UNIVERSITY OF
Southampton
School of Electronics
and Computer Science

Exponential Decay in Probabilistic Trust Models

Mogens Nielsen's 60th Fest

Aarhus, 4/10/09

To your sharpness of mind and kindness of heart

Vladimiro Sassone

I dag er det Mogens' fødselsdag

I dag er det Mogens' fødselsdag

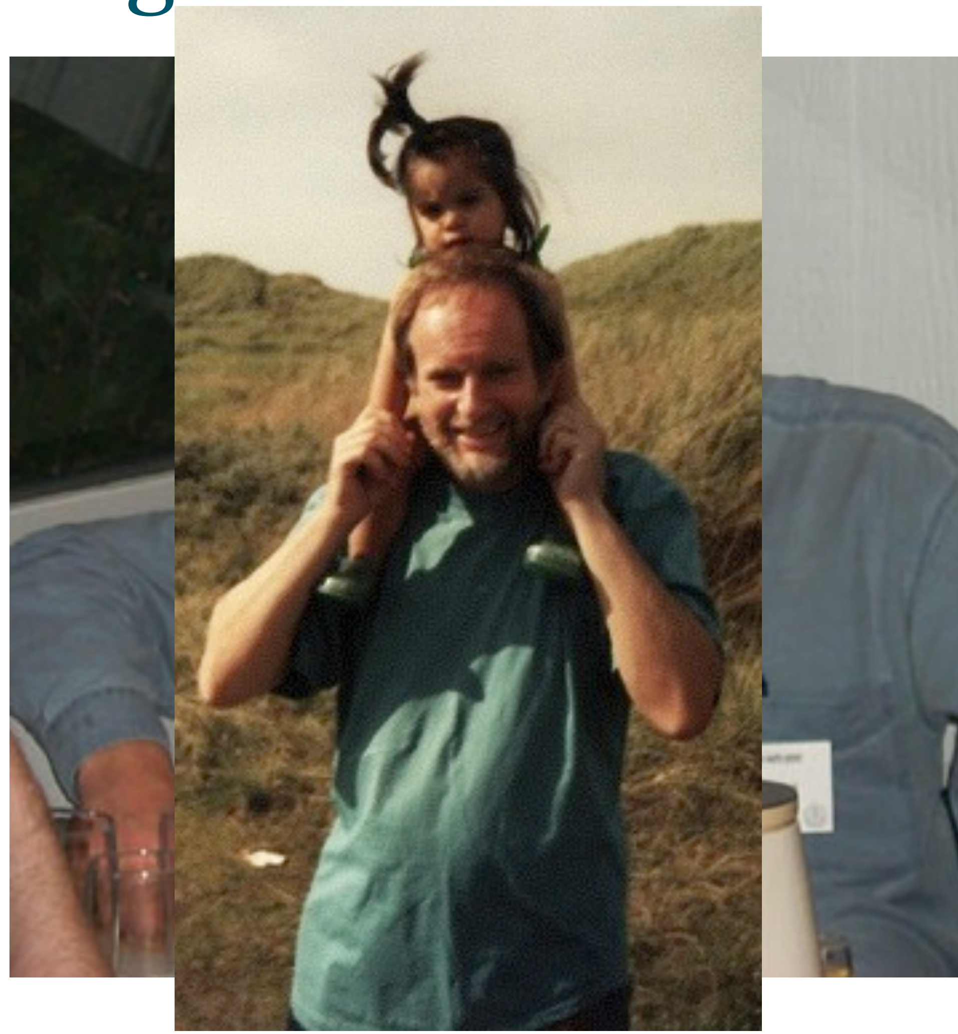
UNIVERSITY OF Southampton
Electronics and Computer Science

I dag er det Mogens' fødselsdag

UNIVERSITY OF Southampton
Electronics and Computer Science

I dag er det Mogens' fødselsdag

I dag er det Mogens' fødselsdag



I dag er det Mogens' fødselsdag

UNIVERSITY OF Southampton
Electronics and Computer Science

I dag er det Mogens' fødselsdag

UNIVERSITY OF Southampton
Electronics and Computer Science

I dag er det Mogens' fødselsdag

UNIVERSITY OF Southampton
Electronics and Computer Science

Trust in UbiCom

UNIVERSITY OF Southampton
Electronics and Computer Science

- Features of Ubiquitous Computing like *scalability*, *mobility*, and *incomplete information* deeply affect security requirements.

- One of the proposed approaches is to use a notion of *computational trust*, resembling the concept of trust among human beings.

Probabilistic Trust

- *Probabilistic models* are used to evaluate trust.
- A probabilistic model assigns a degree of confidence to a principal's ability to predict another principal's behaviour.
- Eg, the behaviour of a principal *A* may be defined as the probability that interaction with *A* yields a certain outcome (eg, success or failure).

Beta Trust Model

- The outcome of an interaction between a principal a *and* a partner b is either *successful* or *unsuccessful:*

$$o \in \{Succ,\ Fail\}$$

- The probability that a partner b interacts successfully with a is governed by the parameter θ where

$$\theta = \Pr(o = Succ)$$

Beta Trust Model

UNIVERSITY OF Southampton
Electronics and Computer Science

- The behaviour of the partner b represented by θ is assumed to be fixed over time.

- The estimated probability of success, $B(Succ\ |o)$, at time t is the expected value of θ *given the sequence of outcomes* $o = \{o_0, o_1, \ldots, o_t\}$

$$B(\,Succ \mid o) = E[\,\theta \mid o\,]$$

Beta Trust Model

UNIVERSITY OF Southampton
Electronics and Computer Science

Using Bayesian inference to learn the parameter θ from observations o.

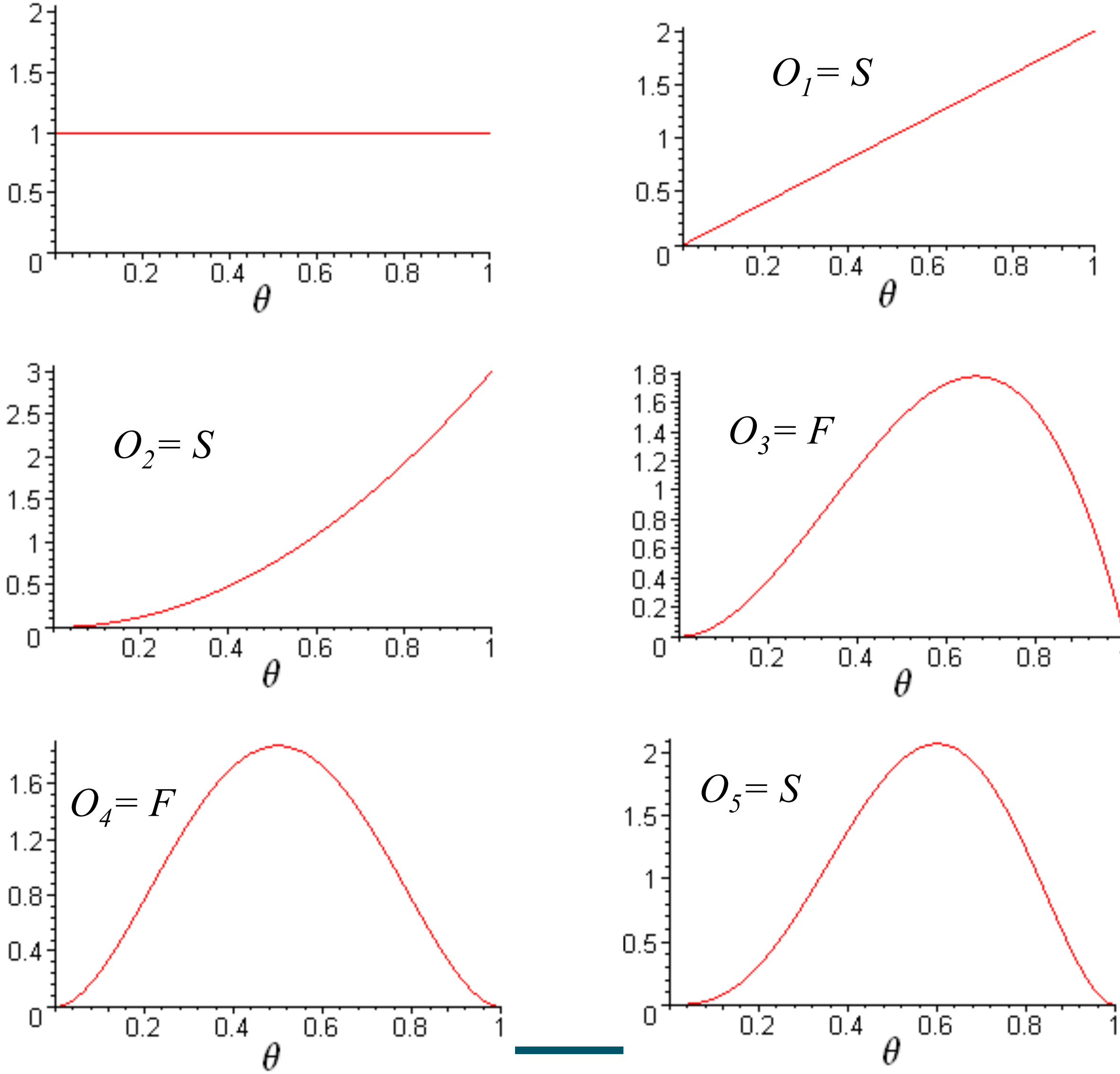
The random variable θ follows Beta distribution function, and therefore

$$B(\,Succ \mid o) = E[\,\theta \mid o\,] = \frac{m(o)+1}{m(o)+n(o)+2}$$

where $m(o)$ is the number of successful interactions in o and $n(o)$ that of unsuccessful ones in o.

Trust Inference Process

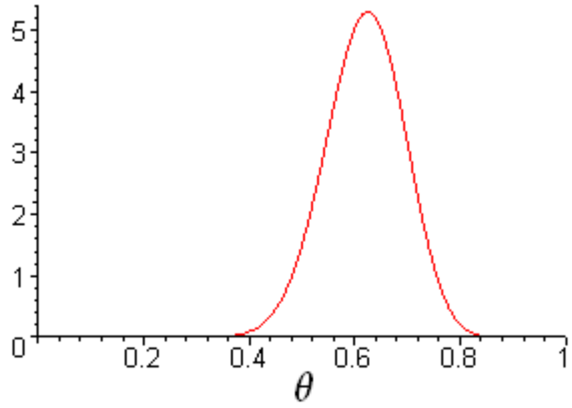
UNIVERSITY OF Southampton
Electronics and Computer Science



Trust Inference Process

UNIVERSITY OF Southampton
Electronics and Computer Science

The distribution of θ after 40 interactions
25 Successful and 15 Failed



UNIVERSITY OF Southampton
Electronics and Computer Science

Limitation of the Beta model

- The assumption that a principal behaviour is fixed is not always realistic:
- The behaviour of a principal may depend on its internal state which may change over time.

Modelling Dynamic Behaviour

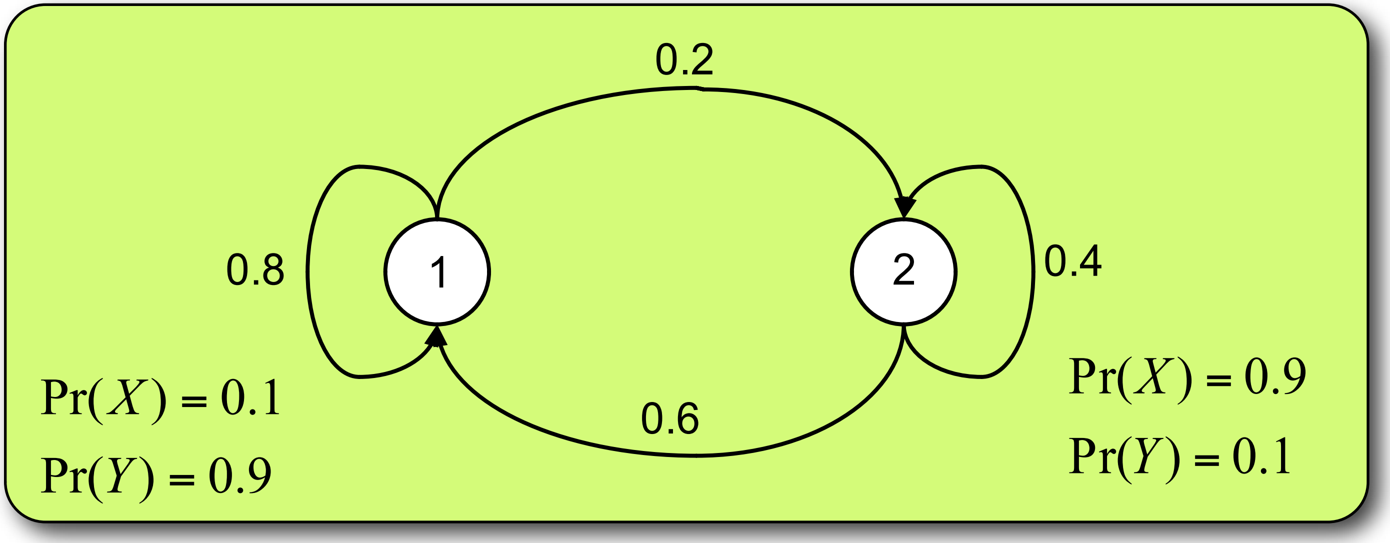
UNIVERSITY OF Southampton
Electronics and Computer Science

- Modelling static behaviour as a probability distribution over outcomes leads to modelling the dynamic behaviour by a *Hidden Markov Model (HMM).*

- A single state in an HMM models the system behaviour at a particular time.

Hidden Markov Model:

UNIVERSITY OF Southampton
Electronics and Computer Science



$$S = \{1,2\}$$

$$V = \{X, Y\}$$

$$A = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} \qquad B = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

Beta Trust Model with Decay

UNIVERSITY OF Southampton
Electronics and Computer Science

- The probability distribution over outcomes changes over time.
- Old observations are given less weight (decayed) than more recent observations.
- Weights of observations are controlled by the decay factor r.

Beta Trust Model with Decay

Given a decay factor $0 \leq r < 1$ and an observation sequence $o=\{o_L,\ldots,o_1,o_0\}$, where o_0 is the last outcome, o_1 is the previous outcome, and so on, then

$$B_r(Succ|O) = \frac{m_r(O)+1}{m_r(O)+n_r(O)+2} \quad , \quad B_r(Fail|O) = \frac{n_r(O)+1}{m_r(O)+n_r(O)+2}$$

Where

$$m_r(O) = \sum_{i=0}^{L} r^i \delta_i(Succ) \quad , \quad n_r(O) = \sum_{i=0}^{L} r^i \delta_i(Fail)$$

And

$$\delta_i(Succ) = \begin{cases} 1 & \text{if } o_i = Succ \\ 0 & \text{otherwise} \end{cases} \quad , \quad \delta_i(Fail) = \begin{cases} 1 & \text{if } o_i = Fail \\ 0 & \text{otherwise} \end{cases}$$

How good is the model ?

UNIVERSITY OF Southampton
Electronics and
Computer Science

- Given a dynamic system modelled by an HMM λ we define Beta estimation error as follows

$$\mathrm{Error}(\lambda, r) = E\left[\,(B(\,Succ \mid o) - \alpha)^2\,\right]$$

where r is the decay factor, and α is the real probability that next outcome is Success

System Stability

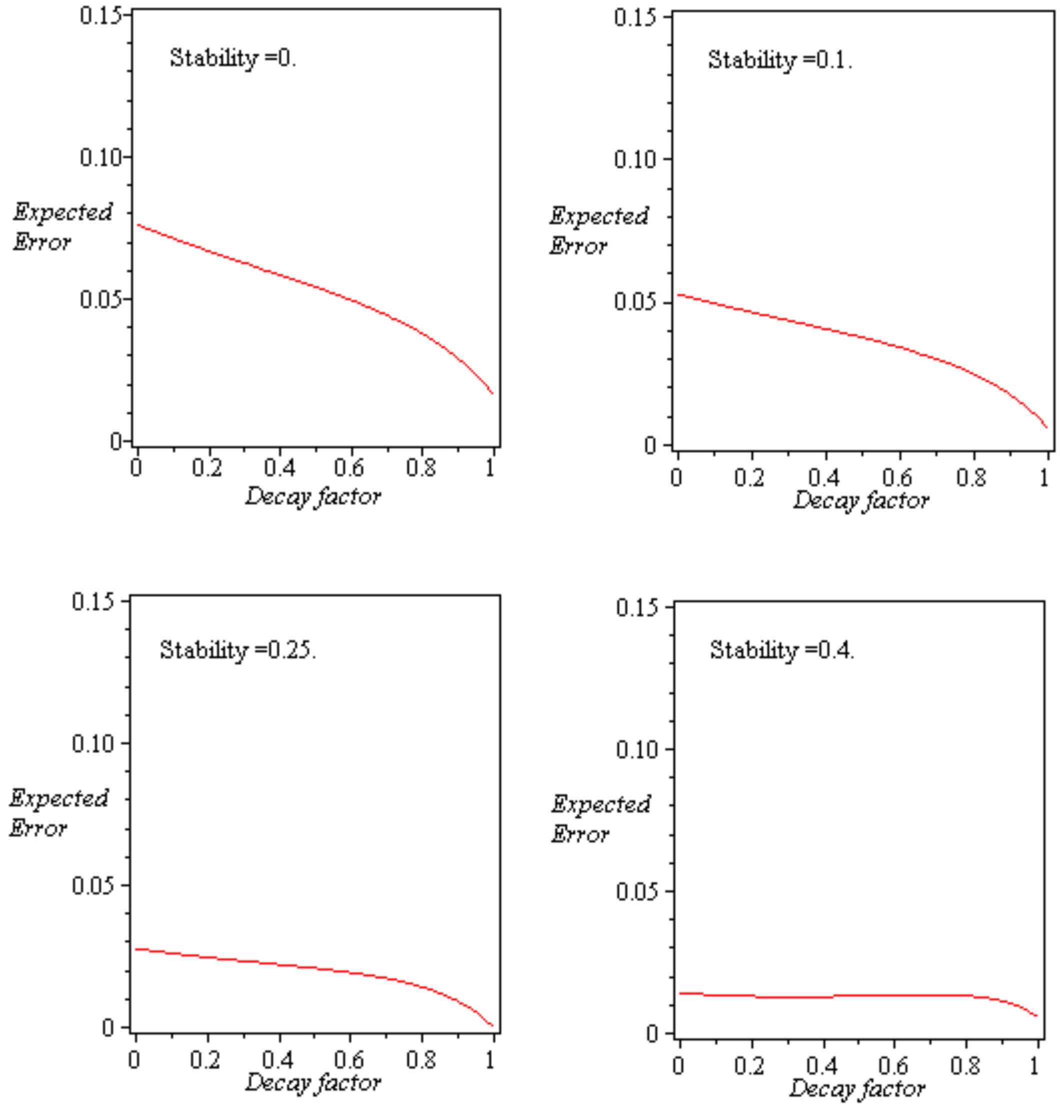
- *System stability* is the expected probability of the HMM remaining in the same state.

- Consider the system modelled by HMM:

$$A_\lambda = \begin{bmatrix} s & \frac{1-s}{3} & \frac{1-s}{3} & \frac{1-s}{3} \\ \frac{1-s}{3} & s & \frac{1-s}{3} & \frac{1-s}{3} \\ \frac{1-s}{3} & \frac{1-s}{3} & s & \frac{1-s}{3} \\ \frac{1-s}{3} & \frac{1-s}{3} & \frac{1-s}{3} & s \end{bmatrix} \qquad \Theta_\lambda = \begin{bmatrix} 1.0 \\ 0.7 \\ 0.3 \\ 0.0 \end{bmatrix}$$

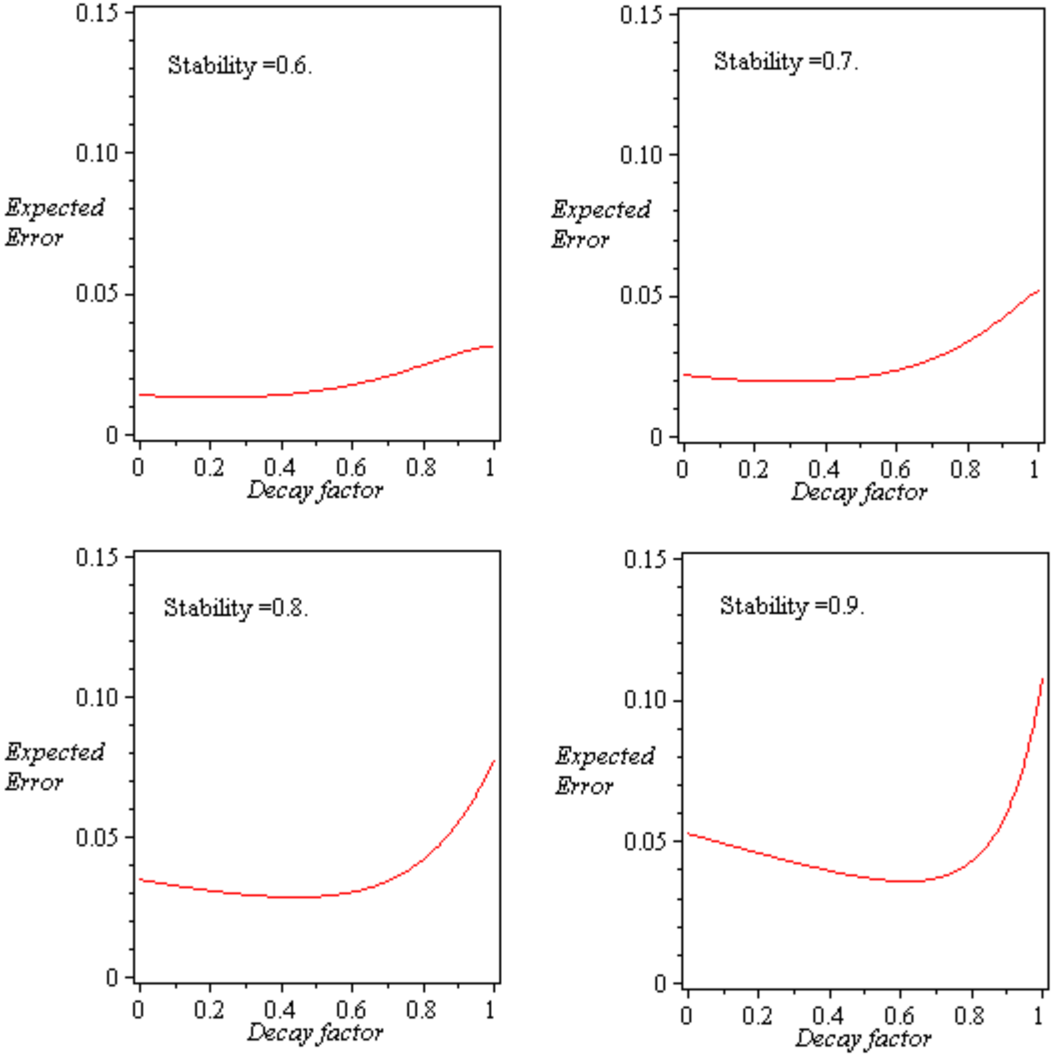
Unstable System

UNIVERSITY OF Southampton
Electronics and Computer Science



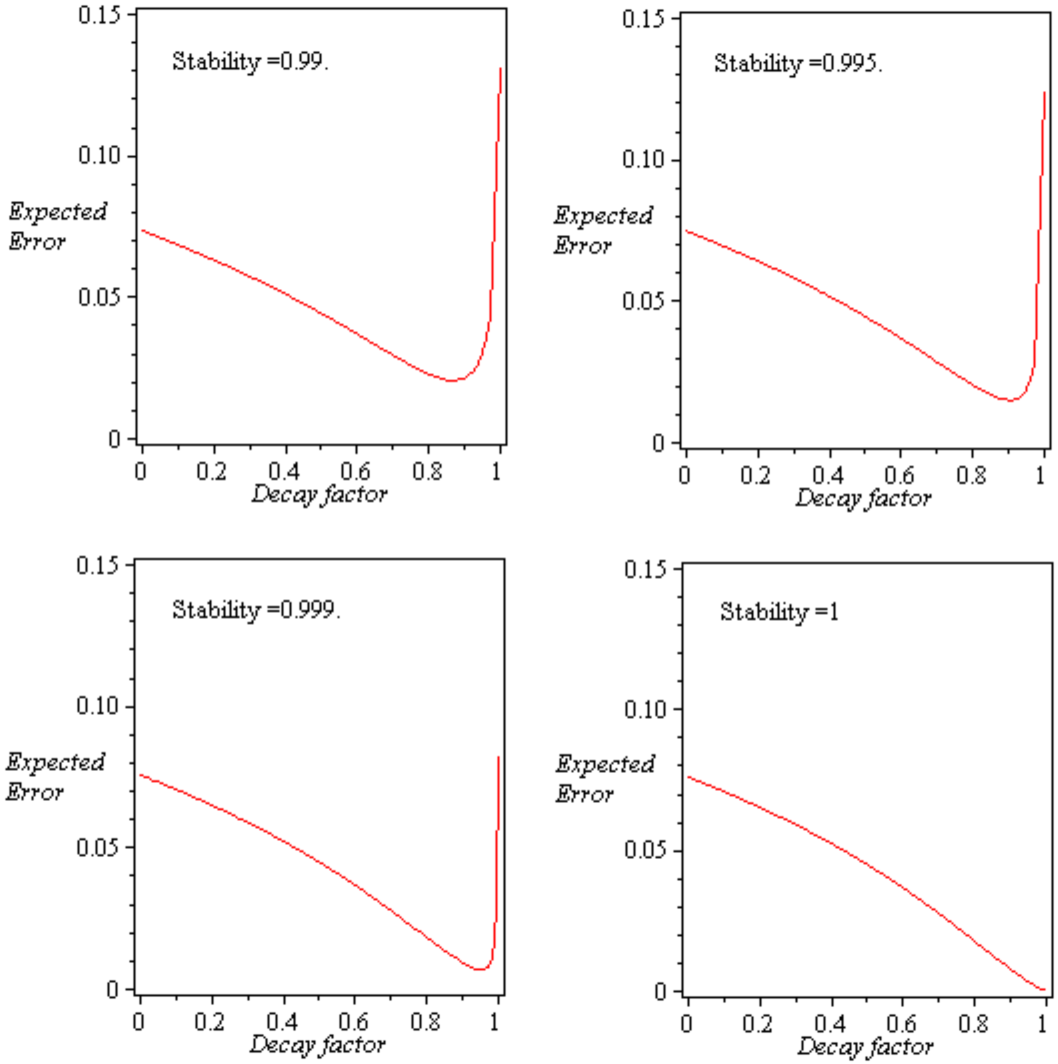
Stable System

University of Southampton
Electronics and Computer Science



Very Stable System

UNIVERSITY OF Southampton
Electronics and Computer Science



Conclusion

UNIVERSITY OF Southampton
Electronics and Computer Science

- Traditional Beta trust models are unable of coping with dynamic behaviour systems.
- Using a decay scheme enhances Beta estimation in cases where the system is very stable.
- Beta estimation error is subject to choosing the optimal value of decay which depends on the system parameters.

Current and Future Work

- Investigate using HMM as a trust model instead of using existing Beta models [FAST 2009 with MN]

- Handling Reputation problems
  - Modelling reputation reports
  - Combining reputation reports to update trust
  - Evaluating confidence in the evaluated trust