{q}$.

- Established measure in statistics for comparing distributions.
- Information-theoretic: the average amount of information discriminating $\mathbf{p}$ from $\mathbf{q}$.

Expected cross entropy

A measure on probabilistic trust algorithms

- Goal of a probabilistic trust algorithm $\mathcal{A}$: given a history $\mathbf{X}$, approximate a distribution on the outcomes $O = \{o_1, \ldots, o_m\}$.
- Different histories $\mathbf{X}$ result in different output distributions $\mathcal{A}(\cdot \mid \mathbf{X})$.

Expected cross entropy from λ to $\mathcal{A}$

$$\mathrm{ED}^n(\lambda \,||\, \mathcal{A}) = \sum_{\mathbf{X} \in O^n} Prob(\mathbf{X} \mid \lambda) \cdot \mathrm{D}(Prob(\cdot \mid \mathbf{X}\,\lambda) \,||\, \mathcal{A}(\cdot \mid \mathbf{X}))$$

Expected cross entropy

A measure on probabilistic trust algorithms

- Goal of a probabilistic trust algorithm $\mathcal{A}$: given a history $\mathbf{X}$, approximate a distribution on the outcomes $O = \{o_1, \ldots, o_m\}$.
- Different histories $\mathbf{X}$ result in different output distributions $\mathcal{A}(\cdot \mid \mathbf{X})$.

Expected cross entropy from $\boldsymbol{\lambda}$ to $\mathcal{A}$

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

An application of cross entropy

A parametric algorithm $\mathcal{A}_\epsilon$

$$\mathcal{A}_\epsilon(\mathbf{s} \mid h) = \frac{\mathrm{N}_{\mathbf{s}}(h) + \epsilon}{|h| + 2\epsilon}, \qquad \mathcal{A}_\epsilon(\mathbf{f} \mid h) = \frac{\mathrm{N}_{\mathbf{f}}(h) + \epsilon}{|h| + 2\epsilon}$$

Theorem

For any $\theta \in [0,1]$, $\theta \neq 1/2$ there exists $\bar{\epsilon} \in [0,\infty)$ that minimises $\mathrm{ED}^n(\lambda_\beta \mid\mid \mathcal{A}_\epsilon)$, simultaneously for all n.

Furthermore, $\mathrm{ED}^n(\lambda_\beta \mid\mid \mathcal{A}_\epsilon)$ is a decreasing function of ϵ on the interval $(0,\bar{\epsilon})$, and increasing on $(\bar{\epsilon},\infty)$.

An application of cross entropy

A parametric algorithm $\mathcal{A}_\epsilon$

$$\mathcal{A}_\epsilon(\mathbf{s} \mid h) = \frac{\mathrm{N}_{\mathbf{s}}(h) + \epsilon}{|h| + 2\epsilon}, \qquad \mathcal{A}_\epsilon(\mathbf{f} \mid h) = \frac{\mathrm{N}_{\mathbf{f}}(h) + \epsilon}{|h| + 2\epsilon}$$

Theorem

For any $\theta \in [0,1]$, $\theta \neq 1/2$ there exists $\bar{\epsilon} \in [0,\infty)$ that minimises $\mathrm{ED}^n(\boldsymbol{\lambda}_\beta \mid\mid \mathcal{A}_\epsilon)$, simultaneously for all n.

Furthermore, $\mathrm{ED}^n(\boldsymbol{\lambda}_\beta \mid\mid \mathcal{A}_\epsilon)$ is a decreasing function of ϵ on the interval $(0,\bar{\epsilon})$, and increasing on $(\bar{\epsilon},\infty)$.

An application of cross entropy

A parametric algorithm $\mathcal{A}_\epsilon$

$$\mathcal{A}_\epsilon(\mathbf{s} \mid h) = \frac{\mathrm{N}_{\mathbf{s}}(h) + \epsilon}{|h| + 2\epsilon}, \qquad \mathcal{A}_\epsilon(\mathbf{f} \mid h) = \frac{\mathrm{N}_{\mathbf{f}}(h) + \epsilon}{|h| + 2\epsilon}$$

Theorem

For any $\theta \in [0,1]$, $\theta \neq 1/2$ there exists $\bar{\epsilon} \in [0,\infty)$ that minimises $\mathrm{ED}^n(\boldsymbol{\lambda}_{\boldsymbol{\beta}} \mid\mid \mathcal{A}_\epsilon)$, simultaneously for all n.

Furthermore, $\mathrm{ED}^n(\boldsymbol{\lambda}_{\boldsymbol{\beta}} \mid\mid \mathcal{A}_\epsilon)$ is a decreasing function of ϵ on the interval $(0,\bar{\epsilon})$, and increasing on $(\bar{\epsilon},\infty)$.

That is, unless behaviour is completely unbiased, there exists a unique best $\mathcal{A}_\epsilon$ algorithm that for all n outperforms all the others.
If $\theta = 1/2$, the larger the ϵ, the better.

An application of cross entropy

A parametric algorithm $\mathcal{A}_\epsilon$

$$\mathcal{A}_\epsilon(\mathbf{s} \mid h) = \frac{\mathrm{N}_\mathbf{s}(h) + \epsilon}{|h| + 2\epsilon}, \qquad \mathcal{A}_\epsilon(\mathbf{f} \mid h) = \frac{\mathrm{N}_\mathbf{f}(h) + \epsilon}{|h| + 2\epsilon}$$

Theorem

For any $\theta \in [0,1]$, $\theta \neq 1/2$ there exists $\bar{\epsilon} \in [0, \infty)$ that minimises $\mathrm{ED}^n(\boldsymbol{\lambda}_\beta \mid\mid \mathcal{A}_\epsilon)$, simultaneously for all n.

Furthermore, $\mathrm{ED}^n(\boldsymbol{\lambda}_\beta \mid\mid \mathcal{A}_\epsilon)$ is a decreasing function of ϵ on the interval $(0, \bar{\epsilon})$, and increasing on $(\bar{\epsilon}, \infty)$.

- Algorithm $\mathcal{A}_0$ is optimal for $\theta = 0$ and for $\theta = 1$.
- Algorithm $\mathcal{A}_1$ is optimal for $\theta = \frac{1}{2} \pm \frac{1}{\sqrt{12}}$.

Outline

A trust model based on event structures

Move from $O = \{\mathbf{s}, \mathbf{f}\}$ to complex outcomes

Interactions and protocols

- At an abstract level, entities in a distributed system interact according to protocols;
- Information about an external entity is just information about (the outcome of) a number of (past) protocol runs with that entity.

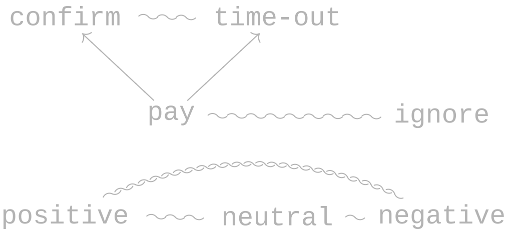
Events as model of information

- A protocol can be specified as a concurrent process, at different levels of abstractions.
- Event structures were invented to give formal semantics to truely concurrent processes, expressing "causation" and "conflict."

A model for behavioural information

- $ES = (E, \leq, \#)$, with E a set of events, $\leq$ and $\#$ relations on E.
- Information about a session is a finite set of events $x \subseteq E$, called a configuration (which is 'conflict-free' and 'causally-closed').
- Information about several interactions is a sequence of outcomes $h = x_1 x_2 \cdots x_n \in \mathcal{C}_{ES}^*$, called a history.

eBay (simplified) example:

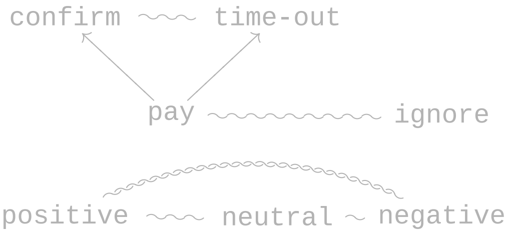


e.g., $h = \{\texttt{pay}, \texttt{confirm}, \texttt{pos}\}\ \{\texttt{pay}, \texttt{confirm}, \texttt{neu}\}\ \{\texttt{pay}\}$

A model for behavioural information

- $ES = (E, \leq, \#)$, with E a set of events, $\leq$ and $\#$ relations on E.
- Information about a session is a finite set of events $x \subseteq E$, called a configuration (which is ‘conflict-free’ and ‘causally-closed’).
- Information about several interactions is a sequence of outcomes $h = x_1 x_2 \cdots x_n \in \mathcal{C}_{ES}^{*}$, called a history.

eBay (simplified) example:

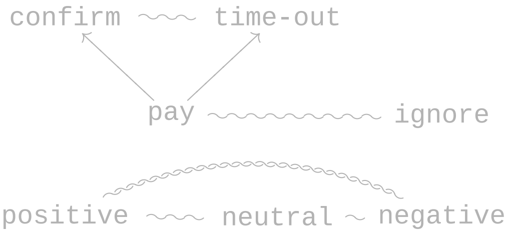


e.g., $h = \{\texttt{pay}, \texttt{confirm}, \texttt{pos}\}\ \{\texttt{pay}, \texttt{confirm}, \texttt{neu}\}\ \{\texttt{pay}\}$

A model for behavioural information

- $ES = (E, \leq, \#)$, with E a set of events, $\leq$ and $\#$ relations on E.
- Information about a session is a finite set of events $x \subseteq E$, called a configuration (which is 'conflict-free' and 'causally-closed').
- Information about several interactions is a sequence of outcomes $h = x_1 x_2 \cdots x_n \in \mathcal{C}_{ES}^*$, called a history.

eBay (simplified) example:

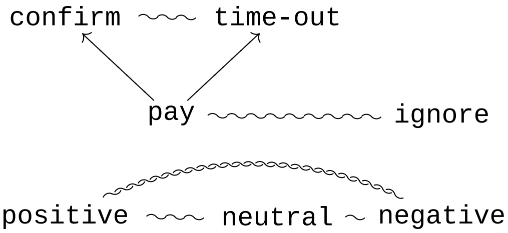


e.g., $h = \{\texttt{pay}, \texttt{confirm}, \texttt{pos}\}\ \{\texttt{pay}, \texttt{confirm}, \texttt{neu}\}\ \{\texttt{pay}\}$

A model for behavioural information

- $ES = (E, \leq, \#)$, with E a set of events, $\leq$ and $\#$ relations on E.
- Information about a session is a finite set of events $x \subseteq E$, called a configuration (which is 'conflict-free' and 'causally-closed').
- Information about several interactions is a sequence of outcomes $h = x_1 x_2 \cdots x_n \in \mathcal{C}_{ES}^*$, called a history.

eBay (simplified) example:

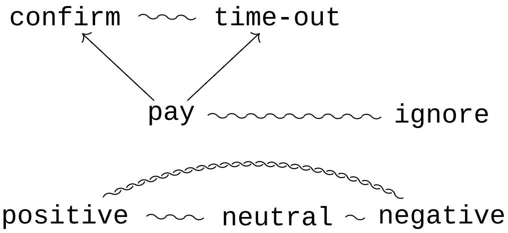


e.g., $h = \{\texttt{pay}, \texttt{confirm}, \texttt{pos}\}\ \{\texttt{pay}, \texttt{confirm}, \texttt{neu}\}\ \{\texttt{pay}\}$

A model for behavioural information

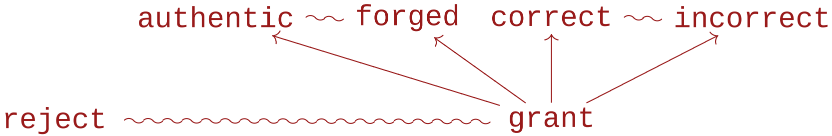
- $ES = (E, \leq, \#)$, with E a set of events, $\leq$ and $\#$ relations on E.
- Information about a session is a finite set of events $x \subseteq E$, called a configuration (which is 'conflict-free' and 'causally-closed').
- Information about several interactions is a sequence of outcomes $h = x_1 x_2 \cdots x_n \in \mathcal{C}^*_{ES}$, called a history.

eBay (simplified) example:



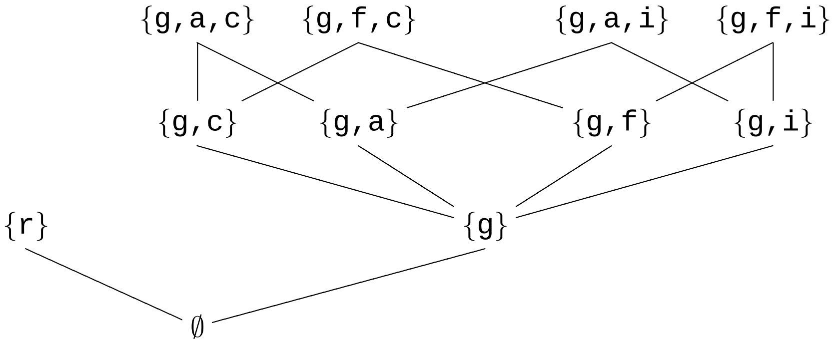
e.g., $h = \{\texttt{pay}, \texttt{confirm}, \texttt{pos}\}\ \{\texttt{pay}, \texttt{confirm}, \texttt{neu}\}\ \{\texttt{pay}\}$

Running example: interactions over an e-purse



Modelling outcomes and behaviour

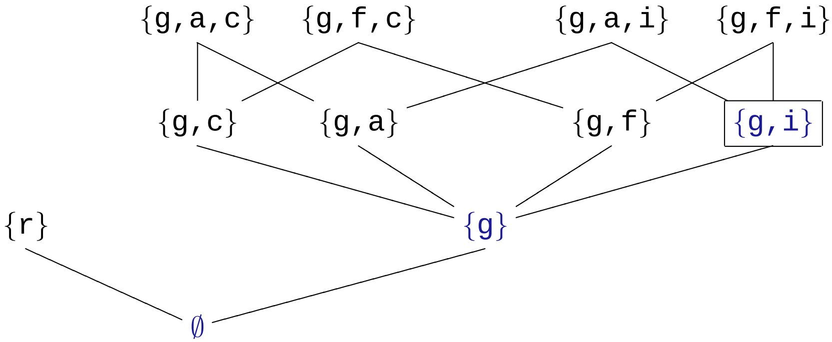
- Outcomes are (maximal) configurations
- The e-purse example:



- Behaviour is a sequence of outcomes

Modelling outcomes and behaviour

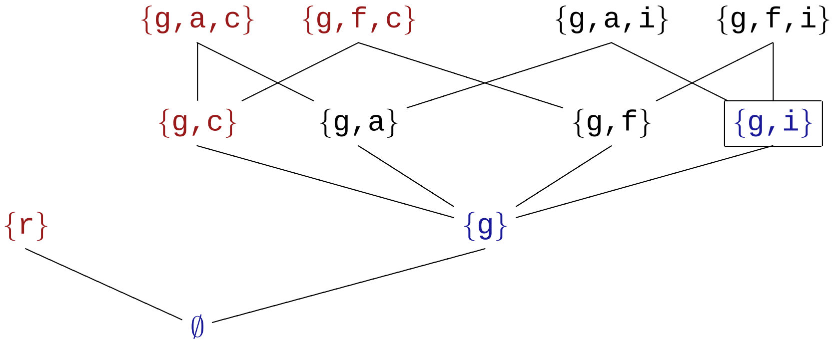
- Outcomes are (maximal) configurations
- The e-purse example:



- Behaviour is a sequence of outcomes

Modelling outcomes and behaviour

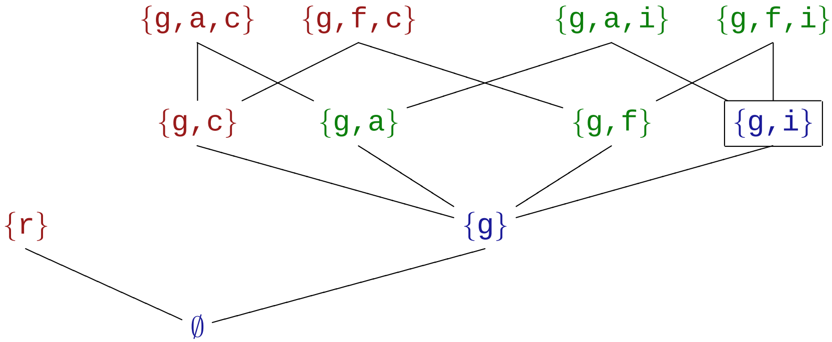
- Outcomes are (maximal) configurations
- The e-purse example:



- Behaviour is a sequence of outcomes

Modelling outcomes and behaviour

- Outcomes are (maximal) configurations
- The e-purse example:



- Behaviour is a sequence of outcomes

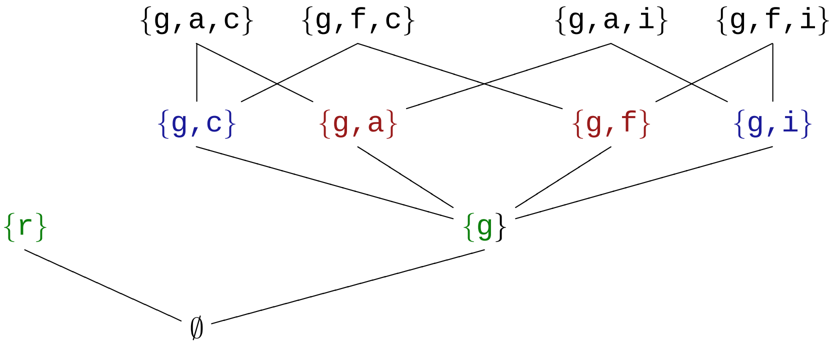
Outline

Confusion-free event structures (Varacca et al)

- Immediate conflict $\#_\mu$: $e \# e'$ and there is x that enables both.
- Confusion free: $\#_\mu$ is transitive and $e \#_\mu e'$ implies $[e) = [e')$.
- Cell: maximal $c \subseteq E$ such that $e, e' \in c$ implies $e \#_\mu e'$.

Confusion-free event structures (Varacca et al)

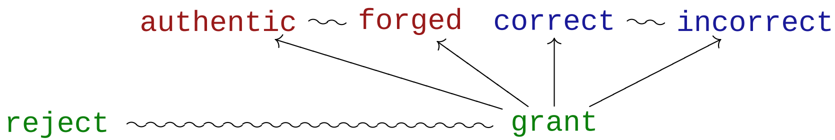
- Immediate conflict $\#_\mu$: $e \# e'$ and there is x that enables both.
- Confusion free: $\#_\mu$ is transitive and $e \#_\mu e'$ implies $[e) = [e')$.
- Cell: maximal $c \subseteq E$ such that $e, e' \in c$ implies $e \#_\mu e'$.



Confusion-free event structures (Varacca et al)

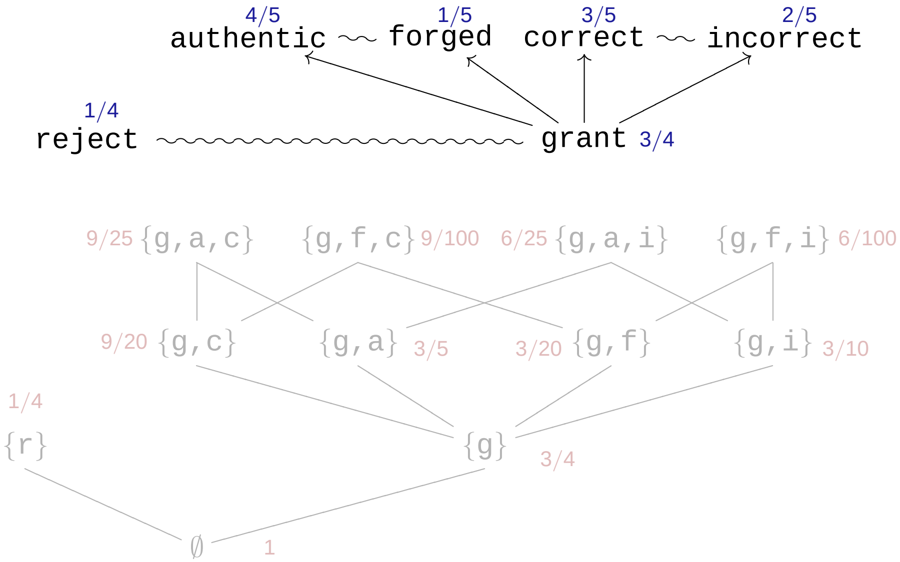
- Immediate conflict $\#_\mu$: $e \# e'$ and there is x that enables both.
- Confusion free: $\#_\mu$ is transitive and $e \#_\mu e'$ implies $[e) = [e')$.
- Cell: maximal $c \subseteq E$ such that $e, e' \in c$ implies $e \#_\mu e'$.

So, there are three cells in the e-purse event structure

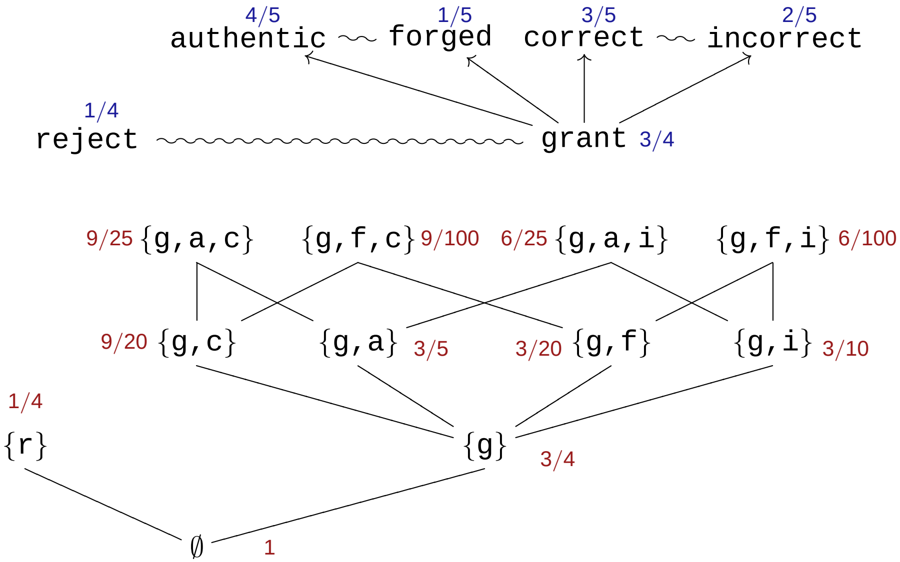


- Cell valuation: a function $p : E \to [0, 1]$ such that $p[c] = 1$, for all c.

Cell valuation



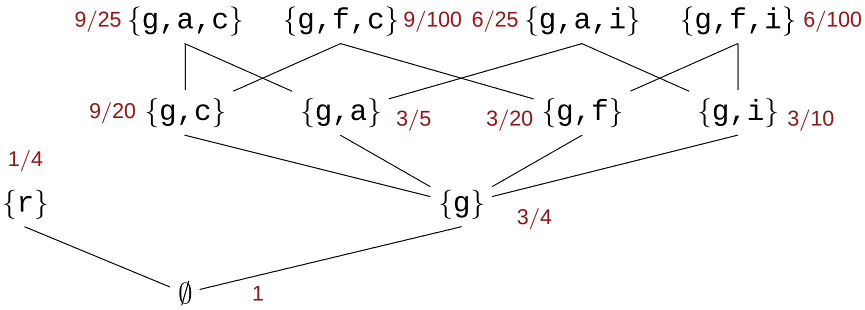
Cell valuation



Properties of cell valuations

Define $p(x) = \prod_{e \in x} p(e)$. Then

- $p(\emptyset) = 1$;
- $p(x) \geq p(x')$ if $x \subseteq x'$;
- p is a probability distribution on maximal configurations.



So, $p(x)$ is the probability that x is contained in the final outcome.

Outline

Estimating cell valuations

How to assign valuations to cells? They are the model's unknowns.

Theorem (Bayes)

$$Prob[\Theta \mid \mathbf{X}\,\lambda] \propto Prob[\mathbf{X} \mid \Theta\,\lambda] \cdot Prob[\Theta \mid \lambda]$$

A second-order notion: we not are interested in **X** or its probability, but in the expected value of **Θ**! So, we will:

- start with a prior hypothesis **Θ**; this will be a cell valuation;
- record the events **X** as they happen during the interactions;
- compute the posterior; this is a new model fitting better with the evidence and allowing us better predictions (in a precise sense).

But: the posteriors need to be (interpretable as) a cell valuations.

Estimating cell valuations

How to assign valuations to cells? They are the model's unknowns.

Theorem (Bayes)

$$Prob[\mathbf{\Theta} \mid \mathbf{X}\, \lambda] \propto Prob[\mathbf{X} \mid \mathbf{\Theta}\, \lambda] \cdot Prob[\mathbf{\Theta} \mid \lambda]$$

A second-order notion: we not are interested in **X** or its probability, but in the expected value of **Θ**! So, we will:

- start with a prior hypothesis **Θ**; this will be a cell valuation;
- record the events **X** as they happen during the interactions;
- compute the posterior; this is a new model fitting better with the evidence and allowing us better predictions (in a precise sense).

But: the posteriors need to be (interpretable as) a cell valuations.

Estimating cell valuations

How to assign valuations to cells? They are the model's unknowns.

Theorem (Bayes)

$$Prob[\Theta \mid X\,\lambda] \propto Prob[X \mid \Theta\,\lambda] \cdot Prob[\Theta \mid \lambda]$$

A second-order notion: we not are interested in X or its probability, but in the expected value of Θ! So, we will:

- start with a prior hypothesis Θ; this will be a cell valuation;
- record the events X as they happen during the interactions;
- compute the posterior; this is a new model fitting better with the evidence and allowing us better predictions (in a precise sense).

But: the posteriors need to be (interpretable as) a cell valuations.

Estimating cell valuations

How to assign valuations to cells? They are the model's unknowns.

Theorem (Bayes)

$$Prob[\mathbf{\Theta} \mid \mathbf{X}\,\lambda] \propto Prob[\mathbf{X} \mid \mathbf{\Theta}\,\lambda] \cdot Prob[\mathbf{\Theta} \mid \lambda]$$

A second-order notion: we not are interested in **X** or its probability, but in the expected value of **Θ**! So, we will:

- start with a prior hypothesis **Θ**; this will be a cell valuation;
- record the events **X** as they happen during the interactions;
- compute the posterior; this is a new model fitting better with the evidence and allowing us better predictions (in a precise sense).

But: the posteriors need to be (interpretable as) a cell valuations.

Estimating cell valuations

How to assign valuations to cells? They are the model's unknowns.

Theorem (Bayes)

$$Prob[\mathbf{\Theta} \mid \mathbf{X}\, \lambda] \propto Prob[\mathbf{X} \mid \mathbf{\Theta}\, \lambda] \cdot Prob[\mathbf{\Theta} \mid \lambda]$$

A second-order notion: we not are interested in $\mathbf{X}$ or its probability, but in the expected value of $\mathbf{\Theta}$! So, we will:

- start with a prior hypothesis $\mathbf{\Theta}$; this will be a cell valuation;
- record the events $\mathbf{X}$ as they happen during the interactions;
- compute the posterior; this is a new model fitting better with the evidence and allowing us better predictions (in a precise sense).

But: the posteriors need to be (interpretable as) a cell valuations.

Estimating cell valuations

How to assign valuations to cells? They are the model's unknowns.

Theorem (Bayes)

$$Prob[\boldsymbol{\Theta} \mid \mathbf{X}\, \lambda] \propto Prob[\mathbf{X} \mid \boldsymbol{\Theta}\, \lambda] \cdot Prob[\boldsymbol{\Theta} \mid \lambda]$$

A second-order notion: we not are interested in $\mathbf{X}$ or its probability, but in the expected value of $\boldsymbol{\Theta}$! So, we will:

- start with a prior hypothesis $\boldsymbol{\Theta}$; this will be a cell valuation;
- record the events $\mathbf{X}$ as they happen during the interactions;
- compute the posterior; this is a new model fitting better with the evidence and allowing us better predictions (in a precise sense).

But: the posteriors need to be (interpretable as) a cell valuations.

Cells vs eventless outcomes

Let $c_1, \ldots, c_M$ be the set of cells of E, with $c_i = \{e^i_1, \ldots, e^i_{K_i}\}$.

- A cell valuation assigns a distribution Θ_{c_i} to each c_i, the same way as an eventless model assigns a distribution θ to $\{\mathbf{s}, \mathbf{f}\}$.
- The occurrence of an x from $\{\mathbf{s}, \mathbf{f}\}$ is a random process with two outcomes, a binomial (Bernoulli) trial on θ.
- The occurrence of an event from cell c_i is a random process with K_i outcomes. That is, a multinomial trial on Θ_{c_i}.

To exploit this analogy we only need to lift the λ_β model to a model based on multinomial experiments.

Cells vs eventless outcomes

Let $c_1, \ldots, c_M$ be the set of cells of E, with $c_i = \{e^i_1, \ldots, e^i_{K_i}\}$.

- A cell valuation assigns a distribution $\mathbf{\Theta}_{c_i}$ to each c_i, the same way as an eventless model assigns a distribution θ to $\{\mathbf{s}, \mathbf{f}\}$.
- The occurrence of an x from $\{\mathbf{s}, \mathbf{f}\}$ is a random process with two outcomes, a binomial (Bernoulli) trial on θ.
- The occurrence of an event from cell c_i is a random process with K_i outcomes. That is, a multinomial trial on $\mathbf{\Theta}_{c_i}$.

To exploit this analogy we only need to lift the λ_β model to a model based on multinomial experiments.

Cells vs eventless outcomes

Let $c_1, \ldots, c_M$ be the set of cells of E, with $c_i = \{e^i_1, \ldots, e^i_{K_i}\}$.

- A cell valuation assigns a distribution $\boldsymbol{\Theta}_{c_i}$ to each c_i, the same way as an eventless model assigns a distribution θ to $\{\mathbf{s}, \mathbf{f}\}$.
- The occurrence of an x from $\{\mathbf{s}, \mathbf{f}\}$ is a random process with two outcomes, a binomial (Bernoulli) trial on θ.
- The occurrence of an event from cell c_i is a random process with K_i outcomes. That is, a multinomial trial on $\boldsymbol{\Theta}_{c_i}$.

To exploit this analogy we only need to lift the λ_β model to a model based on multinomial experiments.

Cells vs eventless outcomes

Let $c_1, \ldots, c_M$ be the set of cells of E, with $c_i = \{e^i_1, \ldots, e^i_{K_i}\}$.

- A cell valuation assigns a distribution $\mathbf{\Theta}_{c_i}$ to each c_i, the same way as an eventless model assigns a distribution θ to $\{\mathbf{s}, \mathbf{f}\}$.
- The occurrence of an x from $\{\mathbf{s}, \mathbf{f}\}$ is a random process with two outcomes, a binomial (Bernoulli) trial on θ.
- The occurrence of an event from cell c_i is a random process with K_i outcomes. That is, a multinomial trial on $\mathbf{\Theta}_{c_i}$.

To exploit this analogy we only need to lift the $\boldsymbol{\lambda}_\beta$ model to a model based on multinomial experiments.

A bit of magic: the Dirichlet probability distribution

The Dirichlet family $\mathcal{D}(\boldsymbol{\Theta} \mid \boldsymbol{\alpha}) \propto \prod \Theta_1^{\alpha_1 - 1} \cdots \Theta_K^{\alpha_K - 1}$

Theorem

The Dirichlet family is a conjugate prior for multinomial trials. That is, if

- *Prob*$[\boldsymbol{\Theta} \mid \lambda]$ *is* $\mathcal{D}(\boldsymbol{\Theta} \mid \alpha_1, ..., \alpha_K)$ *and*
- *Prob*$[\mathbf{X} \mid \boldsymbol{\Theta}\, \lambda]$ *follows the law of multinomial trials* $\Theta_1^{n_1} \cdots \Theta_K^{n_K}$,

then Prob$[\boldsymbol{\Theta} \mid \mathbf{X}\, \lambda]$ *is* $\mathcal{D}(\boldsymbol{\Theta} \mid \alpha_1 + n_1, ..., \alpha_K + n_K)$ *according to Bayes.*

So, we start with a family $\mathcal{D}(\boldsymbol{\Theta}_{c_i} \mid \boldsymbol{\alpha}_{c_i})$, and then use multinomial trials $\mathbf{X} : E \to \omega$ to keep updating the valuation as $\mathcal{D}(\boldsymbol{\Theta}_{c_i} \mid \boldsymbol{\alpha}_{c_i} + \mathbf{X}_{c_i})$.

A bit of magic: the Dirichlet probability distribution

The Dirichlet family $\mathcal{D}(\boldsymbol{\Theta} \mid \boldsymbol{\alpha}) \propto \prod \Theta_1^{\alpha_1 - 1} \cdots \Theta_K^{\alpha_K - 1}$

Theorem

The Dirichlet family is a conjugate prior for multinomial trials. That is, if

- *Prob*$[\boldsymbol{\Theta} \mid \boldsymbol{\lambda}]$ *is* $\mathcal{D}(\boldsymbol{\Theta} \mid \alpha_1, \ldots, \alpha_K)$ *and*
- *Prob*$[\mathbf{X} \mid \boldsymbol{\Theta}\, \boldsymbol{\lambda}]$ *follows the law of multinomial trials* $\Theta_1^{n_1} \cdots \Theta_K^{n_K}$,

then *Prob*$[\boldsymbol{\Theta} \mid \mathbf{X}\, \boldsymbol{\lambda}]$ *is* $\mathcal{D}(\boldsymbol{\Theta} \mid \alpha_1 + n_1, \ldots, \alpha_K + n_K)$ *according to Bayes.*

So, we start with a family $\mathcal{D}(\boldsymbol{\Theta}_{c_i} \mid \boldsymbol{\alpha}_{c_i})$, and then use multinomial trials $\mathbf{X} : E \rightarrow \omega$ to keep updating the valuation as $\mathcal{D}(\boldsymbol{\Theta}_{c_i} \mid \boldsymbol{\alpha}_{c_i} + \mathbf{X}_{c_i})$.

A bit of magic: the Dirichlet probability distribution

The Dirichlet family $\mathcal{D}(\boldsymbol{\Theta} \mid \boldsymbol{\alpha}) \propto \prod \Theta_1^{\alpha_1 - 1} \cdots \Theta_K^{\alpha_K - 1}$

Theorem

The Dirichlet family is a conjugate prior for multinomial trials. That is, if

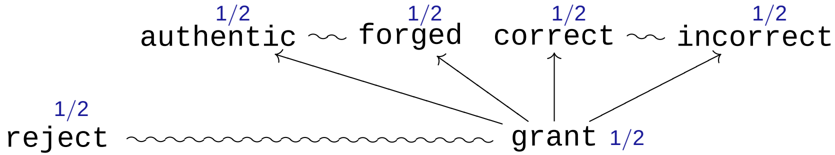
- *$Prob[\boldsymbol{\Theta} \mid \boldsymbol{\lambda}]$ is $\mathcal{D}(\boldsymbol{\Theta} \mid \alpha_1, ..., \alpha_K)$ and*
- *$Prob[\mathbf{X} \mid \boldsymbol{\Theta}\, \boldsymbol{\lambda}]$ follows the law of multinomial trials $\Theta_1^{n_1} \cdots \Theta_K^{n_K}$,*

then $Prob[\boldsymbol{\Theta} \mid \mathbf{X}\, \boldsymbol{\lambda}]$ is $\mathcal{D}(\boldsymbol{\Theta} \mid \alpha_1 + n_1, ..., \alpha_K + n_K)$ according to Bayes.

So, we start with a family $\mathcal{D}(\boldsymbol{\Theta}_{c_i} \mid \boldsymbol{\alpha}_{c_i})$, and then use multinomial trials $\mathbf{X} : E \to \omega$ to keep updating the valuation as $\mathcal{D}(\boldsymbol{\Theta}_{c_i} \mid \boldsymbol{\alpha}_{c_i} + \mathbf{X}_{c_i})$.

The Bayesian process

Start with a uniform distribution for each cell.



Theorem

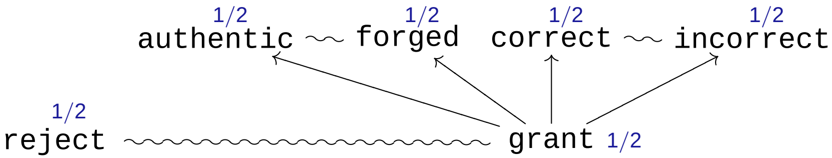
$$E[\Theta_{e^i_j} \mid \mathbf{X}\,\lambda] = \frac{\alpha_{e^i_j} + \mathbf{X}(e^i_j)}{\sum_{k=1}^{K_i}(\alpha_{e^i_k} + \mathbf{X}(e^i_k))}$$

Corollary

$$E[\textit{next outcome is } x \mid \mathbf{X}\,\lambda] = \prod_{e \in x} E[\Theta_e \mid \mathbf{X}\,\lambda]$$

The Bayesian process

Suppose that $\mathbf{X} = \{r \mapsto 2, g \mapsto 8, a \mapsto 7, f \mapsto 1, c \mapsto 3, i \mapsto 5\}$. Then



Theorem

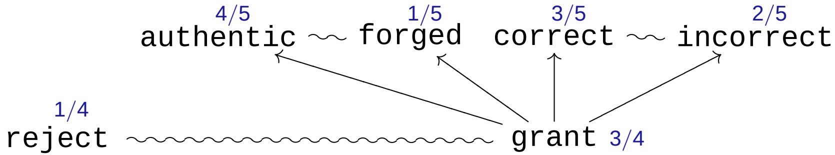
$$E[\Theta_{e^i_j} \mid \mathbf{X}\,\lambda] = \frac{\alpha_{e^i_j} + \mathbf{X}(e^i_j)}{\sum_{k=1}^{K_i}(\alpha_{e^i_k} + \mathbf{X}(e^i_k))}$$

Corollary

$$E[\textit{next outcome is } x \mid \mathbf{X}\,\lambda] = \prod_{e \in x} E[\Theta_e \mid \mathbf{X}\,\lambda]$$

The Bayesian process

Suppose that $\mathbf{X} = \{r \mapsto 2, g \mapsto 8, a \mapsto 7, f \mapsto 1, c \mapsto 3, i \mapsto 5\}$. Then



Theorem

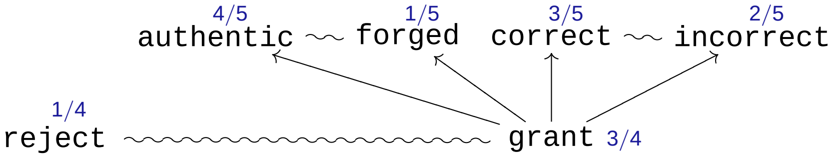
$$E[\Theta_{e^i_j} \mid \mathbf{X}\,\lambda] = \frac{\alpha_{e^i_j} + \mathbf{X}(e^i_j)}{\sum_{k=1}^{K_i} (\alpha_{e^i_k} + \mathbf{X}(e^i_k))}$$

Corollary

$$E[\textit{next outcome is } x \mid \mathbf{X}\,\lambda] = \prod_{e \in x} E[\Theta_e \mid \mathbf{X}\,\lambda]$$

The Bayesian process

Suppose that $\mathbf{X} = \{r \mapsto 2, g \mapsto 8, a \mapsto 7, f \mapsto 1, c \mapsto 3, i \mapsto 5\}$. Then



Theorem

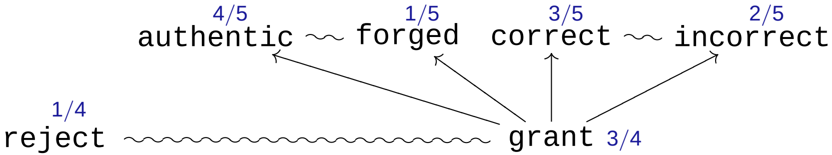
$$E[\Theta_{e^i_j} \mid \mathbf{X}\,\lambda] = \frac{\alpha_{e^i_j} + \mathbf{X}(e^i_j)}{\sum_{k=1}^{K_i}(\alpha_{e^i_k} + \mathbf{X}(e^i_k))}$$

Corollary

$$E[\textit{next outcome is } x \mid \mathbf{X}\,\lambda] = \prod_{e \in x} E[\Theta_e \mid \mathbf{X}\,\lambda]$$

The Bayesian process

Suppose that $\mathbf{X} = \{r \mapsto 2, g \mapsto 8, a \mapsto 7, f \mapsto 1, c \mapsto 3, i \mapsto 5\}$. Then



Theorem

$$E[\Theta_{e^i_j} \mid \mathbf{X}\,\lambda] = \frac{\alpha_{e^i_j} + \mathbf{X}(e^i_j)}{\sum_{k=1}^{K_i}(\alpha_{e^i_k} + \mathbf{X}(e^i_k))}$$

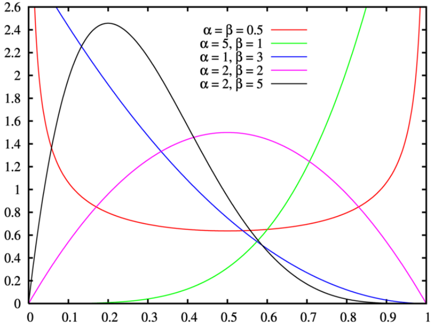
Corollary

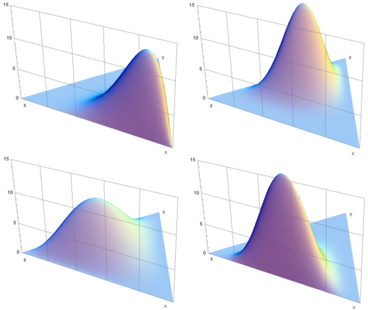
$$E[\textit{next outcome is } x \mid \mathbf{X}\,\lambda] = \prod_{e \in x} E[\Theta_e \mid \mathbf{X}\,\lambda]$$

Interpretation of results

Lifted the trust computational algorithms based on λ_β to our event-base models by replacing

$$\text{Binomials (Bernoulli) trials} \mapsto \text{multinomial trials;}$$
$$\beta\text{-distribution} \mapsto \text{Dirichlet distribution.}$$



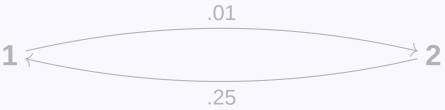


Future directions (1/2)

Hidden Markov Models

Probability parameters can change as the internal state change, probabilistically. HMM is $\lambda = (A, B, \pi)$, where

- A is a Markov chain, describing state transitions;
- B is family of distributions $B_s : O \to [0, 1]$;
- π is the initial state distribution.



$\pi_1 = 1$
$B_1(a) = .95$
$B_1(b) = .05$

$O = \{a, b\}$

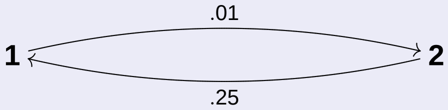
$\pi_2 = 0$
$B_2(a) = .05$
$B_2(b) = .95$

Future directions (1/2)

Hidden Markov Models

Probability parameters can change as the internal state change, probabilistically. HMM is $\lambda = (A, B, \pi)$, where

- A is a Markov chain, describing state transitions;
- B is family of distributions $B_s : O \to [0, 1]$;
- π is the initial state distribution.



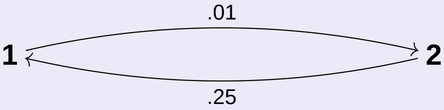
$\pi_1 = 1$
$B_1(a) = .95$
$B_1(b) = .05$

$O = \{a, b\}$

$\pi_2 = 0$
$B_2(a) = .05$
$B_2(b) = .95$

Future directions (2/2)

Hidden Markov Models



$\pi_1 = 1$
$B_1(a) = .95$
$B_1(b) = .05$

$O = \{a, b\}$

$\pi_2 = 0$
$B_2(a) = .05$
$B_2(b) = .95$

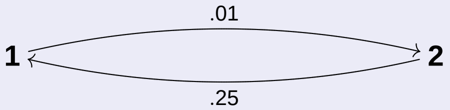
Bayesian analysis:

- What models best explain (and thus predict) observations?
- How to approximate a HMM from a sequence of observations?

History $h = a^{10}b^2$. A counting algorithm would then assign high probability to a occurring next. But he last two b's suggest a state change might have occurred, which would in reality make that probability very low.

Future directions (2/2)

Hidden Markov Models



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

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

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

Bayesian analysis:

- What models best explain (and thus predict) observations?
- How to approximate a HMM from a sequence of observations?

History $h = a^{10}b^2$. A counting algorithm would then assign high probability to a occurring next. But he last two b's suggest a state change might have occurred, which would in reality make that probability very low.

Summary

- A framework for "trust and reputation systems"
  - applications to security and history-based access control.
- Bayesian approach to observations and approximations, formal results based on probability theory. Towards model comparison and complex-outcomes Bayesian model.
- Future work
  - Dynamic models with variable structure.
  - Better integration of reputation in the model.
  - Relationships with game-theoretic models.

Summary

- A framework for "trust and reputation systems"
  - applications to security and history-based access control.
- Bayesian approach to observations and approximations, formal results based on probability theory. Towards model comparison and complex-outcomes Bayesian model.
- Future work
  - Dynamic models with variable structure.
  - Better integration of reputation in the model.
  - Relationships with game-theoretic models.