UNIVERSITY OF SOUTHAMPTON

Probabilistic trust models in network security

by

Ehab M. ElSalamouny

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

March 2011

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Ehab M. ElSalamouny

One of the dominant properties of a global computing network is the incomplete information available to principals about each other. This was the motivation of using the notion of *probabilistic trust* as an approach to security sensitive decision making in modern open and global computing systems. In such systems any principal A uses the outcomes of past interactions with another principal B to construct a probabilistic model approximating the behaviour of B. Using this model, the principal A can take decisions regarding interactions with B by estimating its future actions. Many existing frameworks adopt the so-called 'Beta model'. The main limitation of these frameworks is that they assume the behaviour of any principal to be fixed, which is not realistic in many cases.

In this thesis, we first address the application of probabilistic trust to optimise security protocols, and specifically give an example where the Crowds anonymity protocol is extended to use trust information. We then address the problem of evaluating probabilistic trust in principals exhibiting dynamic behaviours. In this respect, we formally analyse the 'exponential decay' technique as an approach to coping with principals' dynamic behaviours. Given the identified limitations of this technique, a more general framework for trust and reputation is introduced. In this framework, *Hidden Markov Models* (HMMs) are used for modelling the dynamic behaviours of principals. This framework is formally analysed in terms of a notion of 'estimation error'.

Using an experimental approach based on Monte-Carlo methods to evaluate the expected estimation error, the introduced HMM-based framework for trust and reputation is compared to the existing Beta framework. The results show in general that the latter is getting more promising in evaluating trust in principals ('trustees') having dynamic behaviours as longer sequences of observations are available about such trustees.

Contents

Declaration of Authorship xi

Acknowledgements xiii

1 Introduction 1
- 1.1 Innovation . . . 4
  - 1.1.1 Dynamic behaviour modelling . . . 4
  - 1.1.2 Evaluating the quality of trust models . . . 5
  - 1.1.3 Analysis of the beta trust model with the decay principle . . . 5
  - 1.1.4 HMM-based trust and reputation models . . . 6
  - 1.1.5 Using trust information . . . 6
- 1.2 Thesis structure . . . 7

2 Mathematical background 9
- 2.1 Bayesian inference . . . 9
- 2.2 Parameter estimation . . . 10
  - 2.2.1 Bayesian estimation . . . 11
  - 2.2.2 Maximum Likelihood Estimate (MLE) . . . 11
  - 2.2.3 Expectation-Maximisation algorithm . . . 11
- 2.3 Model selection . . . 13
  - 2.3.1 Akaike Information Criterion (AIC) . . . 14
  - 2.3.2 Bayesian Information Criterion (BIC) . . . 14
- 2.4 Markov chains . . . 15
  - 2.4.1 Irreducibility . . . 15
  - 2.4.2 Recurrence . . . 15
  - 2.4.3 Aperiodicity . . . 17
- 2.5 Hidden Markov Models (HMM) . . . 17
  - 2.5.1 Definition . . . 18
  - 2.5.2 HMM basic problems . . . 18
  - 2.5.3 Forward-Backward algorithm . . . 19
  - 2.5.4 Baum-Welch algorithm . . . 21
  - 2.5.5 Links between HMM and probabilistic automata . . . 24
- 2.6 HMM structure determination . . . 25
  - 2.6.1 HMM structure induction by Baum-Welch algorithm . . . 25
  - 2.6.2 Akaike and Bayesian information criteria . . . 25
  - 2.6.3 HMM induction by Bayesian model merging . . . 26
  - 2.6.4 Designing a minimal HMM structure by bisimulation . . . 27

2.7 Stationarity, mixing and ergodicity . . . 28
2.7.1 Random processes . . . 28
2.7.2 Stationarity . . . 30
2.7.3 Mixing . . . 30
2.7.4 Ergodicity . . . 32

3 State of the art 35
3.1 Credential based trust . . . 35
3.1.1 Network security credentials . . . 36
3.1.2 Centralised trust management systems . . . 37
3.1.3 Decentralised trust management systems . . . 38
3.1.4 Trust negotiation . . . 39
3.1.5 Security policies and trust languages . . . 41
3.2 Probabilistic models for trust . . . 42
3.2.1 Beta trust model . . . 43
3.2.2 Dirichlet reputation model . . . 45
3.2.3 The TRAVOS model . . . 46
3.2.4 Event-based trust model . . . 48
3.2.5 Exponential decay . . . 49

4 Application of probabilistic trust to anonymity 51
4.1 Anonymity protocols . . . 53
4.1.1 Analysis framework and notations . . . 53
4.1.2 Quantifying anonymity . . . 54
4.1.3 The Crowds protocol . . . 55
4.1.4 Probable innocence . . . 55
4.2 Using trust information . . . 58
4.2.1 Crowds protocol extended . . . 58
4.2.2 Probable innocence revisited . . . 59
4.2.3 Provably exposed principals . . . 62
4.3 Achieving probable innocence . . . 63
4.3.1 Probability of forwarding . . . 64
4.3.2 Trust values . . . 65
4.3.3 Forwarding policy. . . . 65
4.4 Discussion . . . 66

5 Estimation error of Beta trust model with a decay scheme 69
5.1 The exponential decay principle . . . 70
5.2 Modelling the real system . . . 71
5.3 Beta model with a decay factor . . . 71
5.4 The error function . . . 72
5.4.1 Analysis of the expected Beta estimation error . . . 73
5.5 System stability . . . 82
5.6 Discussion . . . 88

6 HMM-based trust model 91
6.1 Modelling the real system . . . 93
6.1.1 General and relative behaviour . . . 93

6.2 The estimation error . . . 97
6.2.1 Analysis of the expected estimation error for HMM-based trust model . . . 98
6.3 Comparison with Beta-based trust with decay principle . . . 100
6.3.1 Monte-Carlo based evaluation of the expected estimation error . . . 102
6.3.2 Experiments . . . 103
6.4 Likelihood convergence property . . . 106
6.4.1 Enhancing the convergence using multiple observation sequences . . . 108
6.5 Discussion . . . 111

7 HMM-based reputation model 113
7.1 General assumptions . . . 115
7.2 Reputation framework . . . 116
7.2.1 Deriving a reputation algorithm . . . 118
7.2.2 Reputation mixing . . . 122
7.3 Performance analysis . . . 123
7.4 Experimental evaluation . . . 124
7.4.1 Impact of multiple reputation reports . . . 124
7.4.2 Comparison with Beta reputation model . . . 126

8 Conclusion 129
8.1 Main contributions . . . 130
8.1.1 Using probabilistic trust in CROWDS protocol . . . 130
8.1.2 Analysis of the decay principle . . . 131
8.1.3 HMM-based trust model . . . 131
8.1.4 HMM-based reputation model . . . 132
8.2 Future Work . . . 133
8.2.1 Implementation . . . 133
8.2.2 Handling inaccurate reputation reports . . . 134

Bibliography 135

List of Figures

2.1 The evolution of the posterior pdf for the bias-weighting of a coin 12

3.1 The plot of Dirichlet pdf for different values of the parameter $\boldsymbol{\alpha}$ 47

4.1 A message path in the CROWDS protocol 56

5.1 Expected Beta estimation error versus decay factor given stability < 0.5 . 83
5.2 Expected Beta estimation error versus decay factor given stability > 0.5 . 84
5.3 Expected Beta estimation error versus decay factor given stability > 0.9 . 84
5.4 Expected Beta estimation error for the four-states model 87

6.1 State transition system for Example 6.1 94
6.2 Beta and HMM expected estimation errors versus decay factor given stability < 0.5 104
6.3 Beta and HMM estimation errors versus decay factor given stabilities 0.6, 0.7, 0.8, and 0.9 105
6.4 Beta and HMM estimation errors versus decay factor given stabilities 0.95, 1.0 106
6.5 Convergence of $\frac{1}{T} \log P(h_T \mid \eta)$ to the limit $H(\lambda, \eta)$. 107

7.1 The expected estimation error using the HMM-based reputation model. . 125
7.2 The expected estimation error using HMM and Beta reputation models. . 127

Academic Thesis: Declaration of Authorship

I, Ehab Mohamed ElSalamouny declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

Probabilistic trust models in network security

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Either none of this work has been published before submission, or parts of this work have been published as:

E. ElSalamouny, K. Krukow, and V. Sassone. An analysis of the exponential decay principle in probabilistic trust models. *Theoretical Computer Science*, 410(41): 4067-4084, 2009. ISSN 0304-3975.

E. ElSalamouny, V. Sassone, and M. Nielsen. Hmm-based trust model. In Pierpaolo Degano and Joshua Guttman, editors, *Formal Aspects in Security and Trust*, volume 5983 of *Lecture Notes in Computer Science*, pages 21-35. Springer Berlin/Heidelberg, 2010.

V. Sassone, E. ElSalamouny, and S. Hamadou. Trust in crowds: Probabilistic behaviour in anonymity protocols. In Martin Wirsing, Martin Hofmann, and Axel Rauschmayer, editors, *Trustworthy Global Computing*, volume 6084 of *Lecture Notes in Computer Science*, pages 88-102. Springer Berlin/Heidelberg, 2010.

Signed:

Date:

Acknowledgements

Special thanks to my supervisor Prof. Vladimiro Sassone for his ideas, advices, and support and also for sharing me both joyful and hard times during my studies. Thanks to the Egyptian government for funding my PhD project. Thanks also to Dr. Adam Prugel-Bennett for spending his valuable time discussing difficulties in this work. Thanks to Prof. Catuscia Palamidessi for sharing us our thoughts and ideas, and for her hospitality during my visit to the École Polytechnique in France. Thanks to Prof. Mogens Nielsen for his discussions and contribution to this work and also for his hospitality during my visit to the University of Aarhus in Denmark. Thanks to Dr. Karl Krukow and Dr. Sardaouna Hamadou for their ideas and co-authoring in our publications. Lastly but not least, thanks to my family for their love and support all the way.

Chapter 1

Introduction

In a general network of principals, one important requirement is that a principal can make security decisions regarding interactions with other principals. Such decisions aim at minimising the risk of abusing or destroying resources in an interaction. To make such security critical decisions, a principal needs to assess its confidence that an interaction with a potential partner is subjectively secure. Such confidence is referred to as the 'trust' of the given principal (called 'truster') in its potential partner (called 'trustee'). Therefore, a notion of trust is fundamental to security preserving in a network of principals.

Trust has traditionally been formalised in terms of authorisation, which essentially amounts to determining whether a principal that issues a request to use a resource should be trusted with this request and granted the resource. This approach to trust is often referred to as *policy based trust* or *credential based trust* (cf. Section 3.1). Here the trust in a principal is established by obtaining a sufficient amount of credentials (proofs) pertaining to it. Given these credentials, certain policies are applied to grant that principal certain access rights. The recursive problem of validating the credentials is frequently solved by using a trusted third party to serve as an authority for issuing and verifying credentials. In this setting, trust takes binary values in the sense that a principal is either trusted or not trusted. This relies on the fact that trusted principals are *proved* to behave well and their behaviours are well known to their interactions partners.

However, in modern open-ended networks (e.g. the Internet) principals can have autonomously different behaviours and intentions which are incompletely known by other principals and therefore can affect their security. Moreover, it is not practical in such global networks to use third parties to issue and verify credentials because this again raises the question of whether such parties are always trusted or not. Given these attributes, credential-based trust is not entirely appropriate as a basis of interactions in these networks simply because no principal can be assumed to be perfectly trusted. In

fact there is always a probability that a principal exhibits a misbehaviour and violates the agreed protocol.

In these systems, where each principal always has incomplete information about other principals, interactions of a principal A with other principals are not assumed to be at the same level of satisfaction, or even safety, to A. Thus principal A needs to evaluate a quantitative measure for trust in each other principal B using the history of B's behaviour. This information is obtained from past interactions between A and B and also from B's *reputation* collected from other principals. Based on the output of this trust evaluation process, A can choose its interaction partners. Note that the trust value here is not binary as the case in credential-based trust, but rather is a number expressing the level of trustworthiness. This view is known as the *computational trust* and also as *reputation based trust*.

One approach to implementing the computational trust is the *probabilistic trust*, which can broadly be characterised as aiming to build probabilistic models upon which to base predictions about trustees' future actions. Using these models, the trust of a truster A in a trustee B is the probability, estimated by A, of particular outcomes of the next interaction with B. This notion of trust resembles the trusting relationship between humans as seen by Gambetta (1988).

Many systems for probabilistic trust management assume, sometimes implicitly, the following scenario. There is a collection of principals $(p_i \mid i \in I)$, for some finite index set I, which at various points in time can choose to interact in a pair-wise manner; each interaction can result in one of a predefined set of outcomes, $O = \{o_1, \ldots, o_m\}$. Typically, outcomes are determined by behaviours: when a principal p_i interacts with a partner p_j, the behaviour of p_j relative to the protocol used for interaction defines the outcome. Hence, an essential component in the trust management framework is the *behaviour model* of a principal. Jøsang and Ismail (2002) and Teacy et al. (2006), for example, assumed that the outcome of an interaction between two principals is probabilistically sampled from two potential events (*success* and *failure*). Specifically, compliant behaviours represent successful interactions, whilst behaviours which diverge from the interaction protocol determine failure. Nielsen et al. (2007), generalised the outcome of a single interaction (a protocol run) to be a set of sub-events rather than one event. Each sub-event is assumed to be probabilistically sampled from potential sub-events enabled at a particular stage in the protocol run. Despite these differences, the probabilistic systems share the following characteristics.

- They assume a particular probabilistic model for each principal behaviour.
- They put forward algorithms for approximating (or learning) a principal's behaviour using the history of interaction with the given principal. An interaction history is simply the sequence of the outcomes of past interactions with the principal. One framework used for this purpose is Bayesian analysis, where observations

(interaction histories) are used for inferring the probabilistic model of a principal's behaviour.

- A truster p_i uses the behaviour probabilistic model of a trustee p_j, to estimate the probabilities of potential outcomes of a future interaction with p_j. Based on these probabilities, the trust of p_i in p_j is evaluated. For example, Teacy et al. (2006) defined the trust of p_i in p_j to be an estimated probability that the next interaction with p_j is successful.

In many existing frameworks the so-called *beta model* (Jøsang and Ismail, 2002) is adopted. This is a *static* model in the precise sense that the behaviour of any principal p_j is assumed to be representable by a fixed probability distribution over outcomes, invariantly in time. That is each principal p_j is associated with a fixed real number $0 \leq \Theta_j \leq 1$ indicating the assumption that an interaction involving p_j yields success with probability Θ_j. This simple model gives rise to trust computation algorithms that attempt to 'guess' p_j's behaviour by approximating the unknown parameter Θ_j from the history of interactions with p_j (cf., e.g., Sassone et al., 2006).

There are several examples in the literature where the beta model is used, either implicitly or explicitly, including Jøsang and Ismail's beta reputation system (Jøsang and Ismail, 2002), the systems described by Mui et al. (2002)) and by Buchegger and Le Boudec (2004), the Dirichlet reputation systems (Jøsang and Haller, 2007), TRAVOS (Teacy et al., 2006), and the SECURE trust model (Cahill et al., 2003). Recently, the beta model and its extension to interactions with multiple outcomes (the Dirichlet model) have been used to provide a first formal framework for the analysis and comparison of computational trust algorithms (Sassone et al., 2006; Nielsen et al., 2007; Krukow et al., 2008). In practice, these systems have found space in different applications of trust, e.g., online auctioning, peer-to-peer filesharing, and mobile ad-hoc routing.

All the existing systems apply Bayesian data analysis (see e.g. Sivia, 1996) to the history of interactions h with a given principal p_j to estimate the probability Θ_j that an interaction with p_j yields success. In this framework the family of beta probability density functions (pdfs) is used, as a conjugate prior, together with the data h to derive a posterior beta probability density function for Θ_j. The resulting beta function, denoted by $\mathbf{B}(\alpha, \beta)$, has the parameters α and β, where $\alpha = \#_{\mathsf{s}}(h) + 1$ (resp. $\beta = \#_{\mathsf{f}}(h) + 1$) is the number of successful (resp. unsuccessful) interactions in h augmented by one. Full explanation can be found in Section 3.2.1 and the article by Sassone et al. (2006). An important consequence of this representation is that it allows us to estimate the so-called *predictive probability*, i.e., the probability that the next interaction with p_j will yield success (s), given the history h. Such an estimate is given by the expected value

of Θ_j given the distribution $\mathbf{B}(\alpha, \beta)$

$$P(\mathsf{s} \mid h\, \mathbf{B}) = \mathbf{E}_{\mathbf{B}(\alpha,\beta)}(\Theta_j) = \frac{\alpha}{\alpha + \beta} \ .$$

Thus, in this simple and popular model, the predictive probability depends only on the number of past successful interactions and the number of past failures.

The major limitation of the current beta based systems is that they assume a fixed probabilistic behaviour for each principal; that is for each principal, there exists a fixed probability distribution over possible outcomes of its interactions. This assumption of fixed behaviour may not be realistic in many situations, where a principal possibly changes its behaviour over time. Just consider, e.g., the example of an agent which can autonomously switch between two internal states, a normal 'on-service' mode and a 'do-not-disturb' mode. This limitation of the beta systems has been recognised by many researchers e.g. Mui et al. (2002), Teacy et al. (2006), Xiong and Liu (2004), Jøsang and Ismail (2002), and Buchegger and Le Boudec (2004). While in some of these works (e.g. Mui et al., 2002; Teacy et al., 2006) it is only pointed out that the assumption of fixed behaviour is not realistic, other works (e.g. Xiong and Liu, 2004; Jøsang and Ismail, 2002; Buchegger and Le Boudec, 2004) employed the notion of '*decay*' principle to favour recent events over information about older ones.

The decay principle can be implemented in many different ways, e.g., by using a finite 'buffer' to remember only the most recent n events (Xiong and Liu, 2004), or linear and exponential decay functions, where each outcome in the given history is weighted according to the occurrence time (old outcomes are given lower weights than newer ones) (Jøsang and Ismail, 2002; Buchegger and Le Boudec, 2004). Whilst decay-based techniques have proved useful in some applications, it will be shown through the analysis in Chapter 5 that the decay principle is useful (for the purpose of predicting future actions of the trustee) only when the trustee's behaviour is highly *stable*, that is when it is very unlikely to change its behaviour. Another limitation of this technique, which is also shown in Chapter 5 is that the optimal value of the decay parameter depends on the behaviour of the trustee which is hidden from its partners. Therefore there is still need to develop methods to reliably evaluate the trust in trustees which change their behaviours frequently, and at the same time do not require any information about the hidden behaviour of the trustee.

1.1 Innovation

1.1.1 Dynamic behaviour modelling

In this thesis the assumption of fixed behaviour of a trustee, represented by a single probability distribution Θ over outcomes, is lifted. Instead, the behaviour of the trustee

is assumed to be 'dynamic' in a discrete sense. That is after each interaction, the trustee can make a transition from its current state to another one; the probability distribution Θ over potential outcomes of an interaction with the trustee is assumed then to be dependent on the trustee's underlying state rather than being fixed. To facilitate learning the dynamic behaviour of the trustee, it is also assumed that the number of its possible states is finite.

The probabilistic model for the trustee's behaviour is assumed therefore to be state based, where each state is associated with a probability distribution over possible outcomes. Since the transition of the trustee between its states is hidden from its interaction partners, and only its actions are observable, the *Hidden Markov Models* (HMMs) are adopted for this modelling purpose. Such a representation of behaviour enables describing the 'stability' of the trustee as its expected probability of transition from the current state to a different one (cf. Section 5.5).

1.1.2 Evaluating the quality of trust models

The representation of a trustee's behaviour by a finite-state HMM λ provides a framework for analysing the quality of different trust evaluation algorithms. Namely, given a current state of the trustee, the probability distribution over possible outcomes of interaction with the trustee can be evaluated. Such probability distribution is called in this thesis the *real predictive probability distribution*. On the other hand, applying a particular trust model to available observations about the trustee results in another probability distribution (called the *estimated predictive probability distribution*) which is meant to approximate the real predictive probability distribution. The quality of a trust evaluation algorithm is therefore quantified by the expected difference, e.g. the *quadratic distance* (cf. Section 5.4), and Kulback-Leibler divergence (Cover and Thomas, 2006), between the real and estimated predictive distributions. This framework is adopted in this thesis to evaluate and compare between trust models.

1.1.3 Analysis of the beta trust model with the decay principle

As the decay principle was proposed to cope with principals' dynamic behaviours, it is important to identify the cases where such technique is effective and also the cases where it is not. Using HMMs to represent principals' dynamic behaviours, the quality evaluation framework described above in Section 1.1.2 is used to derive an expression for the expected difference ('estimation error') between the real and estimated predictive distributions. This error is analysed in terms of the decay factor as a user-defined parameter for the beta trust model, and also in terms of the trustee's stability which is the tendency of the trustee to preserve a fixed behaviour (i.e. preserve the same probability distribution over outcomes). By deploying such error expression in an experimental

setting, it is shown that the decay principle is effective (i.e. reduces the estimation error) only when the trustee's behaviour is relatively stable, and the optimal decay factor depends basically on the trustee's behaviour which is hidden from the observer.

1.1.4 HMM-based trust and reputation models

Given the limitation of the decay technique in coping with principals' dynamic behaviour, a novel model for trust, called the *HMM-based trust* is proposed in Chapter 6, where the hidden behaviour λ of the trustee is approximated by a finite-state HMM η. The construction of the approximate model η is based on information about past interactions with the trustee using the Baum-Welch training algorithm (cf. Section 2.5.4). The HMM obtained is then used to compute the probability of each potential outcome in the next interaction with the trustee, and therefore make predictions about the outcomes of future interactions. It is shown in an experimental setting that the HMM-based model outperforms the beta model with decay when the trustee's behaviour is unstable (dynamic), i.e. more likely to change its state rather than staying in the same current state.

The HMM-based trust model requires having a sufficiently long history of interaction with the trustee to obtain a reliable approximate behaviour HMM η for the trustee. In many practical situations, the sequence of personal observations available to the truster is not sufficiently long to learn the behaviour of a trustee. In these cases, learning the behaviour of the considered trustee using such a short sequence would not be reliable for obtaining an approximate model η for the trustee. In Chapter 7, a HMM-based reputation model is described where the truster compensates its shortage of information by collecting reputation reports (feedbacks) about the trustee from other principals (*reputation sources*) who also interacted with the same trustee. The format of reputation reports are described along with a reputation 'mixing' algorithm for using these reports in the trust evaluation process.

1.1.5 Using trust information

As described earlier in this chapter, the main objective of evaluating trust in principals (trustees), is to make security sensitive decisions regarding interactions between principals. Therefore, an important question is how trust information can be used in the security policies which control the interactions between principals. The answer to this question is challenging in the case of probabilistic trust, as there is always a risk of interacting with a chosen peer. So security policies which are based on probabilistic trust should be defined such that the likelihood of the risk is minimised. As an example of employing the probabilistic trust in security protocols, Chapter 4 describes an application of the probabilistic trust in the Crowds anonymity protocol (Reiter and Rubin,

1998) aiming at preserving the anonymity of principals, i.e. hiding their identities from observers (eavesdroppers) when they issue web transactions. It is shown that while each member in the protocol has a particular probability of being 'corrupted', a required level of anonymity (privacy) for all members may be achieved by the correct use of trust information.

1.2 Thesis structure

The chapters in this thesis are organised as follows. In Chapter 2, we give a brief mathematical background. This background includes Bayesian analysis as it has been used in modern trust systems for estimating parameters of probabilistic models. We also give a brief review for basic properties of discrete Markov chains (DMCs) followed by a description of HMMs and their main problems.

Chapter 3 describes the traditional and also recent approaches to formalising the notion of trust. That chapter starts by describing the classical approach to trust, known by *credential based trust*, along with concepts related to this approach, e.g. *credentials*, *policies*, *trust negotiation*, and *policy languages*. The chapter follows by describing the modern approach of trust, the probabilistic trust, and models based on this approach, e.g. the *beta reputation model*, and the *Dirichlet reputation models*.

Following the literature review, we focus in Chapter 4 on an application of probabilistic trust as a basis for preserving anonymity, one of the increasingly important security properties in modern computing networks. In this chapter trust information about principals is assumed to be available. It is then described how this trust information is employed to adjust the parameters of a well known anonymity protocols, called the Crowds protocol (Reiter and Rubin, 1998). The chapter provides an overview of the Crowds protocol, and then follows by presenting a proposal for an extension which allows for using trust information.

After Chapter 4, the rest of the dissertation is concerned with the problem of computing trust, and formulating reputation assuming that principals exhibit dynamic behaviour. In this context, the dynamic behaviour of a principal is modelled by a finite-state HMM. Since the decay principle was advised to avoid the assumption of fixed behaviour in the existing beta trust model, Chapter 5 is devoted to analyse in details the existing beta-based trust model with the principle of decay in order to identify its advantages and also describe its limitations.

In Chapter 6, we try to cope with the limitations of the beta-based trust model by introducing the *HMM-based trust model* as an approach to evaluate trust in principals having dynamic behaviours. We describe the foundations of HMM-trust model, and compare it to the existing beta-based trust model. Given the basic foundations of the

HMM-based trust model, we complete the framework of HMM-based trust by encompassing the notion of *reputation* in Chapter 7. The elements of a reputation report are described along with an algorithm for combining available reputation reports. Finally, in Chapter 8 we conclude our work and describe possible extensions.

Chapter 2

Mathematical background

This chapter describes the mathematical foundations required for analysing existing probabilistic trust models, and also for solving the specific research problem of extending these models to capture different patterns of principals' behaviours. This chapter starts by describing the Bayesian inference which is a common framework to analyse observations against assumed hypotheses. In Section 2.2, the problem of parameter estimation is then described along with different methods for solving it. Also the model selection problem and common approaches for handling it are briefly described in Section 2.3.

Because Discrete Markov Chains (DMCs) are used in this work to model transitions of principals between their internal states, DMCs are described along with the related properties, *irreducibility*, *recurrence*, and *aperiodicity* in Section 2.4. A hidden Markov model (HMM) is obtained by associating individual states of a DMC to probability distributions over observables. As HMMs are chosen in this work to model principals' dynamic behaviours, Section 2.5 describes HMMs and their basic problems. We also review, in Section 2.6, different approaches to construct minimal HMMs from data samples. The last section 2.7 in this chapter is devoted to describing the useful ergodicity and mixing properties of DMCs and HMM in the general context of random processes.

2.1 Bayesian inference

Bayesian inference is a statistical inference process in which observations are used to update the probability that a hypothesis may be true (Sivia, 1996). Bayesian inference involves collecting evidence which can be consistent or inconsistent with the assumed hypothesis. As evidence accumulate, the degree of belief in such a hypothesis changes. In this context, the belief is modelled as the probability that the hypothesis is true. The name *'Bayesian'* comes from the use of the *Bayes' theorem* in the inference process. The

Bayes' theorem is stated generally as

$$P(X \mid Y) = \frac{P(Y \mid X) \times P(X)}{P(Y)}, \tag{2.1}$$

where X and Y are two events such that $P(X) > 0$ and $P(Y) > 0$. In terms of a hypothesis and observed data, Bayes' theorem can be expressed as follows.

$$P(\textit{hypothesis} \mid \textit{data}) \propto P(\textit{data} \mid \textit{hypothesis}) \times P(\textit{hypothesis}).$$

The power of Bayes' theorem lies in the fact that it relates the quantity of interest, the belief (probability) that the hypothesis is true, to the term which we have a better chance of being able to express, the probability of the observed data if the hypothesis was true. The term $P(\textit{hypothesis})$ is called the *prior* probability; it represents the belief in the truth of the hypothesis before obtaining the observed data. Using Bayes' theorem, the prior probability is modified by the data likelihood under the hypothesis in question, or $P(\textit{data} \mid \textit{hypothesis})$, and yields the *posterior* probability, $P(\textit{hypothesis} \mid \textit{data})$ which represents the belief in the hypothesis truth in the light of the observed data. Note this process encapsulates the process of learning.

The Bayesian approach have proved successful in several applications to quantify the probability that a hypothesis is true. Many examples, including the following one are described by Jaynes, 2003.

Example 2.1. *Consider the hypothesis that the 'perceived' size of an object is linearly correlated with the distance to the object. Observations (evidences) which confirm this hypothesis are naturally accumulated by the human brain. With the help of Bayesian inference, an adult can assign therefore a high probability to the truth of this hypothesis. However, this truth can be doubted (assigned less probability) by a child walking in a room having a magnifying mirrors.*

The Bayesian approach has also applied to the problem of trust evaluation in networked principals. In this particular problem, evidence (observations) about a principal (trustee) is being collected by its interaction partners (trusters). Using the accumulated evidence, a truster tries to 'learn' the behaviour of the trustee, and therefore assign a probability distribution over possible hypotheses regarding future interactions with the trustee.

2.2 Parameter estimation

Given a probabilistic model which is defined by a set of free parameters, the parameter estimation problem is the task of determining the model's parameter values in the light of observed data, or equivalently finding the model's parameter values which explain the observations in hand.

2.2.1 Bayesian estimation

In this parameter estimation method, Bayesian inference is used to estimate the model parameters using the given observations. For example, suppose it is required to find how much a coin is Head-biased given the outcomes of a sequence of its flips. In terms of the parameter estimation problem, the probabilistic model here is a fixed probability distribution over the outcomes (head, tail) of any flip of the given coin, and it is required to estimate the value of its parameter $r = P(Head)$, that is the probability that a flip experiment of this coin yields Head.

This problem can be formulated by considering an infinite number of hypotheses. Each hypothesis corresponds to a specific value for r in the range from 0 to 1. To proceed with the inference process, it is required to assign a prior probability to each hypothesis, that is an initial probability density function (pdf) for r as a continuous random variable. Each time we get an observation (head or tail), we use the Bayes' theorem to update the prior pdf. Figure 2.1(a) shows the posterior pdf for r after each observation in the sequence $H\ H\ T\ T\ H$, starting from a uniform prior pdf. Figure 2.1(b) shows the posterior pdf after observing 25 heads and 15 tails. It can be seen that the expected value of r gradually approaches a specific value (0.6) which indicates our estimate for r. Note also that the width of the posterior pdf becomes narrower with more data, indicating more confidence in the estimate of r.

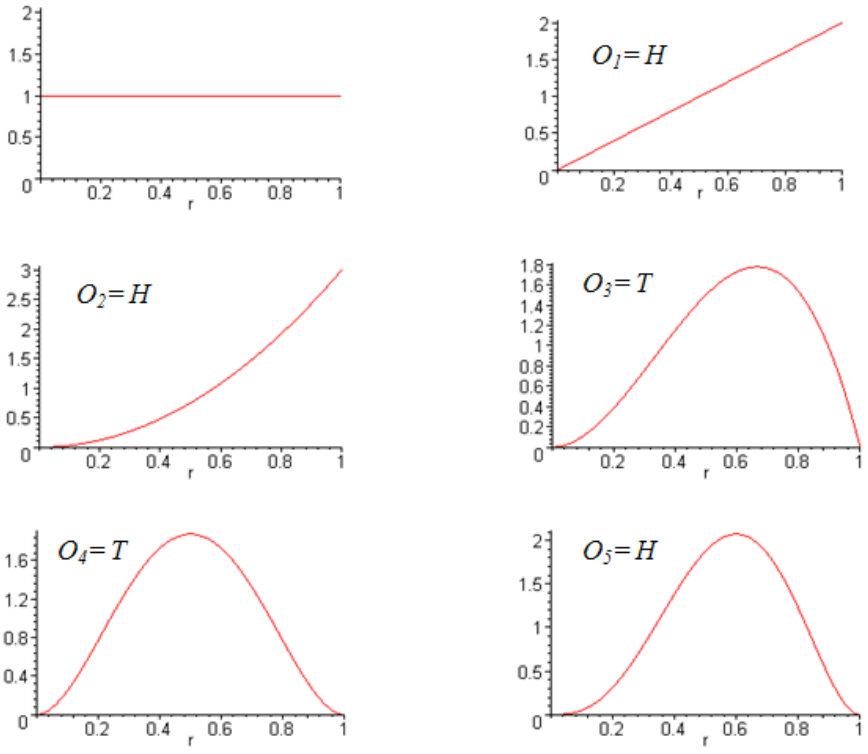
2.2.2 Maximum Likelihood Estimate (MLE)

In this approach parameter values are chosen such that the likelihood of observations under the given model is the maximum (e.g. Myung, 2003). In other words, assuming that the model $\lambda(\theta)$ is defined by the compound (multi-dimensional) parameter θ, the optimum value $\hat{\theta}$ for the parameter θ is determined by the following equation,

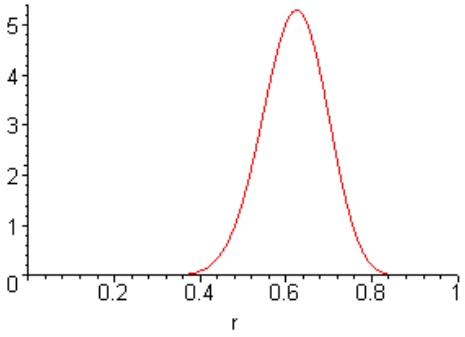
$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, P(O \mid \lambda(\theta)). \tag{2.2}$$

2.2.3 Expectation-Maximisation algorithm

In many cases, the observations O can be seen as incomplete data relying on non-observed (hidden) data X. In such cases the hidden data X corresponding to the observation O is not observed directly, but only indirectly through O. The requirement in such cases is to choose the compound parameter θ of the underlying model such that the probability of the observations, $P(O \mid \theta)$, is maximised. Because observations O are dependent on the hidden data X, maximising $P(O \mid \theta)$ is not straightforward. One efficient iterative algorithm for achieving this task is called *Expectation-Maximization (EM)* algorithm (Dempster et al., 1977). To explain the algorithm,



(a) The posterior pdf for a coin bias-weighting after each symbol in the sequence H,H,T,T,H



(b) The posterior pdf for a coin bias-weighting after a sequence of 25 heads and 15 tails

FIGURE 2.1: The evolution of the posterior pdf for the bias-weighting of a coin

- The following quantity is called the *complete data* log-likelihood under the parameter θ.

$$L(O, X \mid \theta) = \log P(O, X \mid \theta).$$

Since O is known, $L(O, X \mid \theta)$ can be seen as a function of the parameter θ and the hidden random variable X.

- Letting the probability distribution of X be determined by an initial parameter value θ_0, and O, the function $Q(\theta \mid \theta_0)$ is defined to be the expected value of

$L(O, X \mid \theta)$, that is,

$$Q(\theta \mid \theta_0) = \mathbf{E}_X [L(O, X \mid \theta) \mid O, \theta_0] = \sum_X P(X \mid O, \theta_0) L(O, X \mid \theta). \quad (2.3)$$

The function $Q(\theta \mid \theta_0)$ is then called the expectation of the complete data likelihood given the observations O. It is shown by Dempster et al. (1977) that increasing $Q(\theta \mid \theta_0)$ results in increasing observation probability $P(O \mid \theta)$. The algorithm thus goes as follows.

1. Choose an initial parameter value θ_0.
2. (Expectation step): Evaluate $Q(\theta \mid \theta_0)$.
3. (Maximisation Step): Get the value θ_1 which maximises $Q(\theta \mid \theta_0)$, that is
$$\theta_1 = \underset{\theta}{\operatorname{argmax}} Q(\theta \mid \theta_0).$$
4. Let $\theta_0 = \theta_1$, and go to step (2).

The algorithm terminates when no value θ_1 is found such that $Q(\theta_1 \mid \theta_0) > Q(\theta_0 \mid \theta_0)$, i.e. $\theta_1 = \theta_0$. The parameter value in this case corresponds to a local maximum of the data probability function $P(O \mid \theta)$. An important application of the EM algorithm is estimating the parameters of HMMs using given observations. Applying the EM algorithm to this specific problem results in so-called Baum-Welch algorithm which is described in details by Section 2.5.4.

2.3 Model selection

Model selection is the task of selecting a statistical model from a set of potential models, given observations. This task involves using mathematical analysis to choose the best model from the set of potential models. However, what is meant by best is controversial. A good model selection technique balances between goodness of fit to observations on one hand, and complexity on the other hand. The goodness of fit is measured by the discrepancy between observed values and the values expected under the given model. On the other hand, complexity of the model is measured by the number of free parameters required to define the model. More complex models will be better able to adapt their parameters in order to fit the given information, for example, a sixth-order polynomial can exactly fit any six points. However, if the six points are randomly distributed about a line, the model of a straight line would be enough to approximate the six points, and therefore the sixth-order polynomial model would be of unnecessary complexity. A set

of approaches have been proposed to achieve this compromise between the goodness of fit and complexity of a model. In the following subsections, we give a short description for two of these approaches, namely AIC and BIC.

2.3.1 Akaike Information Criterion (AIC)

In AIC (Akaike, 1974), model selection is based on balancing between the complexity of the selected model and its fitness to the given observations. The complexity of a model is defined by the number of independently adjusted parameters within the model. Given a model λ, the *Akaike Information Criterion* AIC, of λ is defined by the equation

$$AIC(\lambda) = (-2)\log(ML(\lambda)) + 2K(\lambda),$$

where $ML(\lambda)$ is the maximum likelihood of the given observations under the model λ, and $K(\lambda)$ is the number of independently adjusted parameters within λ. The maximum likelihood term $ML(\lambda)$ indicates the fitness of the model to the given observations and is defined by the equation,

$$ML(\lambda) = \max_{\theta_1,\theta_2,\ldots,\theta_k} P(O \mid \lambda(\theta_1,\theta_2,\ldots,\theta_k)), \tag{2.4}$$

where $k = K(\lambda)$. When there are several models, the model which gives the minimum value for AIC is selected. Therefore AIC, proposes selecting the model which maximises fitness to the observations, and is as simple as possible.

2.3.2 Bayesian Information Criterion (BIC)

In this model selection method, also called *Schwarz Information Criterion (SIC)*, the criterion BIC of a given model λ is defined by the following equation (Schwarz, 1978).

$$BIC(\lambda) = ML(\lambda) - \frac{1}{2}K(\lambda)\log n, \tag{2.5}$$

where $ML(\lambda)$ is the maximum likelihood of the given observations under the model λ, $K(\lambda)$ is the number of independently adjusted parameters within λ, and n is the number of observations. When there are several models, the model which gives the maximum value for BIC is selected as the best model for observations. Therefore, like AIC, the BIC method balances between the fitness of the model to the observations, expressed by $ML(\lambda)$, on one hand, and its complexity, expressed by $\frac{1}{2}K(\lambda)\log n$ on the other hand. The difference is that BIC leans more than AIC towards lower-dimensional models with larger number of observations n, since $K(\lambda)$ is multiplied by $\log n$.

2.4 Markov chains

As we described in the introduction, we will use hidden Markov models (HMMs) to model the behaviour of any principal in a global computing environment. Since a Markov chain is essentially the underlying state transition system of any HMM, many properties of HMMs are inherited from those of Markov chains. So, in this section we review some fundamental properties of the homogeneous finite-state discrete Markov chains (DMCs). These properties are used in our work to analyse HMMs.

Any DMC is characterised by an $n \times n$ *state transition matrix* $\boldsymbol{A}$, where n is the number of underlying states, and A_{ij} is the probability of transiting to state j, given state i. We consider only *homogeneous* DMC, in which the state transition matrix is fixed over time. For a fuller treatment of the notions of Markov chains, the reader is referred to, e.g., Grimmet and Stirzaker (2001); Norris (1997); Brémaud (1998).

2.4.1 Irreducibility

A DMC is irreducible if each state is reachable from any other state with a positive probability. That is, at any time, from each state i, there is a positive probability to eventually reach each state j. Denoting by A^m_{ij} the (i,j)-entry of the mth power of matrix $\boldsymbol{A}$, it can be proved that A^m_{ij} is the probability to reach state j in exactly m transitions, given the chain is at state i. The condition of irreducibility can then be expressed formally as follows.

Definition 2.1 (irreducibility)**.** For $\boldsymbol{A}$ a DMC and i a state, we say that state i reaches state j, written $i \mapsto j$, whenever $A^m_{ij} > 0$, for some finite m, and that $\boldsymbol{A}$ is irreducible if $i \mapsto j$, for all i and j.

2.4.2 Recurrence

A state i of a DMC can be classified as either *recurrent* or *transient*. The state i is recurrent if starting from i, the chain is guaranteed to eventually return to i, otherwise it is transient. Recurrent states can be *positive* or *null* recurrent. A state i is positive recurrent if starting from i, the expected return time to i is finite ($< \infty$). In the following, we shall write $q_k = i$ to indicate that i is the kth state visited by a DMC in a given run $q_0 q_1 q_2 \cdots$.

Definition 2.2 (classification of states)**.** For $\boldsymbol{A}$ a DMC and i a state, we say that i is:

$$\begin{array}{ll} \textit{recurrent} & \text{if} \quad P(\, q_k = i, \text{ for some } q_0 \cdots q_k \mid q_0 = i \,) = 1; \\ \textit{transient} & \text{otherwise.} \end{array}$$

It can be proved that a state j is recurrent if and only if $\sum_{m=0}^{\infty} A_{jj}^m = \infty$. This characterisation has the following corollary. If state j is recurrent, then for any other state i such that $i \mapsto j$ and $j \mapsto i$, we have also $\sum_{m=0}^{\infty} A_{ii}^m = \infty$, which means that i is also recurrent. This implies that i and j are either both recurrent or both transient. It also follows that in an irreducible chain, where each state is reachable from any other state, either all states are transient, or they all are recurrent.

Let T_i be a random variable representing the *time of the first return* to state i, namely $\min\{\, k \geq 1 \mid q_k = i \,\}$. Using the homogeneity property of DMC, we can define the *mean return time* of state i as

$$\mu_i = \mathbf{E}[\, T_i \mid q_0 = i \,].$$

Definition 2.3 (classification of recurrent states)**.** For $\boldsymbol{A}$ a DMC and i a recurrent state, we say that i is:

$$\begin{aligned} \textit{null} \quad &\text{if} \quad \mu_i = \infty; \\ \textit{positive} \quad &\text{if} \quad \mu_i < \infty. \end{aligned}$$

If all states are positive recurrent, the DMC is said to be positive recurrent. In particular if $\boldsymbol{A}$ is *finite* and *irreducible*, then it is positive recurrent. Positive recurrence of an irreducible DMC guarantees the existence and uniqueness of the *stationary probability distribution* described below.

Definition 2.4 (stationary distribution)**.** A vector $\boldsymbol{\pi} = (\pi_j \mid j \in Q)$ is a stationary distribution on Q if

$$\begin{aligned} &\pi_j \geq 0 \ \text{ for all } j, \quad \text{and } \textstyle\sum_{j \in Q} \pi_j = 1; \\ &\boldsymbol{\pi}\, \boldsymbol{A} = \boldsymbol{\pi}. \end{aligned}$$

So if the stationary probability distribution is the initial distribution over a DMC states, it will remains invariant in time, meaning that at any time the probability distribution over the DMC states is the stationary distribution. In an irreducible chain, the mean return time determines such invariant distribution.

Theorem 2.5 (existence of stationary distribution)**.** *An irreducible Markov chain has a stationary distribution $\boldsymbol{\pi}$ if and only if all its states are positive recurrent. In this case, $\boldsymbol{\pi}$ is the unique stationary distribution and is given by $\pi_i = \mu_i^{-1}$.*

Now we recall the fact that the states of a finite DMC are all positive recurrent if it is irreducible. Since in our research we consider only the finite states DMC and hidden Markov models (described later), we find that the condition of irreducibility guarantees the existence of the stationary probability distribution according to this fact and the

above theorem. The existence of a stationary distribution is not sufficient to describe the asymptotic behaviour of a DMC. However, the following condition of aperiodicity guarantees convergence to the stationary distribution regardless of the DMC's initial distribution.

2.4.3 Aperiodicity

Definition 2.6 (aperiodicity)**.** For $\boldsymbol{A}$ a DMC, the period of i is $d(i) = \gcd\{\, m \mid A_{ii}^m > 0 \,\}$. State i is aperiodic if $d(i) = 1$; and $\boldsymbol{A}$ is *aperiodic* if all its states are such.

Theorem 2.7 (convergence to stationary distribution)**.** *For $\boldsymbol{A}$ an irreducible, and aperiodic Markov chain, $\lim_{m\to\infty} A_{ij}^m = \mu_j^{-1}$, for all i and j.*

The above convergence to the stationary distribution is shown by Theorem (8.9) in (Billingsley, 1995) to be at an exponential rate as follows.

Theorem 2.8 (exponential convergence)**.** *For an irreducible and aperiodic finite-state DMC, it holds that*

$$|A_{ij}^m - \pi_j| \leq C\rho^m \,,$$

where $C \geq 0$ and $0 \leq \rho < 1$.

A DMC which is positive recurrent, and aperiodic is called ***ergodic***. Given the above properties, it is obvious that Theorems (2.5) and (2.7) play the basic role to analyse the asymptotic behaviour of a given ergodic DMC.

2.5 Hidden Markov Models (HMM)

A *Hidden Markov Model (HMM)* (Baum and Petrie, 1966) is a probabilistic finite state machine which has been widely used for probabilistic sequence modelling. The approach of HMM has been used in many applications including speech recognition (Bahl et al., 1993; Rabiner, 1989), DNA and protein modelling (Hughey and Krogh, 1996), information extraction (Seymore et al., 1999), handwritten character recognition (J. Hu; Brown, 1996), and gesture recognition (Eickeler et al., 1998).

A discrete-time first-order HMM is a probabilistic model that describes a stochastic sequence of symbols $O = o_1, o_2, \ldots, o_T$ as being an observation of an underlying (hidden) random sequence of states $Q = q_1, q_2, \ldots, q_T$, where this hidden process is Markovian, i.e. each state q_i depends only on the previous one q_{i-1}. Thus an HMM can be seen as a Markov chain where each state is associated with a particular probability distribution over the set of possible symbols (observations). However, a key difference between HMMs and Markov chains is that in an HMM, state transitions are not observed as is the case

in Markov chains, and only observations are visible. In the following we precisely define discrete-time first-order HMMs, and describe their basic problems. For a more detailed description, the reader is referred to, e.g., Rabiner (1989).

2.5.1 Definition

A discrete HMM is formally defined by the following elements:

- A set $S = \{1, 2, \ldots, N\}$ of (hidden) states.
- A state transition matrix $\boldsymbol{A} = \{A_{ij}\}$ of size $N \times N$, where an element $0 \leq A_{ij} \leq 1$ is the probability of transition from state i to state j:

$$A_{ij} = P\left(q_{t+1} = j \mid q_t = i\right), \qquad 1 \leq i, j \leq N$$

  where q_t denotes the state occupied by the system at time t. As A_{ij} is a probability distribution over the set of states, we always have $\Sigma_{j=1}^{N} A_{ij} = 1$ for any state i.
- A set $V = \{z_1, z_2, \ldots, z_K\}$ of observation symbols. This set is called the alphabet of the model. A sequence of observation symbols is the physical output of the model, rather than the states themselves.
- An emission matrix $\boldsymbol{B} = \{B_i\left(z_k\right)\}$ of size $N \times K$, where an element $0 \leq B_i\left(z_k\right) \leq 1$ is the probability of observing symbol z_k given the current state is i, that is,

$$B_i\left(z_k\right) = P\left(o_t = z_k \mid q_t = i\right), \qquad 1 \leq i \leq N,\, 1 \leq k \leq K$$

  As $B_i\left(z_k\right)$ is a probability distribution over the set of symbols, we always have $\sum_{k=1}^{K} B_i(z_k) = 1$ for any state i.
- Initial state probability distribution $\boldsymbol{\pi} = \{\pi_i\}$, where an element $0 \leq \pi_i \leq 1$ is the probability of being in state i at the time 1, that is,

$$\pi_i = P\left(q_1 = i\right), \qquad 1 \leq i \leq N$$

  Also as $\boldsymbol{\pi}$ is a probability distribution over the set of states, we have $\sum_{i=1}^{N} \pi_i = 1$.

Thus an HMM is completely defined by a five-tuple $\lambda = (S, V, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$. The probability distributions $\boldsymbol{A}$,$\boldsymbol{B}$, and $\boldsymbol{\pi}$ are called the parameters of the given HMM.

2.5.2 HMM basic problems

There are three main problems involved with using HMMs:

1. Given an HMM $\lambda = (S, V, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, and an observation sequence $O = o_1 o_2, \ldots, o_T$, we want to efficiently compute $P(O \mid \lambda)$, that is the probability of the given observation sequence O, given the model λ. This is usually solved by the *forward-backward* algorithm described later in Section 2.5.3.

2. Given an HMM $\lambda = (S, V, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, and an observed sequence $O = o_1 o_2, \ldots, o_T$, we want to determine the state sequence that most probably generated O, that is, $\hat{Q} = \hat{q_1}, \hat{q_2}, \ldots, \hat{q_T}$, where $\hat{q_i} \in S$ such that,

$$\hat{Q} = \underset{Q}{\operatorname{argmax}} P(O, Q \mid \lambda). \tag{2.6}$$

   This problem is solved by the *Viterbi algorithm* (Forney, 1973).

3. Given an observation sequence $O = o_1 o_2, \ldots, o_T$, and an HMM model λ with a specified number of states. We want to determine the values of the parameters $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}$ of λ such that the probability of the sequence O is maximised under λ. That is,

$$\left(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \bar{\boldsymbol{\pi}}\right) = \underset{(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})}{\operatorname{argmax}} P(O \mid \lambda). \tag{2.7}$$

   The solution for this problem is obtained by adopting the maximum likelihood estimate (MLE) approach described in Section 2.2.2. Applying MLE approach to this problem yields the *Baum-Welch algorithm* which is described later in Section 2.5.4.

2.5.3 Forward-Backward algorithm

By the formal definition of a HMM stated in Section 2.5.1, the joint probability of a sequence of observation $O = o_1\, o_2, \cdots, o_T$, and an underlying sequence of states $Q = q_1, q_2, \cdots, q_T$ is given by the following equation

$$P(O, Q \mid \lambda) = \pi_{q_1} \cdot B_{q_1}(o_1) \cdot A_{q_1 q_2} \cdot B_{q_2}(o_2) \cdots A_{q_{T-1} q_T} \cdot B_{q_T}(o_T).$$

The probability of a sequence of outcomes $O = o_1\, o_2 \cdots o_T$ given a HMM λ is given therefore by summing the joint probability $P(O, Q \mid \lambda)$ over all possible underlying sequences Q of states. That is the probability $P(O \mid \lambda)$ is given by the following equation.

$$P(O \mid \lambda) = \sum_{q_1, \ldots, q_T \in S} \pi_{q_1} \cdot B_{q_1}(o_1) \cdot A_{q_1 q_2} \cdot B_{q_2}(o_2) \cdots A_{q_{T-1} q_T} \cdot B_{q_T}(o_T). \tag{2.8}$$

Since the number of permutations of the underlying state sequence Q is N^T, the computation of $P(O \mid \lambda)$ using the above equation requires an order of $2T \cdot N^T$ calculations ($(2T-1)N^T$ multiplications and $N^T - 1$ additions). That is the computation time is

exponential in the sequence length T, and therefore is computationally impractical when the given sequence O is very long.

The *forward-backward* algorithm has been introduced in the literature as a means for practical evaluation of this probability with much lower computation cost. In the following we provide a brief description of this algorithm which evaluates the probability of any observation sequence given a particular HMM model λ efficiently based on dynamic programming techniques. The reader is referred to Rabiner (1989) for more details on this algorithm. Given an HMM $\lambda = (S, V, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, and an observation sequence $O = o_1 o_2, \ldots, o_T$, the probability $P(O \mid \lambda)$ is obtained by inductively evaluating the *forward* variable $\alpha_t(i)$ defined by the following equation

$$\alpha_t(i) = P(o_1 o_2 \ldots o_t, q_t = i \mid \lambda). \tag{2.9}$$

That is $\alpha_t(i)$ is the joint probability of the partial observation sequence, $o_1 o_2 \ldots o_t$, (until time t), and the state i at time t, given the model λ. A procedure, which we call the 'forward' procedure, evaluates $P(O \mid \lambda)$ as follows.

1. Initialization:
$$\alpha_1(i) = \pi_i B_i(o_1), \qquad 1 \leq i \leq N$$

2. Induction:
$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) A_{ij} \right) B_j(o_{t+1}), \qquad 1 \leq t \leq T-1,\ 1 \leq j \leq N$$

3. Termination:
$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

Considering the computation cost of the above procedure, observe that at each iteration in the induction step (for a given t), the values of $\alpha_{t+1}(j)$ are computed for all states $j = 1, 2, \ldots, N$; for each state j, the evaluation of $\alpha_{t+1}(j)$ requires summing up N product terms and one additional product calculation (for multiplying the sum by $B_j(o_{t+1})$). Thus each induction step requires $N(N+1)$ multiplications and $N(N-1)$ additions. Noting that the algorithm involves $T-1$ induction iterations, and one initialization step requiring N multiplications, the whole algorithm requires $N(N+1)(T-1)+N$ multiplications and $N(N-1)(T-1)$ additions. That is the algorithm requires on the order of $N^2 T$ calculations in total.

In a similar manner to the above procedure, the probability $P(O \mid \lambda)$ can be also evaluated by the inductive evaluation of the so called *backward* variable, denoted by $\beta_t(i)$ and defined as follows

$$\beta_t(i) = P(o_{t+1} o_{t+2} \ldots o_T \mid q_t = i, \lambda). \tag{2.10}$$

I.e. $\beta_t(i)$ is the conditional probability of the partial observation sequence $o_{t+1}o_{t+2}\ldots o_T$ (from time $t+1$ to the end) given that the state at time t is i, and the model λ. Using the following procedure, which we call the 'backward' procedure, the probability of the sequence O given the HMM λ can be evaluated.

1. Initialization:
$$\beta_T(i) = 1, \qquad 1 \leq i \leq N$$

2. Induction:
$$\beta_t(i) = \sum_{j=1}^{N} A_{ij}\, B_j(o_{t+1})\, \beta_{t+1}(j), \qquad 1 \leq t \leq T-1,\ 1 \leq j \leq N$$

3. Termination:
$$P(O \mid \lambda) = \sum_{i=1}^{N} \pi_i B_i(o_1)\beta_1(i).$$

Similar to the reasoning used for the forward procedure, the above backward procedure requires also about $N^2\,T$ calculations. Comparing the forward-backward algorithm to the direct computation of the probability using Equation (2.8) which requires about $2T \cdot N^T$ calculations, it is obvious the the former is more efficient for evaluating the probability of a given sequence. That is the computation cost in the forward-backward algorithm is linear in the sequence length T rather than exponential in the case of computation using Equation (2.8).

2.5.4 Baum-Welch algorithm

The problem of estimating the parameters of an HMM λ to maximise the likelihood of given observations exactly coincides with the scenario of observed and hidden data described in Section 2.2.3. In this context the observed data is the observed symbol sequence, and the hidden data is the state sequence generating the observed symbol sequence. Thus the EM algorithm is applicable to the problem of estimating HMM parameters, yielding the so called *Baum-Welch algorithm* (Baum et al., 1970; Rabiner, 1989). It is shown by Rabiner and Juang (1993) and Bilmes (1997) how the Baum-Welch algorithm can be derived from the EM algorithm. In the following we sketch the main lines of this derivation.

Thinking of this problem in terms of the EM framework, we assume that we have an initial (a priori) HMM λ', and aim to derive an a posteriori HMM λ which maximises the expected complete data likelihood which is defined generally by (2.3). Considering an observed sequence of symbols $O = o_1\, o_2\, \ldots\, o_T$ and the underlying hidden sequence of

states $q = q_1\, q_2\, \ldots\, q_T$, the expected complete data likelihood can be written as follows

$$Q(\lambda', \lambda) = \sum_{q} P(q \mid O, \lambda') \log P(O, q \mid \lambda). \tag{2.11}$$

The above function $Q(\lambda', \lambda)$ is also called the *Baum's auxiliary function*. Our objective now is to determine the optimal parameter values of the the a posteriori HMM λ which maximises $Q(\lambda', \lambda)$. For doing so, we start by formulating the term $\log P\left(O, q \mid \lambda\right)$ in terms of the parameters of λ as follows

$$\log P\left(O, q \mid \lambda\right) = \log \pi_{q_1} + \sum_{t=2}^{T} \log A_{q_{t-1} q_t} + \sum_{t=1}^{T} \log B_{q_t}\left(o_t\right), \tag{2.12}$$

where π_i denotes the probability that the initial state (q_1) is i. A_{ij} is the probability of transition from state i to state j. $B_i\left(z_k\right)$ is the probability of observing the outcome z_k at state i. Refer to the description of the HMM elements in Section 2.5.1 for more details about these notations.

Substituting Expression (2.12) in (2.11), the function $Q(\lambda', \lambda)$ can be written as follows

$$\begin{aligned} Q(\lambda', \lambda) = & \sum_{i=1}^{N} P\left(q_1 = i \mid O, \lambda'\right) \log \pi_i + \\ & \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T} P\left(q_{t-1} = i, q_t = j \mid O, \lambda'\right) \log A_{ij} + \\ & \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{t=1}^{T} P\left(q_t = i \mid O, \lambda'\right) \delta\left(o_t, z_k\right) \log B_i\left(z_k\right), \end{aligned} \tag{2.13}$$

where N is the number of states, K is the number of possible observation symbols, and the δ-function $\delta\left(o_t, z_k\right)$ is defined as by:

$$\delta\left(o_t, z_k\right) = \begin{cases} 1 & \text{if } o_t = z_k\,. \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.14}$$

For convenience, Eq. (2.13) which formulates the auxiliary function can be rewritten as follows.

$$Q(\lambda', \lambda) = Q_\pi\left(\boldsymbol{\pi}\right) + \sum_{i=1}^{N} Q_{A_i}\left(\boldsymbol{A_i}\right) + \sum_{i=1}^{N} Q_{B_i}\left(\boldsymbol{B_i}\right), \tag{2.15}$$

where $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_N]$ is the vector representing the initial state probability distribution, $\boldsymbol{A_i} = [A_{i1}, A_{i2}, \ldots, A_{iN}]$ is the vector representing the probability distribution over state transitions from state i to other states, and $\boldsymbol{B_i} = [B_i(z_1), B_i(z_2), \ldots, B_i(z_K)]$ is the vector representing the emission probability distribution over outcomes given state i. The functions $Q_\pi(\boldsymbol{\pi})$, $Q_{A_i}(\boldsymbol{A_i})$, and $Q_{B_i}(\boldsymbol{B_i})$ in the above equation are defined as

follows

$$Q_{\pi}(\boldsymbol{\pi}) = \sum_{i=1}^{N} P\left(q_1 = i \mid O, \lambda'\right) \log \pi_i \,, \tag{2.16}$$

$$Q_{A_i}(\boldsymbol{A_i}) = \sum_{j=1}^{N} \left(\sum_{t=2}^{T} P\left(q_{t-1} = i, q_t = j \mid O, \lambda'\right) \right) \log A_{ij} \,, \tag{2.17}$$

$$Q_{B_i}\left(\boldsymbol{B_i}\right) = \sum_{k=1}^{K} \left(\sum_{t=1}^{T} P\left(q_t = i \mid O, \lambda'\right) \delta\left(o_t, z_k\right) \right) \log B_i\left(z_k\right) \,. \tag{2.18}$$

Observe that each term in Equation (2.15) is a function of a probability distribution which parametrises the a posteriori HMM λ. These distributions ($\boldsymbol{\pi}$, $\boldsymbol{A_i}$, $\boldsymbol{B_i}$ $\forall i : 1 \leq i \leq N$) are independent of each other, that is the choice of one of them does not affect the choice of the others. Therefore the auxiliary function is maximised by maximising each term in (2.15) separately. Observe furthermore that each of equations (2.16),(2.17), and (2.18) is in the following form

$$F\left(y_1, y_2, \ldots, y_V\right) = \sum_{v=1}^{V} w_v \log y_v \qquad \text{where} \qquad \sum_{v=1}^{V} y_v = 1 \,. \tag{2.19}$$

Using the *Lagrange multiplier* technique for optimising a function subject to a constraint, the constrained function F defined above can be easily proved to have a global maximum at the point $(\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_V)$, where $\bar{y}_v$ is given by

$$\bar{y}_v = \frac{w_v}{\sum_{v=1}^{V} w_v} \,.$$

Using the above fact, the parameters of the optimal a posteriori model λ are given as follows

$$\bar{\pi}_i = P\left(q_1 = i \mid O, \lambda'\right) \,, \tag{2.20}$$

$$\bar{A}_{ij} = \frac{\sum_{t=2}^{T} P\left(q_{t-1} = i, q_t = j \mid O, \lambda'\right)}{\sum_{t=2}^{T} P\left(q_{t-1} = i \mid O, \lambda'\right)} \,, \tag{2.21}$$

$$\bar{B}_i\left(z_k\right) = \frac{\sum_{t=1}^{T} P\left(q_t = i \mid O, \lambda'\right) \delta\left(o_t, z_k\right)}{\sum_{t=1}^{T} P\left(q_t = i \mid O, \lambda'\right)} \,. \tag{2.22}$$

In the above equations, the probability $P\left(q_t = i \mid O, \lambda'\right)$ is interpreted as the probability of visiting state i at time t given an observation sequence O and a HMM λ'. This probability is denoted by the variable $\gamma_t(i)$. Also the probability $P\left(q_{t-1} = i, q_t = j \mid h, \lambda'\right)$ can be described as the probability of visiting states i and j at times $t-1$ and t respectively. This probability is denoted by the variable $\xi_{t-1}(i, j)$. In (Rabiner, 1989), it is shown that each of these variables can be efficiently evaluated (in linear time) using the forward

and backward variables $\alpha_t(i)$, $\beta_t(i)$ described earlier in Section 2.5.3.

Equations (2.20), (2.21), and (2.22) are known in the literature as the *HMM parameter re-estimation equations*. Namely, given the parameters of the a priori HMM λ', these equations estimate the parameters of the a posteriori HMM λ. This re-estimation process describe therefore one iteration in the Baum-Welch algorithm. In terms of the variables $\gamma_t(i)$ and $\xi_{t-1}(i,j)$, the HMM parameter re-estimation equations can be written as follows

$$\bar{\pi}_i = \gamma_1(i)\,, \tag{2.23}$$

$$\bar{A}_{ij} = \frac{\sum_{t=2}^{T} \xi_{t-1}(i,j)}{\sum_{t=2}^{T} \gamma_{t-1}(i)}\,, \tag{2.24}$$

$$\bar{B}_i(z_k) = \frac{\sum_{t=1,\, o_t=z_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}\,. \tag{2.25}$$

With respect to the a priori HMM λ', the above equations can be rewritten in a more descriptive form as follows.

$$\bar{\pi}_i = \text{expected number of times of visiting state } i \text{ at time } (t=1)$$

$$\bar{A}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$

$$\bar{B}_i\,(z_k) = \frac{\text{expected number of times in state } i \text{ and observing symbol } z_k}{\text{expected number of times in state } i}$$

Since the Baum-Welch algorithm is an instance of the general EM algorithm, it has the same limitation that it converges to a local maximum for the likelihood function rather than the global one. However, according to Rabiner (1989), either random (subject to stochastic and nonzero value constraints) or uniform initial estimates of $\boldsymbol{\pi}$ and $\boldsymbol{A}$ parameters could give useful reestimates of these parameters. Another problem is that the Baum-Welch algorithm requires assuming a specific number of states.

2.5.5 Links between HMM and probabilistic automata

In general, many syntactic objects including HMMs, *Probabilistic Finite-states Automata (PFA)*, *Deterministic Probabilistic Finite-states Automata (DPFA)*, and *λ-Probabilistic Finite-states Automata (λ-PFA)* have been used to model and generate probabilistic distributions over sets of possible infinite cardinality of sequences. Formal definitions and properties of these objects are given by Vidal et al. (2005a) and Dupont et al. (2005).

Comparing HMM to other such probabilistic devices, it is shown by Dupont et al. (2005) and Vidal et al. (2005b) that HMM is equivalent to Probabilistic Finite Automata

(PFA) and λ-PFA. In other words, these models can be converted to each other, and have therefore the same expressive power. On the other hand, it is shown by Vidal et al. (2005a) that there exist probabilistic distributions that can be generated by PFA but not by other probabilistic models like Deterministic Probabilistic Finite Automata (DPFA). Therefore, HMM and PFA are more expressive than other probabilistic models like DPFA.

2.6 HMM structure determination

As our aim is to approximate the dynamic behaviour of a principal by a HMM, this section provides a review of different approaches to determine the structure of the HMM. Generally, For modelling applications the HMM is estimated from sample data. In other words, all these applications involve learning or adjusting the HMM to such data. A practical and fundamental issue to be addressed when using HMM is the determination of its structure, namely the topology (the non-zero transitions and emissions) and the number of states. In this section we review the approaches used to tackle this problem.

2.6.1 HMM structure induction by Baum-Welch algorithm

One can think of the problem of determining the topology of an HMM as estimating its state transition and emission probabilities assuming the model is fully connected. The transition/emission which is estimated to be of zero probability (within an error range) indicates the absence of such transition/emission. The Baum-Welch algorithm, described by Section 2.5.4, and also by Rabiner (1989), can be used for estimating the HMM parameters. However this algorithm converges to a local optimum, not necessarily the global one, of the data likelihood function. Since the convergence optimum is highly dependent on the choice of initial parameter values, one approach is to run the algorithm starting from many different initial points in the parameter space, and choose the best estimate. However, this approach is computationally expensive since the parameter space is large due to the number of free parameters in the case of fully connected HMM. It is worth noting that Baum-Welch algorithm assumes knowledge of the model size, and therefore a method is still needed for determining the appropriate number of states.

2.6.2 Akaike and Bayesian information criteria

The problem of the HMM structure determination can be seen as a model selection problem; thus the traditional model selection approaches AIC (Akaike, 1974), and BIC (Schwarz, 1978) described previously in Section 2.3 are applicable. These methods are based on trading off the likelihood of given data against the model complexity. In the

case of HMM model selection, the complexity of a model depends on its size, namely its number of states. Therefore these methods can be applied to HMM by training several models, with different sizes to maximise data likelihood, and then choosing the one which maximises a certain selection criterion. The main drawback of these approaches is that they are computationally expensive since at least one full training session is required for each candidate model size.

2.6.3 HMM induction by Bayesian model merging

This method is based on a general model inference approach called *Best-First model merging* (Omohundro, 1992). In this approach, a complex model is constructed by combining simple component models. Each of the component models fits a portion of the available data. The whole model is simplified (generalised) by selectively merging two of its component models. The *best-first* aspect is to always choose to merge the pair of component models such that the loss in the data likelihood is minimised. The process of merging component models is repeated until a stopping criterion is met.

Stolcke and Omohundro (1993) described the application of this approach to HMM structure learning. The learning process is performed as follows.

- An initial model which exactly generates the available symbol strings is constructed. In this initial model, each input string is represented by a unique state path with one state per symbol. Each state emits the corresponding symbol with probability 1. Between any successive states in any path, there exists one transition whose probability is 1. The state paths are reachable from the start state with a uniform probability distribution. The likelihood of data under this initial model is 1.

- The current model is incrementally generalised by merging selected state pairs. As the current model is generalised, the likelihood of data is decreased. States chosen for merging are those whose merging maximises the model posterior probability $P(M \mid x)$ which is defined by the Bayes' rule

$$P(M \mid x) \propto P(x \mid M) P(M).$$

  Where $P(x \mid M)$ is the likelihood of data x, and $P(M)$ is the model prior probability. The prior $P(M)$ is chosen to be biased toward simple models; therefore maximising the posterior implies trading off the data likelihood against model simplicity.

- When samples are newly available, the current model structure is adjusted to incorporate the new data, and then generalised by state merging.

- The model induction stops when any potential state merge results in decreasing the model posterior.

One advantage of this approach is that HMM structure learning is performed in an incremental way. The model size is adjusted when new observations are available. By comparison, traditional model selection methods estimate models of different sizes from scratch in order to select the best one. However the disadvantage of this approach is that it finds a local optima for the model posterior probability function rather than the global optima.

2.6.4 Designing a minimal HMM structure by bisimulation

This approach, introduced by Bicego et al. (2001), is based on identifying the number of states in the minimal model by applying the notion of probabilistic bisimulation given by Larsen and Skou (1991) as follows.

Given the data strings drawn from the alphabet $V = \{v_1, v_2, \ldots, v_M\}$,

1. Train an initial HMM with the number of states N reasonably large with respect to the given application. This number is determined using application-specific heuristics. Let $\lambda = (A, B, \pi)$ be the resulting HMM.
2. Transform the resulting model into a *Fully Probabilistic Labeled Transition System (FPLTS)* G by the following procedure.
   - The set of states in G is the same set of states in λ.
   - Each transition between two states S_i, S_j in λ where $a_{ij} > 0$ is replaced by M edges whose labels are $\langle a_{ij}, v_k, B_i(k)\rangle$, where $B_i(k)$ is the probability of emitting the symbol v_k from state S_i.

   In this conversion each of $a_{ij}, B_i(k)$ is approximated to the nearest one of defined probability levels between 0 and 1. This probability quantization expresses the accuracy of comparisons between states.
3. Run the probabilistic bisimulation algorithm, given by Baier et al. (2000), on G to compute bisimulation equivalence classes. The optimal number of states N' in the target HMM is the number of equivalence classes in G.
4. The optimal HMM is finally obtained by training an HMM with N' states.

Since this approach is based on identifying equivalence between states, the resulting minimal HMM model is significantly close to the initial large one in terms of the data likelihood, and the difference depends entirely on the probability approximation. Therefore the resulting model has almost the same classification power as the large model.

Comparing this approach to the BIC method, it has been proved experimentally that in problems with a small alphabet (e.g. DNA modelling), this approach is faster since it needs only two training sessions.

2.7 Stationarity, mixing and ergodicity

In this section we provide a brief description for the notion of a *random process* as a generalisation of markov chains and hidden markov models. We describe the properties of stationarity, mixing, and ergodicity which may hold for a random process. The definitions of these properties are given along with the conditions on a random process to enjoy each of these properties. We will then state the *ergodic theorem*, and show that it applies to a HMM under certain conditions on its underlying DMC. This result is of special importance in our analysis of trust models in the following chapters as the HMM is adopted as a model for the principal's behaviour. More details of these properties for random processes can be found in e.g. (Grimmet and Stirzaker, 2001) and (Norris, 1997). Wider interpretations and applications of these properties in terms of the *ergodic theory* are described in e.g. (Petersen, 1990) and (Billingsley, 1965).

2.7.1 Random processes

A *random process* Y, expressed as $\{Y(t) : t \in T\}$, is a sequence of random variables indexed by a set T which represents the *time*. Each random variable $Y(t)$ represents an observation at the time instant $t \in T$. The value of $Y(t)$ is drawn from the sample set S. The set of all possible sequences of observations is denoted by S^T.

The process Y is called *continuous-time* random process if the indexing set T is the set of real numbers $\mathbb{R}$, while it is called *discrete-time* random process if T is chosen as the set of integer numbers $\mathbb{Z}$. In the following text we confine ourselves to the *discrete time* processes. Typical examples of a random process include the finite-state discrete-time markov chain (DMC), where S is the finite set of its states, and also a discrete-time HMM, where S is the set of observations.

A simple event A of the random process Y is specified by restricting the values of $Y(t)$ at particular time instants. Thus the event A is formally represented by the following *cylinder set*.

$$A = \{Y(i_1) = j_1, Y(i_2) = j_2, \cdots, Y(i_n) = j_n\}, \tag{2.26}$$

which represents the event of observing the values $j_1, j_2, \ldots, j_n$ at times $i_1, i_2, \ldots, i_n$ respectively. Composite events can be expressed by applying the set operations (union, intersection, and complement) on cylinder sets of the above form.

The *time shift* transformation τ is a mapping from the set of events to itself. That is τ transforms an event A into another event, denoted by $\tau(A)$ by shifting the valuation restriction of the random variables $Y(t)$ one step *'backward'* in time. Given the event A expressed by Equation (2.26), the transformation τ is defined by the following equation.

$$\tau(A) = \{Y(i_1 - 1) = j_1, Y(i_2 - 1) = j_2, \cdots, Y(i_n - 1) = j_n\}. \tag{2.27}$$

As a notation, the event $\tau(\tau(A))$ is denoted by $\tau^2(A)$, and in general the event $\tau(\tau^k(A))$ is denoted by $\tau^{k+1}(A)$ for $k \geq 0$. Thus $\tau^h(A)$ is simply the event A *shifted* backward h times where $h \geq 0$.

$$\tau^h(A) = \{Y(i_1 - h) = j_1, Y(i_2 - h) = j_2, \cdots, Y(i_n - h) = j_n\}. \tag{2.28}$$

It also follows that the inverse transformation τ^{-1} shifts an event A *'forward'* one step in time. That is

$$\tau^{-1}(A) = \{Y(i_1 + 1) = j_1, Y(i_2 + 1) = j_2, \cdots, Y(i_n + 1) = j_n\},$$

and in general, for $h \geq 0$,

$$\tau^{-h}(A) = \{Y(i_1 + h) = j_1, Y(i_2 + h) = j_2, \cdots, Y(i_n + h) = j_n\}.$$

Definition 2.9 (invariant event)**.** An event A is called *invariant* if it holds that

$$\tau^{-1}(A) = A = \tau(A).$$

That is an event A is invariant if it does not change when shifted. Given a DMC Y having a set of S, examples of invariant events are given below.

1. The event A_1 of converging to a state s, expressed by the following equation, is invariant because it does not change if it is time-shifted by τ.

$$A_1 = \left\{\lim_{t\to\infty} Y(t) = s\right\}.$$

2. Consider the event A_2 that the sequence observations $Y(t)$ are asymptotically (when $t \to \infty$) restricted to members of the set $R \subset S$. The event A_2, expressed as follows, is invariant.

$$A_2 = \{Y(t) \in R \quad \text{for all large } t\}.$$

   In particular, if $R = S$, the event A_2 occurs with probability 1, and therefore is called *a trivial event*.

Since an invariant event does not change by shifting it in time, it can be seen as the

event which specifies a condition on all observations in a sequence regardless of time. In the case of a DMC, event A_2, for instance, describes the condition that the whole sequence converges to a set of states R, i.e. the observed states are eventually trapped in a set R of states (called a *recurrent class*). This convergence should be seen as a condition which applies to the whole sequence regardless of time.

2.7.2 Stationarity

Definition 2.10 (stationary process)**.** A random process $\{Y(t) : t \in T\}$ is *stationary* if for all events A, it holds that

$$P\left(\tau^{-k}(A)\right) = P(A) = P\left(\tau^{k}(A)\right)$$

for any $k \geq 0$.

That is for a stationary random process, the probability of any event is invariant with respect to the time shift τ. This means that any finite sequence has the same probability whether it is observed at time t or $t + k$, where $k \geq 0$. Examples of stationary random processes include stationary DMC, and A HMM with underlying stationary DMC. For both these models, the probability of the event

$$\{Y(k) = j_0, Y(k+1) = j_1, \cdots, Y(k+n) = j_n\}$$

depends only on the stationary distribution π, given by Definition (2.4), and the elements of the given sequence $j_0, j_1, \cdots, j_n$ regardless of the starting time k.

2.7.3 Mixing

Now we move to another property for the random processes, the *mixing* property defined as follows.

Definition 2.11 (mixing process)**.** A stationary random process $\{Y(t) : t \in T\}$ is called *mixing* if for any two events A and B it holds that,

$$\lim_{k\to\infty} P\left(A \cap \tau^{-k}(B)\right) = P(A)\,P(B).$$

Note that the above condition for mixing can be also written as

$$\lim_{k\to\infty} P\left(\tau^{-k}(B) \mid A\right) = P(B).$$

In the above formulation, $\tau^{-k}(B)$ represents particular conditions on the future observations (at time k), while A represents conditions on the present or past observations.

Therefore, a random process is mixing if conditions on the future observations tends to be probabilistically independent of initial conditions in the past. It can be easily shown that an aperiodic irreducible DMC is mixing using the fact of the convergence of the transition matrix $\boldsymbol{A}^k$ (See Theorem (2.7)). In the following, an analogous statement for HMMs is given.

Proposition 2.12 (HMM mixing)**.** *A stationary finite-state HMM* $(S, V, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ *is mixing if the transition matrix* $\boldsymbol{A}$ *is irreducible and aperiodic.*

Proof. Let Y be the random process generated by the the HMM $(S, V, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$. Consider the following two events of Y

$$\begin{aligned} A &= \{Y(1) = u_1, Y(2) = u_2 \dots Y(n) = u_n\}, \\ B &= \{Y(1) = v_1, Y(2) = v_2 \dots Y(m) = v_m\}. \end{aligned}$$

If we choose k such that $k \geq n$, the time ranges of A and $\tau^{-k}(B)$ do not overlap. The joint event $A \cap \tau^{-k}(B)$ can then be expressed as follows

$$\begin{aligned} A \cap \tau^{-k}(B) = \{&Y(1) = u_1, Y(2) = u_2, \cdots, Y(n) = u_n, \\ &Y(k+1) = v_1, Y(k+2) = v_2, \cdots, Y(k+m) = v_m\}. \end{aligned} \tag{2.29}$$

As mentioned in Section 2.4.1, the probability of reaching state j from state i in exactly k transitions is given by the ijth entry of the matrix $\boldsymbol{A}^k$, denoted by A^k_{ij}. Using Equation (2.8), and rewriting the summation, we get the following expression for the joint probability $P\left(A \cap \tau^{-k}(B)\right)$.

$$\begin{aligned} &\sum_{q_1} \pi_{q_1} \cdot B_{q_1}(u_1) \cdot \sum_{q_2} A_{q_1 q_2} \cdot B_{q_2}(u_2) \cdots \sum_{q_n} A_{q_{n-1} q_n} \cdot B_{q_n}(u_n) \cdot \\ &\cdot \sum_{q_{k+1}} A^{(k-n+1)}_{q_n q_{k+1}} \cdot B_{q_k}(v_1) \cdot \sum_{q_{k+2}} A_{q_{k+1} q_{k+2}} \cdot B_{q_{k+2}}(v_2) \cdots \\ &\cdot \sum_{q_{k+m}} A_{q_{k+m-1} q_{k+m}} \cdot B_{q_{k+m}}(v_m). \end{aligned} \tag{2.30}$$

For simplifying the above expression we need to introduce some notations. Let N be the size of the set of states S. For any observation $v \in V$, we define the $N \times N$ diagonal matrix $\boldsymbol{\Phi}(v)$, and the column vector $\boldsymbol{\Phi^c}(v)$ as follows

$$\boldsymbol{\Phi}(v) = \begin{bmatrix} B_1(v) & 0 & \dots & \dots \\ 0 & B_2(v) & \dots & \dots \\ \vdots & \dots & B_{N-1}(v) & \vdots \\ \dots & \dots & \dots & B_N(v) \end{bmatrix} \tag{2.31}$$

$$\mathbf{\Phi^c}(v) = \begin{bmatrix} B_1(v) \\ B_2(v) \\ \vdots \\ B_N(v) \end{bmatrix} \tag{2.32}$$

Using these notations the probability of $A \cap \tau^{-k}(B)$ can be expressed by rewriting Expression (2.30) as a product of matrices.

$$\begin{aligned} P\left(A \cap \tau^{-k}(B)\right) = \boldsymbol{\pi} \cdot \mathbf{\Phi}(u_1) \cdot \boldsymbol{A} \cdot \mathbf{\Phi}(u_2) \cdots \boldsymbol{A} \cdot \mathbf{\Phi}(u_n) \cdot \\ \cdot \boldsymbol{A}^{(k-n+1)} \cdot \mathbf{\Phi}(v_1) \cdot \boldsymbol{A} \cdot \mathbf{\Phi}(v_2) \cdots \\ \cdot \boldsymbol{A} \cdot \mathbf{\Phi^c}(v_m)\,. \end{aligned} \tag{2.33}$$

Taking the limit when $k \to \infty$, we get the following equation.

$$\begin{aligned} \lim_{k\to\infty} P\left(A \cap \tau^{-k}(B)\right) = \boldsymbol{\pi} \cdot \mathbf{\Phi}(u_1) \cdot \boldsymbol{A} \cdot \mathbf{\Phi}(u_2) \cdots \boldsymbol{A} \cdot \mathbf{\Phi}(u_n) \cdot \\ \cdot \left(\lim_{k\to\infty} \boldsymbol{A}^{(k-n+1)}\right) \cdot \mathbf{\Phi}(v_1) \cdot \boldsymbol{A} \cdot \mathbf{\Phi}(v_2) \cdots \\ \cdot \boldsymbol{A} \cdot \mathbf{\Phi^c}(v_m)\,. \end{aligned} \tag{2.34}$$

By Theorem (2.7), the matrix $\boldsymbol{A}^{(k-n+1)}$ converges as $k \to \infty$ if $\boldsymbol{A}$ is irreducible and aperiodic. In this case each row in the matrix $\lim_{k\to\infty} \boldsymbol{A}^{(k-n+1)}$ is identical to the stationary distribution vector $\boldsymbol{\pi}$. Thus

$$\begin{aligned} \mathbf{\Phi}(u_n) \cdot \left(\lim_{k\to\infty} \boldsymbol{A}^{(k-n+1)}\right) &= \begin{bmatrix} B_1(u_n)\cdot\pi_1 & B_1(u_n)\cdot\pi_2 & \dots & B_1(u_n)\cdot\pi_N \\ B_2(u_n)\cdot\pi_1 & B_2(u_n)\cdot\pi_2 & \dots & B_2(u_n)\cdot\pi_N \\ \vdots & \dots & \dots & \vdots \\ B_N(u_n)\cdot\pi_1 & B_N(u_n)\cdot\pi_2 & \dots & B_N(u_n)\cdot\pi_N \end{bmatrix} \\ &= \mathbf{\Phi^c}(u_n) \cdot \boldsymbol{\pi}. \end{aligned} \tag{2.35}$$

Substituting Equation (2.35) in Equation (2.34) it follows that

$$\begin{aligned} \lim_{k\to\infty} P\left(A \cap \tau^{-k}(B)\right) = \boldsymbol{\pi} \cdot \mathbf{\Phi}(u_1) \cdot \boldsymbol{A} \cdot \mathbf{\Phi}(u_2) \cdots \boldsymbol{A} \cdot \mathbf{\Phi^c}(u_n) \cdot \\ \cdot \boldsymbol{\pi} \cdot \mathbf{\Phi}(v_1) \cdot \boldsymbol{A} \cdot \mathbf{\Phi}(v_2) \cdots \cdot \boldsymbol{A} \cdot \mathbf{\Phi^c}(v_m) \\ = P(A) \cdot P(B). \end{aligned}$$

□

2.7.4 Ergodicity

An ergodic process is defined in terms of invariant events defined by Definition (2.9).

Definition 2.13 (ergodic process)**.** A random process is called *ergodic* if the probability of each invariant event is either 0 or 1.

Any event with probability 0 or 1 is called 'trivial', as it either 'always' or 'never' occurs. Therefore according to the above definition, the ergodic process is the one where each invariant event (describing a time-invariant condition on the whole sequence of observations) is trivial. An important relationship between the ergodicity and mixing properties is stated by the following proposition.

Proposition 2.14. *Any mixing process is ergodic.*

Namely, for any invariant event B, it holds that $\tau^{-k}(B) = B$, for all $k \geq 0$. By Definition (2.11), if a process is mixing then $P(A \cap B) = P(A)\,P(B)$. Taking $A = B$ it follows that $P(B) = (P(B))^2$ which implies that $P(B)$ is either 0 or 1.

Theorem 2.15 (ergodic theorem)**.** *For a stationary and ergodic random process Y, let f be a real-valued function defined on the set S^T of sequences generated by Y. Given a sequence $\boldsymbol{x} \in S^T$, let $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots$ be a sequence of functions defined by*

$$f_k(\boldsymbol{x}) = f\left(\tau^k(\boldsymbol{x})\right).$$

If $\left|\mathbf{E}[f]\right| < \infty$, then

$$\frac{1}{n}\sum_{k=0}^{n-1} f_k(\boldsymbol{x}) \to \mathbf{E}[f] \qquad \textit{almost surely.}$$

In the above formulation for the ergodic theorem, $\frac{1}{n}\sum_{k=0}^{n} f_k(\boldsymbol{x})$ is called the *'time average'* of the function f evaluated on given sequence $\boldsymbol{x}$. The time average is therefore a characteristic of the underlying sequence $\boldsymbol{x}$. On the other hand, the expected value $\mathbf{E}[f]$ is called the *'space average'* of the function f, and therefore is a characteristic of the probability distribution over all potential sequences. The ergodic theorem establishes the almost sure convergence of the time average to the space average when the random process is ergodic.

In our concern about the problem of evaluating trust in networked principals, the sequence of outcomes observed over time by interacting with a principal is seen as a random process, which reflects the dynamic behaviour of the principal. A trust evaluation algorithm can be therefore seen as a function of such an observed sequence. Opting to use ergodic HMMs for modelling principals' behaviours, the ergodic theorem plays an important role in analysing formal properties of trust models. In particular, this theorem will be used to evaluate the *beta-based trust* in terms of the parameters of the principal's HMM, as shown in Chapter 5.

Chapter 3

State of the art

Security preservation in a network of principals is often associated with mechanisms which ensure that principals conform to specified access control policies when accessing resources belonging to each other. These policies are aimed to preventing, or at least minimising, the abuse of such resources. To see the tight link between security and trust, consider a principal A requesting to access a resource r owned by another principal B. Assuming that the resource r is precious to the principal B, the access to r is granted by B to the requesting principal A only if B has a sufficient confidence that A will not abuse or destroy r; that is if B *trusts* A. Here many questions arise about this new notion of trust between network principals. These questions address for example the problems of formulating, computing, and updating the trust in a given principal. They also address the problem of possible mechanisms of communicating trust information between principals in a form of *reputation reports*.

In this chapter, different approaches to trust management are described. We start in Section 3.1 by describing the traditional *credential based trust*, where a principal is trusted to access a resource if sufficient credentials (proofs) entitling the principal to access the resource are available. Subsequently, Section 3.2 describes the more recent approach of probabilistic trust, where the trust in a principal is modelled by a probability distribution over potential outcomes of interactions with the principal.

3.1 Credential based trust

Trust in a principal is traditionally viewed as an entire confidence that a principal is entitled to access a particular resource. This confidence is based on a proof of entitlement, called a *credential*, which is issued to the principal by a trusted authority. That is a principal is trusted to access a resource if it is able to provide a sufficient amount of credentials which proves its entitlement to access that resource. The access of any resource

is usually controlled by a specific policy which determines the necessary credentials in order to grant the resource access to a requesting principal, i.e. to establish trust in a principal to access the resource. This is why this view of trust is known as *credential based trust* and also *policy based trust*. This view of trust has a 'binary' nature in the sense that a principal is either trusted to access a resource or not. This differs from the 'gradual' view of trust which defines a range of trust levels as described later in this chapter.

In this section we give an overview of the credential based trust. We first describe the concepts associated with credentials, including their forms, issuers, and the mechanisms of exchanging policies and credentials between interacting parties. Existing trust management systems based on credentials are then described. Afterwards, typical aspects of these system are discussed. In particular, we describe the notion of *trust negotiation* to establish trust between interacting parties through exchanging policies and credentials while minimising the disclosure of sensitive or private information about these principals. Languages for specifying policies and credentials are also discussed.

3.1.1 Network security credentials

Digital credentials (or simply *credentials*) can be described as statements concerning a principal which are issued by trusted third parties (organizations) and shown to other principals (or organizations) (Chaum, 1985). Digital credentials are therefore similar to paper credentials that people carry in their wallets (e.g. tickets, passports, licenses) in the sense that they prove (certify) certain attributes assigned to a principal, or the entitlement of the principal to particular rights.

A credential is usually formulated in the form of an *attribute certificate* (AC), where a *certificate authority* issues a (signed) certificate to a principal (called the *subject*) stating a particular attribute. In this way, a subject is trusted to have a particular attribute only if a valid attribute certificate stating such attribute possession is available and digitally signed by the appropriate authority. The validity of attribute certificates is assured by ensuring, using cryptographic mechanisms, that the certificates are really produced and signed by a trusted attribute authority.

While attribute certificates provide a means of authorization, i.e. determining which principals can access individual resources, another type of certificates, known as *identity certificates* certify the identities of subjects, and are issued by *identity authorities*. An identity certificate includes both the subject's identity information (name, address, etc.) and the subject's public key. The identity certificates provide a means of authenticating messages' senders. Consider, for example, a message which is claimed to be sent by Bob. Assuming that such message is digitally signed using the sender's private key, a receiver Alice can use Bob's public key, which is certified by a valid identity certificate to prove or

disprove that Bob is the real message sender. A remarkable application which employs identity certificates is the *Pretty Good Privacy (PGP)* (PGPi website, 2000), which provides secure data exchange (especially emails). In addition to its encryption/decryption functions to preserve the confidentiality of a message, it employs the identity certificates to provide the function of authenticating the sources of messages.

For formalising digital certificates, Rivest and Lampson (1996) introduced SDSI, a simple infrastructure for identity certificates where public keys of principals are bound to unique principal names. Rather than assuming the availability of a global directory of unique principal names, SDSI allows for *local name spaces* where principals create their local names to others. Local names can be linked together to form a longer name referring to some principal. If for example "smith" refers to a principal by the local name "bob", who refers to another principal by the local name "alice", then "smith" can refer to the latter by `(ref:bob alice)`.

Ellison et al. (1999) then proposed a variant model called SPKI which extends SDSI capabilities to allow for attribute certificates in addition to identity certificates. The two models (SDSI and SPKI) are merged together and currently known as SDSI/SPKI. While identity certificates bind subjects' names to their public keys to allow for authentication, the attribute certificates bind attributes (or authorities) to the subjects' names for the purpose of authorization. In some cases both authentication and authorization functions are required to grant a principal the access to a resource. Here the attribute certificate provided by a requester must be linked (bound) to the requester's identity certificate (which may be issued by a different authority). Different approaches of binding the attribute and identity certificates are described by Park and Sandhu (2000).

In SDSI/SPKI framework (Ellison et al., 1999), an attribute certificate is also called *authorization certificate* if the certified attribute assigned to a subject represents a permission to perform a specific task. In particular, this permission can allow the holder to issue further authorization certificates to other principals. This implements the notion of 'delegating' an authority from one principal to another. With this mechanism of authority delegation, a set of credentials (certificates) can be linked in a *credential chain*, in which a credential is issued by a certifying authority A to a subject B allowing her to issue another credential to C. By tracing this chain backward to the trusted certifying authority A, the credential held by C is verified and hence, trust can be established in C. In other words, the credential chains provides a means for transferring trust transitively to principals.

3.1.2 Centralised trust management systems

A simplistic approach to establish trust between principals is to employ a (trusted) central server mediating between interacting parties. The job of this central server is to

issue the essential credentials to its registered principals. A traditional protocol which adopted this approach is Kerberos (Neuman and Ts'o, 1994), which was originally designed to enable principals to authenticate their partners. According to this protocol, a client wanting to interact with a server submits a request to a central "authentication server", which generates a session key and encrypt it twice to produce two separate credentials called "tickets" both sent to the client. One of these tickets can be decrypted only by the client while the other can by decrypted only by the server. The client decrypts its ticket to uncover the session key, and sends the other ticket to the server which extracts the session key. Sharing the same session key between the client and server, enables them to verify the identity of each other by exchanging messages encrypted by the shared session key.

While the central trust management approach implemented by Kerberos satisfies the requirement of establishing trust between interacting parties by exchanging credentials, this approach suffers from the dependency on a central server (the authentication server), which plays the role of a global certifying authority. Another framework which also relied on central trusted authorities for verifying identities is X.509, which was introduced by the International Telecommunication Union (ITU). In its update (ITU-T, 1993), the X.509 authentication framework made a step towards flexibility by basing the trust establishment on credential chains (also called 'certification paths') originating from the trusted authorities rather than single credentials issued directly by these authorities. Despite this remarkable improvement, the assumption of central authorities is still essential for verifying credential chains.

3.1.3 Decentralised trust management systems

Coping with the limitation of reliance on central authorities, a system called *'Pretty Good Privacy'* (PGP) was developed by Zimmermann (PGPi website (2000)) to establish trust between principals exchanging emails. In PGP, each principal A arbitrarily specifies its own basic set of trusted principals by holding their identity certificates (possibly signed by them). Any of these trusted principals e.g. B can also 'introduce' another principal C to A (by issuing an identity certificate for C). As A trusts B, it can also trust C. Therefore, trust management in PGP is 'decentralised' in the sense that the trust in a principal is assessed based on considering mutual trust between principals rather than seeking credential chains originating from central authorities.

While PGP could successfully avoid the restriction of central authorities, it is designed for a specific application, that is to provide privacy in email exchange. A trend of research has then directed towards building trust management systems which can be used rather by many applications, and are also decentralised. This problem was first addressed by Blaze *et al.* who developed a system called PolicyMaker (Blaze et al., 1996, 1998). PolicyMaker is a language enabling each principal to specify verified credentials, and

also policies which describe the conditions of trusting other principals. The function of PolicyMaker is then to answer queries about the trust in a given principal using existing policies and credentials. Following PolicyMaker, its authors developed a variant and simpler system called KeyNote (Blaze et al., 1999). The simplicity of KeyNote is achieved by identifying principals by public keys rather than names. The problem of binding public keys to principal names is assumed to be resolved outside KeyNote.

3.1.4 Trust negotiation

By definition, a credential implemented as an identity or attribute certificate, contains information about the identity of the subject. Therefore, this private information is disclosed when the credential is exchanged. Furthermore, when the principal issues a request to access a resource belonging to a server, the server usually provides the requester with the resource access policy. The disclosure of the policy itself can involve disclosing sensitive information about the server. For example a server which announce 'no access' during a period of time due to maintenance may reflect the vulnerability of this server to attacks as some protection mechanisms may be disabled in this period.

It would be an important requirement for both the client and server to disclose as little information as possible (including policies and credentials) to achieve a successful interaction. The process of establishing the sufficient trust between two parties is known as *trust negotiation* (Winsborough et al., 2000). It is an iterative process of exchanging policies and credentials between two parties to establish mutual trust between them and hence complete a transaction.

The notion of trust negotiation was initially introduced by Winsborough et al. (2000), which considers different potential strategies that negotiating principals can use for exchanging their credentials. In this work, a credential is always protected by a *credential access policy (CAP)*. That is a negotiating principal can disclose a credential to its negotiation partner only if the credential's associated CAP is satisfied. In this respect, two strategies are presented.

1. The *eager* strategy, where the principals turn over all their credentials as soon as their CAPs are satisfied, without waiting for these credentials to be requested. While this strategy is simple and efficient, it suffers from the drawback that a trust negotiation can involve disclosing unnecessary credentials.

2. The *parsimonious* strategy, which is summarised as follows. A principal A receives a request for a credential C_A from the other principal B. After checking the C_A's CAP, the principal A responds by requesting only the credentials C_B (owned by B) whose disclosures are necessary and sufficient to satisfy the CAP of the previously requested credential C_A. While this strategy is also efficient and tends to minimise

the credential disclosures (on 'need-to-know' basis), it has the drawback of the necessity of exchanging requests, which may again reveal sensitive information about principals.

Considering the sensitivity of both credentials and access policies, a system called TrustBuilder was presented and prototyped by Winselett *et al.* in Winslett et al. (2002). This system establishes trust between two parties by devoting a *security agent* to each party for handling trust negotiation. A security agent is composed of the following three basic components:

1. *negotiation strategy module*, whose function is to determine the next policy or credential to be disclosed to the other party according the current phase of negotiation.
2. *policy compliance checker*, which determines the necessary credentials required to satisfy a policy disclosed by the other party.
3. *credential verification module*, which verifies the other party's credentials. This involves signature validation, revocation check, and credential chain discovery if needed.

A more recent system implementing trust negotiation is PeerTrust (Nejdl et al., 2004). In this system, the authors have developed a trust negotiation language which facilitates describing policies and credentials as *first order Horn rules* of the following form.

$$lit_0 \leftarrow lit_1, \ldots, lit_n$$

where lit_i is a predicate $\mathbf{P}(t_1, \ldots, t_m)$ which may state an attribute of a principal (peer). The trust negotiation process amounts therefore to resolving together the set of rules (policies and credential) available to a principal to prove or disprove the trustworthy of the interacting peer to access a resource. This recursively involves querying only the relevant rules, which could be provided by different peers. Therefore not more than sufficient policies/credentials are disclosed to establish trust between two interacting peers. Note here that the trust negotiation process does not necessarily involve only the two interacting parties, but can also involve other parties, e.g. the certifying authorities.

Following the same approach of representing policies by logic rules, another trust negotiation framework called PROTUNE has been introduced by Bonatti and Olmedilla (2005). The rules of PROTUNE allow for a remarkable class of predicates called 'provisional predicates'. Unlike other classes of predicates which are evaluated by querying the current negotiation state of the negotiating peer (the set of true-valued predicates), a provisional predicate can be made true by performing appropriate actions. An example of provisional predicates is $\mathbf{credential}(C, K)$ which is evaluated to true if the negotiating peer already has a credential C signed by a principal whose public key is K; if not,

the predicate **credential**(C, K) can still be made true by asking the appropriate principal to provide the credential C. Another example is **do**(*service_request*), which is made true if the peer successfully completes an application dependent procedure initiated by invoking *service_request*.

3.1.5 Security policies and trust languages

While the systems described above in Section 3.1.4 are addressing the problem of formulating policies and credentials, they also describe the mechanisms of exchanging them in the process of trust negotiation such that the disclosure of sensitive information is minimised. Independently, some substantial works have been focused solely on the problem of formulating security policies resulting in different languages for expressing policies.

A notable system in this area which has been presented by Uszok and his colleagues is KAoS (Uszok et al., 2003). KAoS provides a set of tools and services intended to specify, manage, and enforce policies in application domains, where the entities and actions in the given domain, and also the policies themselves are described as an ontology. In this ontological representation, a policy is represented as an instance of a particular policy type with values assigned to the relevant properties, e.g. the action class controlled by the policy, the policy priority, etc. Basing the definition and management of policies on ontological representation of the application domain gives KAoS an advantage of adaptability to different domains.

Rather than using the ontology representation as a basis of specifying security policies, another approach is to define the policies in the form of logic rules. This approach has been adopted in many systems including PeerTrust (Nejdl et al., 2004) and PROTUNE (Bonatti and Olmedilla, 2005) trust management systems which were briefly described in Section 3.1.4.

In large scale enterprises containing millions of principals, security administration is not as simple as small enterprises. If each assertion of an access permission to a principal is specified by a separate policy, the total number of the enterprise policies can therefore be extremely large to be manageable. Using the fact that many principals are usually asserted the same set of permissions, a common approach to reducing the number of policies is known as *role-based access control (RBAC)* (Sandhu, 1996). A *role* is simply a set of permissions which are together asserted to (or revoked from) an entity. A number of policy specification languages have adopted the RBAC approach, and hence called *role-based* languages. An example of such languages is `Cassandra` (Becker and Sewell, 2004). Instead of having a separate policy for asserting each single permission to a principal, one `Cassandra` policy rule can assign/revoke a set of permissions as one role to/from principals, given some constraints represented as predicates. This separates the definitions of roles themselves from the policies, and therefore enhances

the manageability of the system security.

Independently of expressing security policies, a trend of research has been directed to formalising and expressing the notion of trust itself, that is defining trust values, and specification of algorithmic rules assigning these trust values to principals. In this direction Nielsen and his colleagues have described in (Nielsen and Krukow, 2003), and (Carbone et al., 2003) simple languages which facilitate expressing *trust policies*. Here, a trust policy specifies how the trust in a given principal is evaluated. It is notable in this work that the binary view of trust is generalised. That is, the trust in a principal is not, in general, restricted to *trusted* and *untrusted*, but can be rather drawn from a set of values ordered in a lattice structure. The authors also make a clear distinction between the *trust policy* which specifies how trust is computed, and the *security policy* which specifies the decisions made by a principal given its evaluated trust in other principals.

3.2 Probabilistic models for trust

From the overview in Section 3.1, the credential based trust has a binary nature. That is a principal is either trusted or untrusted to perform an action. Credentials are issued, as proofs of trust, to a principal if its behaviour is well known, by the issuing authorities, to comply with specific security obligations. By exchanging credentials between principals, they can base their mutual interactions on 'proved' knowledge about each other's behaviour. This trust approach is appropriate in closed networks, where principals have sufficient information (through credentials) about their peers.

Nevertheless, in modern, large-scale networks (e.g. the Internet), interacting principals can have autonomously different behaviours and intentions which are incompletely known by each other. This incomplete knowledge available to principals about each other makes credentials not the appropriate evidence of trust because no principal is 'perfectly' trusted, that is guaranteed to conform to interaction-related policies. However, the trustworthiness of a principal can be reflected by the history of its interactions with other principals (interaction history). Based on this idea, an approach of trust has evolved where a principal (truster) *tr* evaluates a quantitative measure for its trust in another principal (trustee) *te* using *te*'s interaction history. Note that the trust value here is not binary as the case in credential-based trust, but rather a number expressing the level of trustworthiness. This view of trust is known as the *computational trust* and also as *reputation based trust*.

In this dissertation we focus on the notion of *probabilistic trust* which subsumes the general category of the computational trust. The probabilistic trust can broadly be characterised as aiming to build probabilistic models for principals' behaviours using the outcomes of historical interactions. Using these models, the trust of a truster *tr* in a trustee *te* is the probability, estimated by *tr*, of particular outcomes of the next

interaction with *te*. This notion of trust resembles the trusting relationship between humans as seen by Gambetta (1988). In the following we discuss the research work done in this area.

3.2.1 Beta trust model

As pointed out in Chapter 1, the most important component of any probabilistic trust model is the behaviour model which is used to estimate the probabilities of future outcomes. According to the *beta trust model* introduced by Jøsang and Ismail (2002), and followed by other works, e.g. Teacy et al. (2006); Mui et al. (2002); Buchegger and Le Boudec (2004), the behaviour of a trustee *te* is modelled by a fixed probability distribution $\mathbf{\Theta}_{te}$ over all possible outcomes of an interaction with *te*. Thus given a sequence of outcomes $h = o_1 \cdots o_n$, the problem of estimating $\mathbf{\Theta}_{te}$ can be solved by Bayesian parameter estimation described in Section 2.2.1, where the parameter in this case is the distribution $\mathbf{\Theta}_{te}$.

In the beta trust model the outcomes are either success s or failure f. Therefore, $\mathbf{\Theta}_{te}$ can be represented by a single probability θ_{te}, the probability that an interaction with the given trustee *te* will be successful. Under the assumption of fixed θ_{te}, a sequence of n outcomes $h = o_1 \cdots o_n$ is a sequence of Bernoulli trials, and the number of successful outcomes in h is probabilistically distributed according to a binomial distribution

$$P(h \text{ consists of } k \text{ successes}) = \binom{n}{k} \theta_{te}^{k} (1 - \theta_{te})^{n-k}.$$

It has been shown in the literature (see e.g. Casella and Berger, 2001) that the beta probability density function (pdf) indexed by the parameters α and β

$$f(\theta_{te} \mid \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\,\Gamma(\beta)}\, {\theta_{te}}^{\alpha-1} (1 - \theta_{te})^{\beta-1},$$

where Γ is the gamma function, is a conjugate prior to the binomial distribution. That is if $f(\theta_{te} \mid \alpha_{pr}, \beta_{pr})$ is chosen as the a priori pdf of θ_{te}, then given a sequence h of outcomes, the resulting a posteri pdf of θ_{te} is $f(\theta_{te} \mid \alpha_{post}, \beta_{post})$, the beta pdf with parameters α_{post} and β_{post}, where the a posteri parameters are related to the a priori ones and the outcome sequence h by the following equations

$$\alpha_{post} = \#_{\mathsf{s}}(h) + \alpha_{pr} \qquad \text{and} \qquad \beta_{post} = \#_{\mathsf{f}}(h) + \beta_{pr},$$

where $\#_{\mathsf{s}}(h)$ and $\#_{\mathsf{f}}(h)$ are the numbers of successful and unsuccessful interactions in h respectively.

Here the estimate for θ_{te}, the probability of having successful interaction, is naturally evaluated as the expected value of θ_{te} according to its a posteri pdf. Using the properties

of the beta pdf, this expected value is given by

$$\mathbf{E}\left[\theta_{te}\right] = \frac{\alpha_{post}}{\alpha_{post} + \beta_{post}} \,.$$

Observe that when the a priori pdf is chosen with parameters $\alpha_{pr} = 1$ and $\beta_{pr} = 1$, the a priori is exactly the uniform pdf which assigns equal likelihood to all values of θ_{te} in the range $[0, 1]$. Such pdf indicates therefore 'unbiased' prior belief about θ_{te}, as no value is more likely than another.

Taking the uniform pdf as the a priori pdf for θ_{te}, the parameters of the a posteriori pdf α_{post}, β_{post} are related to the sequence h of outcomes as follows.

$$\alpha_{post} = \#_{\mathsf{s}}\left(h\right) + 1 \qquad \text{and} \qquad \beta_{post} = \#_{\mathsf{f}}\left(h\right) + 1 \,,$$

and the beta estimate for θ_{te}, which we denote by $\mathcal{B}\left(\mathsf{s} \mid h\right)$, is therefore given by

$$\mathcal{B}\left(\mathsf{s} \mid h\right) = \frac{\#_{\mathsf{s}}\left(h\right) + 1}{\#_{\mathsf{s}}\left(h\right) + \#_{\mathsf{f}}\left(h\right) + 2} \,. \tag{3.1}$$

One advantage of the above model of the trust evaluation is that it encompasses a mechanism for handling the *reputation*, i.e. using feedbacks (called *reputation reports* or *ratings*) provided by other network peers (called *reputation sources*) about the trustee te under consideration to enhance the trust evaluation process. Given the representation of te's behaviour by a fixed probability distribution (parametrised by θ_{te}), each interaction with the trustee is seen as a Bernoulli trial regardless of the interacting partner. The sequence h in the estimating equation (3.1) is correctly seen, therefore, as the sequence of the outcomes of all historical interactions with te regardless of its partners in these interactions. This allows for formulating a reputation report from a reputation source rs as the pair

$$(\#_{\mathsf{s}}\left(h_{rs}\right) \;,\; \#_{\mathsf{f}}\left(h_{rs}\right)). \tag{3.2}$$

where $\#_{\mathsf{s}}\left(h_{rs}\right)$ (respectively $\#_{\mathsf{f}}\left(h_{rs}\right)$) is the count of successful (respectively unsuccessful) interactions between rs and te in the sequence h_{rs} of personal interactions between rs and te. With a set of reputation reports (provided by different reputation sources), a truster can evaluate the ingredients of the beta trust equation (3.1), as follows

$$\begin{aligned} \#_{\mathsf{s}}\left(h\right) &= \sum_{rs} \#_{\mathsf{s}}\left(h_{rs}\right) , \\ \#_{\mathsf{f}}\left(h\right) &= \sum_{rs} \#_{\mathsf{f}}\left(h_{rs}\right) . \end{aligned} \tag{3.3}$$

3.2.2 Dirichlet reputation model

Following the beta model for trust and reputation, Jøsang and Haller (2007) introduced the *Dirichlet reputation model* which is also followed by Nielsen et al. (2007). This model generalises the beta reputation model such that the outcome of an interaction is not restricted to be binary (success or failure), but rather takes a value from any set of discrete rating levels, e.g. {*very bad - bad - average - good - excellent*}.

Given a set $R = \{1, 2, \ldots, k\}$ of possible outcomes, the Dirichlet reputation model keeps the assumption that each trustee te has a fixed behaviour represented by a probability distribution $\boldsymbol{\Theta}_{te}$ over the set R. That is,

$$\boldsymbol{\Theta}_{te} = (\theta_1, \theta_2, \ldots, \theta_k), \tag{3.4}$$

where θ_i denotes the probability that an interaction with te yields outcome i, and $\sum_{i=1}^{k} \theta_i = 1$. Similar to the beta trust model, the Bayesian framework is used to estimate the unknown distribution $\boldsymbol{\Theta}_{te}$. For this purpose $\boldsymbol{\Theta}_{te}$ is seen as a (vector) random variable, for which we seek a (multidimensional) probability density function which can be updated using given observations.

Given that in Dirichlet reputation model a single outcome is not restricted to be binary; a sequence h of outcomes is actually a sequence of independent *multinomial* trials rather than binomial (Bernolli) trials. Given a sequence h of length n, the probability of $\#_1(h), \#_2(h), \ldots, \#_k(h)$ being the numbers of occurrences of outcomes $1, 2, \ldots, k$ respectively in h is given by the *multinomial distribution* defined as follows

$$P(\#_1(h), \#_2(h), \ldots, \#_k(h)) = \frac{n!}{\#_1(h)!\ \#_2(h)!\ \ldots\ \#_k(h)!} \prod_{i=1}^{k} \theta_i^{\#_i(h)}.$$

It is shown in the literature (see e.g. Gelman et al., 2003) that the following Dirichlet probability density function (pdf) indexed by the vector parameter $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_k)$

$$Dir\left(\boldsymbol{\Theta}_{te} \mid \boldsymbol{\alpha}\right) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma\left(\alpha_i\right)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \tag{3.5}$$

is a conjugate prior to the multinomial distribution, and hence the Dirichlet pdf is chosen in the Bayesian framework to model the k-dimensional random variable $\boldsymbol{\Theta}_{te}$. Specifically, if $Dir\left(\boldsymbol{\Theta}_{te} \mid \boldsymbol{\alpha}\right)$ is the a priori pdf of $\boldsymbol{\Theta}_{te}$, and h is a given sequence of observations, then the a posteriori pdf of $\boldsymbol{\Theta}_{te}$ is $Dir\left(\boldsymbol{\Theta}_{te} \mid \boldsymbol{\alpha}'\right)$, the Dirichlet pdf with a posteriori parameter

$$\boldsymbol{\alpha}' = (\alpha_1', \alpha_2', \ldots, \alpha_k'). \tag{3.6}$$

The a posteriori parameter $\boldsymbol{\alpha}'$ is related to the a priori one $\boldsymbol{\alpha}$ and the observation

sequence h by the equation

$$\alpha'_i = \#_i(h) + \alpha_i, \tag{3.7}$$

where $\#_i(h)$ is the number of occurrences of outcome i in h, and $1 \leq i \leq k$. One feature of the parameter updating mechanism described by Eq. (3.7) is that prior information about the pdf can be encoded by setting the a priori parameters $(\alpha_1, \alpha_2, \ldots, \alpha_k)$. In particular, using the setting $(1, 1, \ldots, 1)$ makes the a priori pdf exactly the uniform distribution (see Eq. (3.5)), indicating unbiased prior knowledge about $\boldsymbol{\Theta}_{te}$.

Since $Dir(\boldsymbol{\Theta}_{te} \mid \boldsymbol{\alpha})$ is a function of multi-dimensional vector $\boldsymbol{\Theta}_{te} = (\theta_1, \theta_2, \ldots, \theta_k)$, it is challenging to be visualised when $k > 2$. However for the specific case $k = 3$, Jøsang and Haller (2007) proposed a convenient method for visualising the Dirichlet pdf. This method is based on the observation that the constraint $\theta_1 + \theta_2 + \theta_3 = 1$ on the elements of the variable vector $\boldsymbol{\Theta}_{te}$, defines the domain of $Dir(\boldsymbol{\Theta}_{te} \mid \boldsymbol{\alpha})$ in the 3-dimensional space as an equilateral triangle with the corners $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. Each of these points indicates the extreme bias of the pdf towards one outcome. Thus the pdf can be plotted by laying this triangle horizontally and plotting the pdf vertically. Figure 3.1 shows the plot of this function for different values of the parameter $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$.

From Eq. (3.7), the a posteriori parameter α'_i is obtained by accumulating the number of experienced occurrences of the corresponding outcome i. This also enables formulating a reputation report as the vector

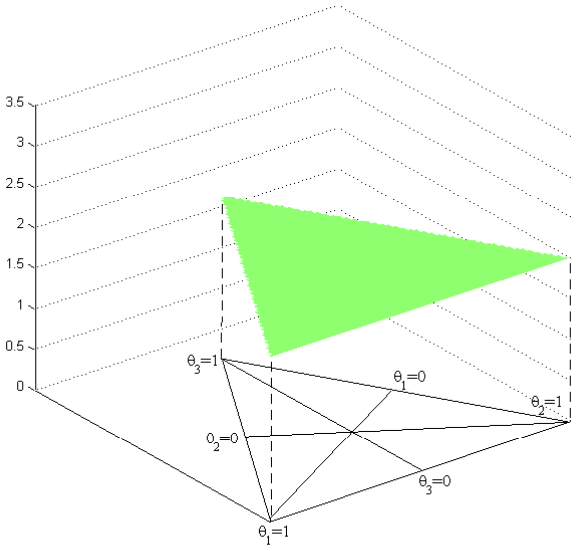
$$(\#_1(h), \#_2(h), \ldots, \#_k(h)). \tag{3.8}$$

Note that the above Dirichlet reputation report generalises the beta reputation report (3.2). A set of reputation reports for a particular trustee te can be then added to each other by a truster, which then uses Eq. (3.7) to get the a posteriori parameters of the trustee's Dirichlet pdf. By the properties of the Dirichlet pdf, the expected value of the k-dimensional random variable $\boldsymbol{\Theta}_{te}$ (described by (3.4)) is evaluated as follows

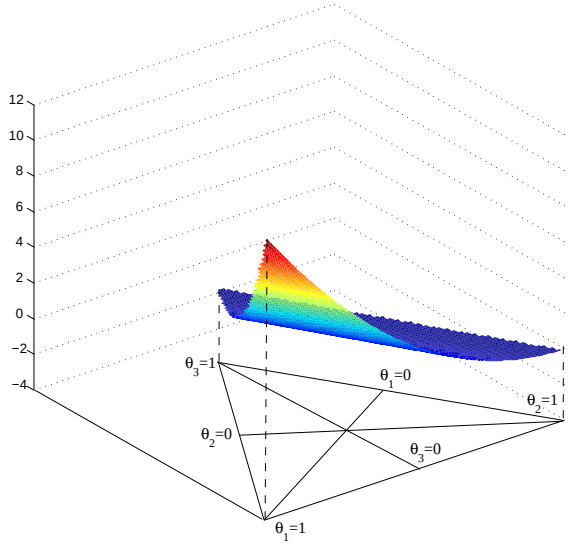
$$\mathbf{E}[\theta_i] = \frac{\alpha'_i}{\sum_{j=1}^{k} \alpha'_j}.$$

3.2.3 The TRAVOS model

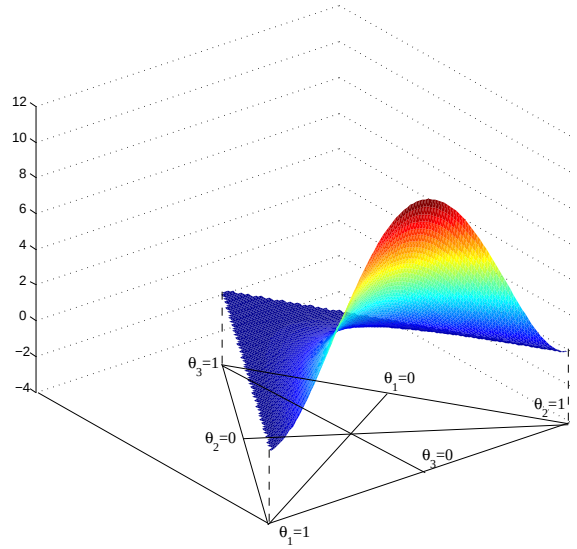
In *TRAVOS (Trust and Reputation model for Agent-based Virtual OrganisationS)*, described by Teacy et al. (2006), the beta trust and reputation model is adopted to evaluate trust between principals in a multi-agent environment. From the perspective of an agent, an interaction with another one has two possible outcomes (success or failure). Namely, an interaction between a truster tr and a trustee te is considered successful by tr if te fulfils its obligations in the interaction. This model assumes that trustees have fixed behaviours, that is te fulfils its obligations with a particular probability θ_{te}. Based on



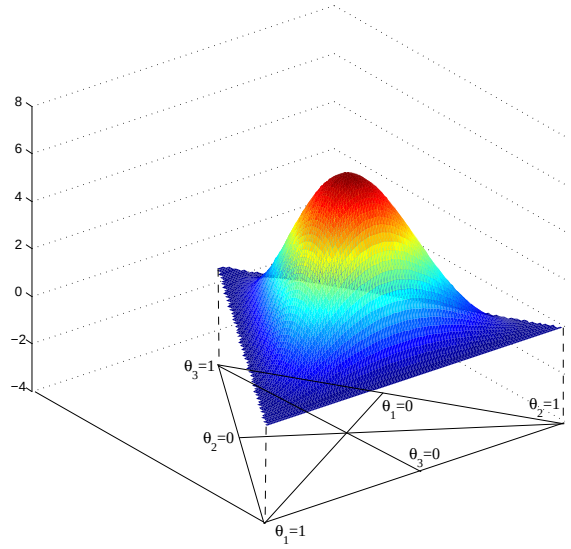
The Dirichlet pdf for $\boldsymbol{\alpha} = (1, 1, 1)$



The Dirichlet pdf for $\boldsymbol{\alpha} = (3, 1, 1)$



The Dirichlet pdf for $\boldsymbol{\alpha} = (3, 3, 1)$



The Dirichlet pdf for $\boldsymbol{\alpha} = (3, 3, 3)$

FIGURE 3.1: The plot of Dirichlet pdf for different values of the parameter $\boldsymbol{\alpha}$

this assumption the trust of the truster tr in te is defined to be an estimated value for θ_{te} from the perspective of tr.

The truster tr uses the outcomes of past interactions with te to evaluate its trust in te. This evaluation is performed using the Bayesian inference. In the inference process (trust assessment), the truster tr uses not only its own observations from direct interactions with the trustee te, but also reputation reports (in the form given by (3.2)) collected from other agents regarding their interactions with te.

While keeping the assumption that each agent has a fixed behaviour (represented by a fixed probability of yielding successful interactions), TRAVOS made a step forward by addressing the problem that a reputation report given by a reputation source about a trustee te may be inaccurate, i.e. the report does not reflect the actual history of interactions between the source and the trustee te. With respect to this issue, TRAVOS advised an approach, where a truster tr estimates the probability that a reputation report is accurate using the source's past opinions (reports) about the trustee te and the actual observations made by the truster itself about te. The truster uses this probability of accuracy to discount the source's report about te. More details about this approach of handling inaccurate reports are given by Teacy et al. (2006).

3.2.4 Event-based trust model

In Event-based trust models introduced by Nielsen et al. (2007), the application of probabilistic trust is generalised to include cases where the outcome of a single interaction between two principals is a sequence of events rather than one event (success or failure). A single interaction is seen as a run of the protocol which governs the interaction between principals. The outcome of an interaction is thus a set of events where each event is assumed to be probabilistically sampled from potential events enabled at a particular stage in the protocol run.

Here, the interaction protocol is formally modelled by the notion of *probabilistic confusion-free event structure* introduced by Varacca et al. (2004). At each particular point in the protocol run, the set of possible events is called *a choice point*. An interaction outcome, called *a configuration*, is simply a set of events, where each event is probabilistically sampled from a choice point during the protocol run. The probability distribution at each choice point over its events is assumed to be immutable over time and therefore, fixed behaviour of principals is assumed.

Based on the above assumption of fixed behaviour, each choice point C is associated with a fixed probability distribution $\mathbf{\Theta}_C$ over the set of events enabled at C. The trust in a principal amounts therefore to estimating the probability distribution associated with each choice point; this enables estimating the probabilities of possible interaction outcomes (configurations).

Given the observations (outcomes) of the past interactions with a trustee, the Bayesian inference framework is used for estimating the distribution $\boldsymbol{\Theta}_C$ associated with each choice point C. Since $\boldsymbol{\Theta}_C$ is defined over a set of multiple events, the procedure of estimating $\boldsymbol{\Theta}_C$ for a single choice point C, associated with a trustee, coincides with the procedure of estimating the parameters of a fixed behaviour trustee using the Dirichlet reputation model (described above in Section 3.2.2). The difference here is that this inference procedure has to be performed for each trustee's choice point.

3.2.5 Exponential decay

It is worth noting that all systems described above have adopted the assumption that the behaviour of each principal (trustee) is 'fixed'; that is a principal's behaviour is modelled by a fixed probability distribution over the possible outcomes of an interaction with such a principal. This assumption simplifies both modelling a principal behaviour (as a probability distribution), and also the process of estimating the parameters of this behaviour model (using Bayesian inference). However, this assumption represents a major limitation in the above systems. In fact, a principal can in general change its behaviour over time. This limitation has been recognised by many works (e.g. Mui et al., 2002; Teacy et al., 2006; Xiong and Liu, 2004; Jøsang and Ismail, 2002; Buchegger and Le Boudec, 2004; Nielsen et al., 2007). This is the reason of using the principle of *exponential decay*, also known as the *'forgetting'* as introduced by Jøsang and Ismail (2002).

The exponential decay principle is based on the idea that recent interaction outcomes reflect the current behaviour of a principal more significantly than old outcomes. Since the probabilistic trust in a principal corresponds to its current behaviour (as a probability distribution over possible interaction outcomes), the decay principle is proposed to reduce (or 'decay') the effect of old observations on trust evaluation. This is implemented by associating each outcome with a weight, such that old outcomes are given less weight than more recent outcomes. More precisely, given an observation sequence $h = o_1\, o_2\, \ldots\, o_n$, an outcome o_i is given the weight r^{n-i}, where $0 \leq r \leq 1$ is known as the *decay* or *forgetting* factor. In particular, the oldest outcome o_1 is given the weight r^{n-1} while the most recent one o_n is given the largest weight 1. This results in decaying (forgetting) old outcomes while taking recent ones into account.

With the weighting scheme described above the exponential decay has been introduced to the beta reputation model by using the sum of weights associated with successes (respectively failures) in place of the plain count $\#_{\mathsf{s}}(h)$ of successes (respectively $\#_{\mathsf{f}}(h)$ of failures) in a given observation sequence h (see Eq. (3.1)). In the same way, the mechanism of weighting observations by the decay factor is also applicable to the more general Dirichlet model (described above in Section 3.2.2). In addition to their introduction of the Dirichlet model, Josang and Haller, detailed in their paper (Jøsang and

Haller, 2007) the usage of the exponential decay to enhance the model.

Later in Chapter 5 we will provide a detailed analysis for beta trust model with exponential decay scheme in order to evaluate the estimation precision of this model given a system exhibiting dynamic behaviour. In particular we show that the decay principle does not improve beta estimation in all cases and it is highly effective when the system is unlikely to change its behaviour.

Chapter 4

Application of probabilistic trust to anonymity

This chapter addresses the general question of how trust information is used to maximise the security and reliability of interactions between principals. The answer to this question is straightforward in the case of credential based trust where a principal is either trusted or not; a principal interacts with another only if they trust each others. In the case of probabilistic trust, the usage of trust ratings is not of the same simplicity. Here, trust ratings are represented as probabilities associated with trustees. This implies that there is always a risk of interaction with any partner. Entire avoidance of the risk means that a principal rarely interacts with others. A principal needs therefore to trade-off the risk of interacting with others (depending on his trust in them) and the benefit gained from such interactions.

In this chapter, we describe incorporating the notion of probabilistic trust in anonymity protocols which aim at preserving the privacy (anonymity) of network principals. In particular, we focus on a well known anonymity protocol, called the Crowds. This protocol is originally proposed by Reiter and Rubin (1998) to provide anonymous web transactions, that is to hide the identity of transaction initiators from an external attacker who may be monitoring or controlling the target server or another user's computer.

In the Crowds anonymity protocol, a group of principals cooperate to hide the identity of each other when sending a message to a server. Instead of delivering the massage to the server directly, a message takes a random path to the server through the protocol members, such that each node on the message path does not disclose the identity of its preceding node. In this sense, a member is 'honest' if he follows the anonymity protocol by forwarding the message to another member, and also not disclosing the identity of its preceding node in the message path. Otherwise, the member is said to be 'corrupted'. We also refer to a corrupted member as an *attacker*. The corruption of a member can occur when his computer gets infected by a malicious program (e.g. a virus

or a spyware) downloaded from a remote host. In this case the corrupted member is monitored (and possibly controlled) by the remote host. Thus, the event that a message is captured by an infected node implies disclosing the identity of the preceding node (on the message path) to an attacker. A corrupted member switches back to be honest when the malicious program is removed (or suppressed) by the member's security system (e.g. antiviral software).

The existing specification of the CROWDS protocol assumes that a participating member is either always 'honest' or always 'corrupted'. Based on this assumption, the analysis given by Reiter and Rubin (1998) shows that this protocol provides a level of anonymity (privacy) called *probable innocence* if the number of corrupted members is under certain limit. In this chapter we adopt a more realistic assumption that when a member receives a message, he behaves honestly with a certain probability. This probability depends therefore on the robustness of the member's security system, such as the strength of his firewall, the quality of his anti-virus system, and so on. Here the role of probabilistic trust comes to light as it refers to the probability that a principal follows the agreed interaction protocol. In the case of CROWDS, the probabilistic trust in a member, is therefore an estimate for his probability of being honest.

Trust rating (or simply, the trust) in a given member is determined by a trust management module according to the available evidence about the trustee. This evidence can be, for example, information about the anti-malware system employed by the member. Using benchmark data or market reviews (reputation) of such a system, the trust in it, and hence in the employing trustee can be evaluated. In doing so, the trust management module consults certain authorities, e.g. *AV-Comparative* (http://www.av-comparatives.org/en/home), which publish the results of testing different anti-malware systems against samples of malicious codes. Using such results, the likelihoods (trust values) of such systems to prevent attacks are estimated. Additionally, the malicious behaviour of a corrupted member may be also reflected by observables, e.g. delaying or blocking the messages forwarded to him for the purpose of lowering the reliability of the protocol. Such observables, if available to the trust management module, can contribute to the trust evaluation.

Under the assumptions of probabilistic behaviour of individual protocol participants, and the availability of trust information about them, we extend the specification of the CROWDS protocol to utilise the available trust information about each participant. Then, through the analysis of the extended protocol (referred to as *Crowds-trust*) we describe how the parameters of the protocol can be adjusted based on the given trust values to provide a desirable level of anonymity to each participating member. The results of this chapter have been also presented in our recent paper (Sassone et al., 2010).

4.1 Anonymity protocols

The objective of any anonymity protocol is to hide aspects of the communication between a user and a server (receiver). According to Pfitzmann and Waidner (1987), three anonymity properties can be provided by an anonymity protocol: *sender anonymity*, meaning that the identity of a message sender is hidden; *receiver anonymity*, meaning that the identity of a message receiver is hidden, and *unlinkability of sender and receiver*, meaning that the fact that a sender and receiver are communicating with each other is hidden though each of them can be identified to be engaged in some communication.

This chapter specifically considers the protocols providing sender anonymity. The objective of these protocols is to conceal the identities of users interacting with target servers from eavesdroppers who try to identify these users based on observations. An eavesdropper can be the target server receiving a message from the user, or an intermediate node in the path between the user and the server. While the content of a message sent by the user can be protected using encryption techniques, it is not straightforward to hide the identity of the sending user from the server. Namely, If a user forwards its message directly to the server, the network address of the sender is exposed to the server, and hence the user is identified (assuming the owner of the address is known by the server).

The first approach to hide senders' identities from servers was to employ an additional party (a *proxy*), which receives messages from the users and direct them to the destination servers. Examples for system using this approach include some web sites (e.g. www.anonymizer.com), and the Lucent's *Personal Web Assistant* (Gabber et al., 1997). Indeed the major disadvantage of this approach is it suffers from a single point of failure. If the proxy is hijacked by an attacker, the anonymity of all users is broken. The Crowds protocol (Reiter and Rubin, 1998) is then designed so that a user message is directed to the server through a random path of users, rather than a single proxy. This does not only hides the identity of the message originator from the end server, but also from an attacker controlling an intermediate node in the message path.

4.1.1 Analysis framework and notations

The anonymity protocols can be seen as an instance of the the more general information-hiding protocols designed to hide a certain fact (secret) from an observer (attacker) who tries to infer the fact from observations resulting from the protocol runs. An information-hiding protocol aims therefore at weakening the linkage between the fact and its dependent observables. In the case of the sender anonymity, the secret is the identity of a user initiating a transaction, while the related observable depends on the protocol itself and the type of attack. If, for instance, the attacker controls (or monitors) a 'corrupted' node j in the message path to the server, the observable can be the identity of the node which forwards the message to j.

In this chapter we adopt a general analysis framework commonly used in probabilistic approaches to anonymity and information flow (e.g. Halpern and O'Neill, 2005, Chatzikokolakis et al., 2008a, Malacaria and Chen, 2008, and Smith, 2009). In this framework, the identity of a message initiator is called the *anonymous* input to the protocol, and modelled by a random variable A ranging on a set $\mathcal{A}$ of finite values. A protocol run results also in an observable which is modelled by a random variable O ranging on a finite set $\mathcal{O}$ of finite values. A protocol is then represented by the matrix of the conditional probabilities $P(O = o_j \,|\, A = a_i)$ (or simply $P(o_j \,|\, a_i)$). The probability $P(o_j \,|\, a_i)$ is the probability of observing o_j given that the anonymous input is a_i, where $o_j \in \mathcal{O}$ and $a_i \in \mathcal{A}$.

It is assumed that the anonymous 'random' input (the value of A) is generated according to an *a priori* publicly-known probability distribution. An adversary or eavesdropper can see the output (observable) O of a protocol, but not the input. He is then interested in deriving the value of the input from the observed output.

4.1.2 Quantifying anonymity

Based on the above probabilistic framework, an attacker may not have 'certain' knowledge about the value of the protocol's hidden input A (e.g. the sender's identity). However, he could, using an observed output, o_j, to calculate a subjective probability distribution over the set $\mathcal{A}$ of possible values of A. That is to calculate the conditional probabilities $P(a_i \,|\, o_j)$, the probability of a_i being the input value given that the output o_j is observed, for all $a_i \in \mathcal{A}$. This conditional probability distribution has been used in some works (e.g. Reiter and Rubin, 1998, Halpern and O'Neill, 2005) to quantify the anonymity of a given protocol. Regarding a_i as the event that a user i initiates a transaction, the lower is $P(a_i \,|\, o_j)$, the higher level of anonymity is provided to a_i.

A notable trend of research goes also to use information-theoretic concepts to quantify the quality of information hiding and anonymity protocols in the above probabilistic setting (e.g. Shmatikov and Wang, 2006; Chatzikokolakis et al., 2007, 2008a,b; Bhargava and Palamidessi, 2005; Malacaria and Chen, 2008; Hamadou et al., 2010). In these works, information 'leaked' to an observer is quantified as the loss of *uncertainty* about protocol inputs, due to an observation. Here the uncertainty about the hidden input A before and after an observation O is modelled respectively by the entropy $H(A)$ and conditional entropy $H(A \,|\, O)$ of the random variable A. The difference $I(A\,;\, O) = H(A) - H(A \,|\, O)$, known as the *mutual information* between A and O, is then used to quantify the information leaked to the observer due to O. The formal definitions and properties of the entropy, conditional entropy, and mutual information can be found in e.g. (Cover and Thomas, 2006).

As the main concern in this chapter is about incorporating trust in anonymity proto-

cols, we adopt the former (and simpler) approach where the anonymity of a user i is determined by the probability $P(a_i \mid o_j)$, the probability that i is the message initiator, given that an observable o_j is detected by an observer.

4.1.3 The Crowds protocol

Crowds is a protocol proposed by Reiter and Rubin (1998) to allow Internet users performing anonymous web transactions by protecting their identities as originators of messages. The central idea to ensure anonymity is that the originator forwards the message to another, randomly-selected user, which in turn forwards the message to another user, and so on until the message reaches its destination (the end server). This routing process ensures that, when a user is detected sending a message, there is a substantial probability that he is simply forwarding it on behalf of somebody else.

More specifically, a crowd is a *fixed* number of users participating in the protocol. Some members (users) in the crowd may be corrupted (the *attackers*), and they can collaborate in order to discover the originator's identity. The purpose of the protocol is to protect the identity of the message originator from the attackers. When an *originator* –also known as *initiator*– wants to communicate with a server, he creates a random *path* between himself and the server through the crowd by the following process.

- *Initial step:* the initiator selects randomly a member of the crowd (possibly himself) and forwards the request to him. We refer to the latter user as the *forwarder*.
- *Forwarding steps:* a forwarder, upon receiving a request, flips a *biased* coin. With probability $1 - p_f$ he delivers the request to the end server. With probability p_f he selects randomly a new forwarder (possibly himself) and forwards the request to him. The new forwarder repeats the same forwarding process.

The response from the server to the originator follows the same path in the opposite direction. Each user (including corrupted users) is assumed to have only access to messages routed through him, so that he only knows the identities of his immediate predecessor and successor in the path, and the server. Figure (4.1) shows the operation of Crowds, where a message originated by the user 2 follows a random path through the users $1, 3, 3, 8, 9$ to the target server 2. Note that in this run of the protocol, the user 3, randomly, chose himself as the next forwarder.

4.1.4 Probable innocence

Reiter and Rubin (1998)) proposed a hierarchy of anonymity notions in the context of Crowds. These range from '*absolute privacy*,' where the attacker cannot perceive

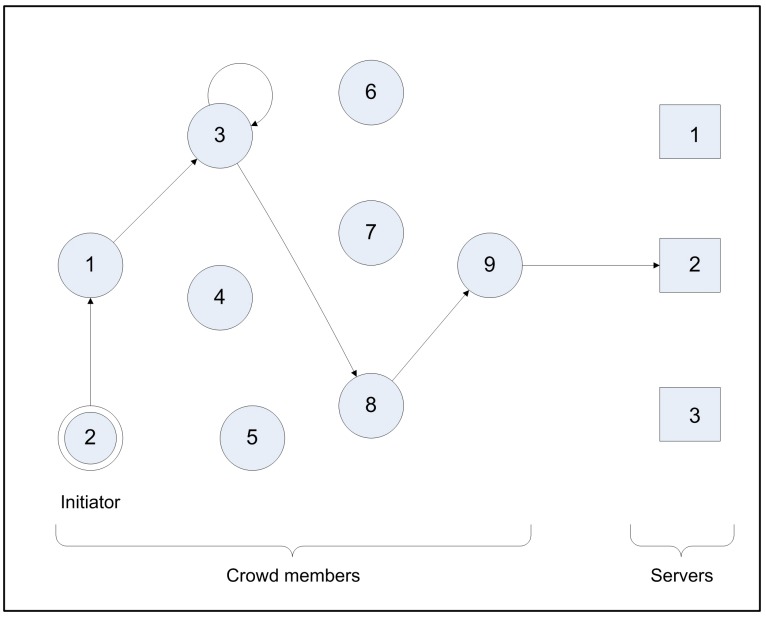


FIGURE 4.1: A message path in the CROWDS protocol

the presence of communication, to '*provably exposed*,' where the attacker can prove the sender and receiver relationship. Clearly, as most protocols used in practice, CROWDS cannot ensure absolute privacy in presence of attackers or corrupted users, but can only provide weaker notions of anonymity. In particular, in (Reiter and Rubin, 1998) the authors propose an anonymity notion called *probable innocence* and prove that, under some conditions on the protocol parameters, CROWDS ensures the probable innocence property to the originator. Informally, they define it as follows:

> *A sender is probably innocent if, from the attacker's point of view, the sender appears no more likely to be the originator than to not be the originator.* (4.1)

In other words, the attacker may have reason to suspect the sender being more likely than any other potential sender to be the originator, but it still appears at least as likely that he is not.

Let n be the number of users participating in the protocol and let c and m be the number of the corrupted and honest users, respectively, with $n = m + c$. Since anonymity makes only sense for honest users, we define the set of anonymous events as $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$, where a_i indicates that user i is the initiator of the message.

As it is usually the case in the analysis of CROWDS, we assume that when a message is received by a corrupted member, he forwards it immediately to the end server, since forwarding it any further cannot help him learn anything more about the identity of the

originator. Thus in any given path, there is at most one detected user: the first honest member to forward the message to a corrupted member. Therefore we define the set of observable events as $\mathcal{O} = \{o_1, o_2, \ldots, o_m\}$, where o_j indicates that user j forwarded a message to a corrupted user. In this case it is also said that user j *is detected* by the attacker.

In (Reiter and Rubin, 1998), the authors formalize their notion of probable innocence via the conditional probability $P(I \mid H)$ that the initiator is detected given that any user is detected at all. Here H denotes the event that there is an attacker in the path, and I is the event that it is precisely the initiator (as a node in the path) who forwarded the message to the attacker.

Precisely, probable innocence holds if $P(I \mid H) \leq \frac{1}{2}$. In the setting, described by Section 4.1.1, this probability can be written as $\sum_i P(o_i \mid a_i,\, H)\, P(a_i \mid H)$. Since the Crowds protocol is *symmetric*, i.e. each initiator has the same probability of being detected, the probability $P(o_i \mid a_i,\, H)$ is the same for all users i. Therefore, the notion of probable innocence used by Reiter and Rubin (1998) (i.e. $P(I \mid H) \leq \frac{1}{2}$) translates in our setting as:

$$\forall i.\ P(o_i \mid a_i, H) \leq 1/2 \tag{4.2}$$

It is shown by Reiter and Rubin (1998) that, in Crowds, the following holds:

$$P(o_j \mid a_i, H) \;=\; \begin{cases} 1 - \frac{m-1}{n} p_f & j = i \\ \frac{1}{n}\, p_f & j \neq i \end{cases} \tag{4.3}$$

Therefore, probable innocence (4.2) holds if and only if

$$m \geq \frac{c+1}{p_f - \frac{1}{2}} p_f$$

As previously noticed in several papers (e.g. Chatzikokolakis and Palamidessi, 2006), there is a mismatch between the idea of probable innocence expressed informally in (4.1) and the property actually proved by Reiter and Rubin, cf. (4.2). The former, indeed, seems to correspond to the following interpretation given by Halpern and O'Neill (2005):

$$\forall i.\ P(a_i \mid o_i) \leq 1/2 \tag{4.4}$$

The properties (4.2) and (4.4) however coincide under the standard assumption in Crowds that the *a priori* distribution is uniform, i.e. that each honest user has equal probability of being the initiator.

Finally it is remarkable that the concept of probable innocence was recently generalised by Hamadou et al. (2010). Instead of just comparing the probability of being innocent with the probability of being guilty, the paper considers the degree of the probability of

being innocent. Formally, given a real number $\alpha \in [0, 1]$, a protocol satisfies α-probable innocence if and only if

$$\forall i.\ P(a_i \mid o_i) \leq \alpha \tag{4.5}$$

Clearly α-probable innocence coincides with the standard probable innocence for $\alpha = 1/2$.

4.2 Using trust information

The existing analysis of Crowds by Reiter and Rubin (1998) as well as other works on anonymity assumes that participants in the protocol are divided into two classes: *honest* members who always behave correctly and the *bad guys* or *attackers* who try to break the protocol. Obviously, the clear separation between honest ('trustworthy') members and attackers makes the analysis easier. However, this is clearly not a realistic assumption for open and dynamic systems in the era of ubiquitous computing. In such systems, a principal is not always guaranteed to behave honestly. Precisely, there is always a probability that a principal gets corrupt, and violates the agreed protocol. This calls for using the probabilistic notion of trust to reason about users behaviours, where the probabilistic trust in a user is an estimated probability that he behaves correctly (honestly).

In this section, the Crowds protocol is reformulated under the novel scenario where each user behaves honestly with a certain probability, which is given by a trust value associated with this user. Trust values for individual users are assumed to be computed by a separate trust management module. We then study the effect of such probabilistic behaviours of users on the anonymity properties of the protocol.

4.2.1 Crowds protocol extended

We now extend the Crowds protocol to take into account the trust levels of its participating members. We associate a trust level $t_{ij} \in [0, 1]$ to each pair of users i and j to indicate the trust of user i in user j. Here t_{ij} denotes the probability that when the principal i chooses principal j as a forwarder, j behaves honestly and protects i's identity. Accordingly each user i defines his *policy of forwarding* to other members (including himself) based on his trust in them. A policy of forwarding for a user i is a probability distribution $\{q_{i1}, q_{i2}, \cdots, q_{in}\}$, such that for all i, $\sum_{j=1}^{n} q_{ij} = 1$. Here q_{ij} denotes the probability that j is chosen as a forwarder by i (given that i has decided to forward the message).

Defining trust at an individual level as described above is certainly desirable in general. However for some applications – specifically the Crowds protocol – it is more reasonable

to consider a global notion of trust where trust in a user is common for everybody. In other words $t_{ij} = t_{kj}$ for all i and k. Indeed, in the case of the CROWDS protocol, we want a trust in a user to reflect her robustness to being *corrupted* (a.k.a. *infected*). Allowing each member to adopt his own level of trust would make the value of trust subjective and could hardly reflect the user's actual robustness against corruption.

It is therefore assumed that a trust in a user is global. Its value could be established cooperatively by all members of the crowd or by a local authority (the *blinder* in case of the CROWDS protocol) based on evidence provided by the user. Accordingly, in the *rest of this chapter*, we will simply write t_i to denote the trust level of user i. Similarly we require that the policy of forwarding to be common to all members of the 'crowd'. In other words, every participant treats any given user in the same way, as all of them have the same trust in him. We therefore write $\{q_1, q_2, \cdots, q_n\}$ to represent the common forwarding policy.

Under these assumptions we now extend the protocol. When an initiator wants to communicate with a server, he creates a random *path* between himself and the server through the crowd by the following process.

- *Initial step:* with probability q_j the initiator selects a member j of the crowd (possibly himself) according to the policy of forwarding $\{q_1, q_2, \cdots, q_n\}$ and forwards the request to him. We refer to the latter user as the *forwarder*.
- *Forwarding steps:* a forwarder, upon receiving a request, flips a *biased* coin. With probability $1 - p_f$ he delivers the request to the end server. With probability $p_f \times q_k$ he selects a new forwarder k (possibly himself) and forwards the request to him. The new forwarder repeats the same forwarding process.

4.2.2 Probable innocence revisited

In order to study the anonymity provided by the extended protocol, we first establish our hypotheses of analysis. As in Section 4.1.4, we assume that corrupted members will always deliver a request to forward immediately to the end server, since forwarding it any further cannot help the attacker learn anything more about the identity of the originator. Consequently when an infected user initiates a transaction, his message is directly delivered to the end server.[1]

It is also assumed that the anonymous paths from users to servers are one-way. That is, we do not consider the cases where there might be responses from servers to users which would normally follow the same paths in reverse direction. Under these assumptions there is always at most one corrupted member appearing on a path and corrupted members always occupy the last position in the paths. Had we considered the reverse

[1] His anonymity is broken at the start, so there is no need to continue the anonymity protocol.

direction too, there would be members appearing in a path while being not infected in the forward direction, become corrupted by the time they receive the response from the server. Hence they would report their predecessor as detected members, because these are closer in the path to the initiator than the member detected in the forward direction. We leave this significant case for future work.

Finally, while the analysis of the original Crowds considers the anonymous actions and observables (detections) of only honest users, each user i, in our current analysis, has probability t_i of being honest when he initiates a request. Thus we extend the set of anonymous events a_i and observable events o_i to the whole set of participating members.

Under these assumptions we study the privacy level ensured to each member participating in the protocol. This privacy level is indicated by the probability $P(a_i \mid o_i)$, which can be written as

$$P(a_i \mid o_i) = \frac{P(a_i, o_i)}{P(o_i)} \tag{4.6}$$

We first evaluate the denominator in the above expression. Let H_k be the event that the first corrupted node in the message path to the server occupies the kth position, where $k \geq 0$. Note that H_0 means that the initiator himself is corrupted.

$$P(o_i, H_k) = \begin{cases} \frac{1}{n}(1 - t_i) & k = 0 \\ \frac{1}{n} t_i \sum_{j=1}^{n} q_j (1 - t_j) & k = 1 \\ \sum_{j=1}^{n} \frac{1}{n} t_j \left(\sum_{j=1}^{n} q_j t_j \right)^{k-2} \cdot & \\ \qquad q_i t_i \left(\sum_{j=1}^{n} q_j (1 - t_j) \right) \cdot p_f^{k-1} & k \geq 2 \end{cases} \tag{4.7}$$

The above equation for the case $k \geq 2$ is implied by the fact that the message is initiated by any honest participant, forwarded to $k - 2$ honest principals before it is passed to the detected principal i, and finally to a corrupted one. For convenience, we will write $T = \sum_{j=1}^{n} q_j t_j$ and $S = \sum_{j=1}^{n} t_j$. Since the joint events $\{o_i, H_k\}$, for $k \geq 0$ are mutually exclusive, we evaluate $P(o_i)$ as follows.

$$\begin{aligned} P(o_i) &= \sum_{k=0}^{\infty} P(o_i, H_k) \\ &= \frac{1}{n}(1 - t_i) + \frac{1}{n} t_i (1 - T) \\ &\quad + \sum_{k=2}^{\infty} \frac{1}{n} S T^{k-2} \cdot q_i t_i \, (1 - T) \cdot p_f^{k-1} \\ &= \frac{1}{n} \left(1 - t_i T + S p_f q_i t_i \left(\frac{1 - T}{1 - p_f T} \right) \right) \end{aligned} \tag{4.8}$$

From Equation (4.8), it is worth noticing that $P(o_i) = 0$ only if $T = 1$ and $t_i = 1$. Observe that $T = 1$ means that $t_j = 1$ for all participants j where $q_j \neq 0$, i.e., all

forwarders are always honest. In this case i is never detected by any forwarder. If moreover $t_i = 1$, the principal i is never detected by himself. Thus in the case where $T = 1$ and $t_i = 1$ the principal i is never detected by any corrupted node.

Now we turn to evaluating the probability $P(a_i, o_i)$ appearing as the numerator in Equation (4.6). To such purpose, we first formulate the probability $P(a_i, H_k, o_i)$, i.e., the probability that i is the initiator and is also detected by a corrupted node at position k in the message path.

$$P(a_i, H_k, o_i) = \begin{cases} \frac{1}{n}(1 - t_i) & k = 0 \\ \frac{1}{n} t_i \sum_{j=1}^{n} q_j (1 - t_j) & k = 1 \\ \frac{1}{n} t_i \left(\sum_{j=1}^{n} q_j t_j \right)^{k-2} \cdot & \\ \qquad q_i t_i \left(\sum_{j=1}^{n} q_j (1 - t_j) \right) \cdot p_f^{k-1} & k \geq 2 \end{cases} \tag{4.9}$$

Similar to the argument of Equation (4.7), the formula in the case $k \geq 2$ is implied by the fact that the message is initiated by the principal i, forwarded to $k-2$ honest principals before it is passed back to i, and finally to a corrupted principal. Since the joint events $\{a_i, H_k, o_i\}$, for $k \geq 0$ are mutually exclusive, we evaluate $P(a_i, o_i)$ as follows.

$$\begin{aligned} P(a_i, o_i) &= \sum_{k=0}^{\infty} P(a_i, H_k, o_i) \\ &= \frac{1}{n}(1 - t_i) + \frac{1}{n} t_i (1 - T) \\ &\quad + \sum_{k=2}^{\infty} \frac{1}{n} t_i T^{k-2} \cdot q_i t_i \, (1 - T) \cdot p_f^{k-1} \\ &= \frac{1}{n} \left(1 - t_i T + p_f q_i t_i^2 \left(\frac{1 - T}{1 - p_f T} \right) \right) \end{aligned} \tag{4.10}$$

Assuming $P(o_i) \neq 0$, we substitute Equations (4.8) and (4.10) in Equation (4.6), and we therefore get,

$$P\left(a_i \mid o_i\right) = \frac{1 - t_i T + p_f q_i t_i^2 \left(\frac{1-T}{1-p_f T} \right)}{1 - t_i T + S p_f q_i t_i \left(\frac{1-T}{1-p_f T} \right)} \tag{4.11}$$

From Equation (4.11), we observe that for a detectable principal i (i.e., $P(o_i) \neq 0$), it holds that $P(a_i \mid o_i) > 0$. That is, there is always a non zero probability that i is the initiator if he is detected. This confirms the fact that Crowds never achieves the highest degree of anonymity known as *absolute privacy* described by Reiter and Rubin (1998).

4.2.3 Provably exposed principals

It would also be interesting to investigate the conditions under which the protocol can only ensure the degree of anonymity known as *provably exposed* to a given principal i. This degree of anonymity, defined by Reiter and Rubin (1998), represents the lowest level of anonymity where an attacker can prove the identity of the message initiator. This happens when i is the only possible initiator, given that i is detected, i.e. $P(a_i \mid o_i) = 1$. These conditions are precisely stated by the following proposition.

Proposition 4.1 (Provably exposed)**.** *For all user i such that $P(o_i) \neq 0$, we have $P(a_i \mid o_i) = 1$ if and only if one of the following conditions holds:*

1. $p_f = 0$
2. $t_i = 0$
3. $q_i = 0$
4. $T = 1$
5. $S = t_i$

Proof. Solving the following equation $P(a_i \mid o_i) = 1$ using the formula given by Equation (4.11) yields only the above conditions. □

The following paragraphs discuss the meaning of these results. First note that $p_f = 0$ means that the initiator picks his first forwarder (if he is not already corrupted) according to the forwarding policy $\{q_1, \cdots, q_n\}$ (initial step of the protocol) who has to directly deliver the message to the end server regardless of his being corrupted or not. Thus, in this case a path is always at most of length 2 (not counting the end server). Hence, i can only be detected at position 0 (by himself if he is initially corrupted) or at position 1 by his forwarder when the latter is corrupted. Therefore in both cases i is the only possible initiator. That is if a principal i is detected then he must be the initiator.

In the case where $t_i = 0$, i is always corrupted and therefore when he initiates a message, he will detect himself and deliver the message directly to the end server (by assumption). Hence nobody except himself will detect him. Thus i is detected if and only if he is the initiator.

Consider the case where $q_i = 0$, that is i is never chosen as a forwarder. In this case, similar to the case $p_f = 0$, i can be detected only at position 0 (by himself if corrupted), or at position 1 if he forwards his message to a corrupted user. Indeed, when a corrupted node detects i, he is sure that i is the initiator because it can never be a forwarder.

The case $T = 1$ happens if and only if $t_j = 1$ for all $q_j \neq 0$, that is only 'permanently' honest members can be chosen as forwarders. In this case also, i is detected only if he initiates a message and he is corrupted at the same time. Hence he detects himself. Thus detecting i implies that i is the initiator.

Finally, suppose that $S = t_i$. Here $t_j = 0$ for all $j \neq i$, that is all participants other than i are permanently corrupted. In this case if i is detected then he is the only possible initiator because otherwise the initiator would be detected by himself at level 0 and hence i is not detected. Therefore, in this case, if i is detected, he must be the initiator.

It is worth noticing that the original Crowds protocol is the protocol obtained by assuming that each principal i is either always honest or always corrupted, i.e., $t_i \in \{0, 1\}$, and by choosing a uniform forwarding policy, that is for all j,

$$q_j = \frac{1}{n} \ .$$

Thus when the number of corrupted principals is c, we have

$$T = \sum_{j=1}^{n} q_j t_j = \frac{n-c}{n} \ ,$$

and

$$S = \sum_{j=1}^{n} t_j = n - c \ .$$

By substituting the values of q_j, T and S in Equation (4.11) for a honest initiator i ($t_i = 1$), we get

$$P\left(a_i \mid o_i\right) = 1 - p_f \left(\frac{n-c-1}{n}\right) \ .$$

The above expression is the same expression derived by Reiter and Rubin (1998) for the original Crowds and given by Equation (4.3).

4.3 Achieving probable innocence

For any fixed number of principals n, the extended protocol described in the previous section has three main parameters: the forwarding probability p_f, members' trust values $\{t_1, \cdots, t_n\}$, and the forwarding policy $\{q_1, \cdots, q_n\}$. This section studies how each of them affects the anonymity of the participating members. We begin by the probability of forwarding p_f.

4.3.1 Probability of forwarding

The following result states that for fixed trust values $\{t_1, \cdots, t_n\}$ and forwarding policy $\{q_1, \cdots, q_n\}$, the probability $P(a_i \mid o_i)$ for any participant i is a monotonically decreasing function with respect to the forwarding probability p_f.

Theorem 4.2 (Monotonicity)**.**

$$\forall i.\ \frac{\partial P(a_i \mid o_i)}{\partial p_f} \leq 0.$$

Proof. By differentiating $P(a_i \mid o_i)$, given by Equation (4.11), with respect to p_f we have

$$\frac{\partial P(a_i \mid o_i)}{\partial p_f} = \frac{t_i\, q_i\, (1-T)\, (1-t_i T)\, (t_i - S)}{((1-p_f T)(1-t_i T) + p_f S q_i t_i (1-T))^2} \tag{4.12}$$

Given that $0 \leq t_j \leq 1$ for any principal j, and $T = \sum_{j=1}^{n} q_j t_j$, we have $0 \leq T \leq 1$ and $0 \leq t_i T \leq 1$. We have also $t_i \leq S$, because $S = \sum_{j=1}^{n} t_j$. Thus,

$$\frac{\partial P(a_i \mid o_i)}{\partial p_f} \leq 0$$

That is $P(a_i \mid o_i)$ is either fixed or decreasing with respect to p_f. □

From Equation (4.12) above, $P(a_i \mid o_i)$ is fixed irrespective of p_f if and only if i is always corrupted ($t_i = 0$), i is never used as a forwarder ($q_i = 0$), all forwarders are honest ($T = 1$), or all participants other than i are corrupted ($S = t_i$). It has been shown by Proposition 4.1 in the previous section that $P(a_i \mid o_i) = 1$ in these cases.

Theorem 4.2 justifies using a high value of p_f as it decreases the probability of identifying the initiator and therefore enhance his privacy. However, large p_f implies that the message path to the server is longer and therefore the performance of the protocol is degraded. Thus a trade-off is required for choosing the forwarding probability p_f.

Corollary 4.3 (Anonymity range)**.**

$$\forall i.\ 1 \geq P(a_i \mid o_i) \geq 1 - \frac{q_i t_i \sum_{j \neq i}^{n} t_j}{1 - t_i \sum_{j \neq i}^{n} q_j t_j + q_i t_i \sum_{j \neq i}^{n} t_j}$$

Proof. By Theorem 4.2, and taking into account that $0 \leq p_f \leq 1$, the above range for $P(a_i \mid o_i)$ is obtained by substituting $p_f = 0$ and $p_f = 1$ in Equation (4.11). □

The above corollary describes the range of the probability that a principal i is the initiator given that i is detected. Note that with $p_f = 0$, any initiator i is *provably exposed* to the attacker (i.e. $P(a_i \mid o_i) = 1$), by Proposition 4.1. Taking $p_f = 1$, on the other end, minimizes $P(a_i \mid o_i)$, but in this case the message never reaches the server.

4.3.2 Trust values

We now turn our focus to the trust values. Observe that the anonymity of a member i, indicated by $P(a_i \mid o_i)$, is affected by the trust values t_j of all participating members. Therefore, the above lower bound can be used as a criterion to decide whether a new member i is accepted to join the network or not based on his trust t_i. For instance, such a criterion can be chosen to achieve the *α-probable innocence* according to the following theorem.

Theorem 4.4 (α-probable innocence)**.** *Let $\alpha \in [0,1]$ be a positive value. If*

$$\forall i.\ \frac{q_i t_i \sum_{j\neq i}^{n} t_j}{1 - t_i \sum_{j\neq i}^{n} q_j t_j + q_i t_i \sum_{j\neq i}^{n} t_j} \geq 1 - \alpha$$

then the extended protocol ensures α-probable innocence to all its participating members.

Proof. Results from Corollary 4.3 and Definition (4.5). □

4.3.3 Forwarding policy.

We now propose a strategy for choosing a forwarding policy $\{q_1, \cdots, q_n\}$ based on the trust information $\{t_1, \cdots, t_n\}$ in order to achieve α-probable innocence for a given degree of privacy α. The main idea is that the forwarding probabilities q_j are adjusted depending on the given trust information t_j.

Choosing the forwarding policy q_i for a given user i can then be done by maintaining the lower bounds of $P(a_i \mid o_i)$ below a chosen threshold α, i.e. by achieving α-probable innocence. By Theorem 4.4 the plausible values of q_i are obtained by solving the following system of linear inequalities.

$$\begin{aligned} 1 - \alpha \ &\leq\ \frac{q_i t_i \sum_{j\neq i}^{n} t_j}{1 - t_i \sum_{j\neq i}^{n} q_j t_j + q_i t_i \sum_{j\neq i}^{n} t_j} \qquad 1 \leq i \leq n \\ 1 \ &=\ \sum_{i=1}^{n} q_i \end{aligned}$$

Example 4.1. *Consider an instance of the Crowds-Trust protocol where three principals are involved. Let the trust values in these principals be:*

$$t_1 = 0.70, \qquad t_2 = 0.97, \qquad t_3 = 0.99$$

Solving the above problem for $\alpha = \frac{1}{2}$ *yields the two solutions:*

$$0.2479 \leq q_2 \leq 0.2620$$
$$1.1411 - 3.4138\, q_2 \leq q_3 \leq 0.5479 - 1.0206\, q_2$$
$$q1 = 1 - q_2 - q_3$$

and

$$0.2620 \leq q_2 \leq 0.3074$$
$$0.3197 - 0.2784\, q_2 \leq q_3 \leq 0.5479 - 1.0206\, q_2$$
$$q_1 = 1 - q_2 - q_3\ .$$

Thus the following forwarding distribution satisfies the $\frac{1}{2}$*-probable innocence:*

$$q_1 = 0.4575, \qquad q_2 = 0.2620, \qquad q_3 = 0.2805\ .$$

However, if the uniform distribution is used (as in the original Crowds *protocol), i.e.,* $q_1 = q_2 = q_3 = \frac{1}{3}$*, probable innocence is not achievable because the minimum value of* $P(a_1 \mid o_1)$ *according to Corollary 4.3 is* 0.543*, which is greater than* $\frac{1}{2}$*. Note that the above set of constraints are not always solvable, in which case the required level of anonymity cannot be provided to all members.*

In the above example, observe that the forwarding policy increases the frequency at which the less reliable user 1 will be involved in a message path. This makes it hard for an attacker detecting 1 to identify him as the initiator, because the attacker knows that 1 frequently forwards messages for others, and hence it is substantially probable that the initiator is another user. This compensates the relatively high probability (0.3) of 1 being detected by himself. However, the higher security for 1 is of course achieved at the price of a lower overall security for other, more reliable, users (2 and 3). Namely, forwarding their messages to the least reliable member (1) makes them more probable to be detected by 1. Thus the above policy which grants a specific level of anonymity to all users can be seen as a 'social' approach to crowds membership. The flexibility of the extended protocol means that the forwarding policy can be chosen to provide a lower degree of anonymity to a subset of the members to guarantee probable innocence to a larger crowd ('*social strategy*'), or to reject principals having the low trust values who, therefore, exhibit a greater threat to others ('*rational strategy*').

4.4 Discussion

In this chapter we presented an application of the probabilistic trust to satisfy an important security property, that is the anonymity of network users. In particular, we focused

on the CROWDS anonymity protocol and asked the question of how its existing analyses are affected by postulating the presence of principals with probabilistic behaviours. This amounts to providing each member i of the crowd with a probabilistic trust value t_i denoting the probability of i being honest (conforming with the protocol). The trust value t_i indicates also the robustness of i against corruption. As the focus in this chapter is to illustrate the usage of trust values rather than inferring them, it is assumed that these values are already given by a separate trust management module.

For allowing using trust values, the CROWDS is extended to associate each user i a preference level of forwarding q_i denoting the probability of choosing him as the next forwarder in the routing process. This allows for specifying the probabilities q_i, as the interaction policy, according to the participants trust values.

Given the probability of forwarding p_f, a level of anonymity α, and the trust levels $t_1, t_2, \cdots, t_n$ of the crowd's members, we have identified the conditions on the probability of choosing a forwarder which are necessary to achieve α-probable innocence. Thus, in the presence of untrusted members, the protocol users can exploit these results to derive an interaction policy $q_1, q_2, \cdots, q_n$, if there exists any, that ensures them a level of satisfactory anonymity.

Chapter 5

Estimation error of Beta trust model with a decay scheme

As described in Chapter 1, the main objective of any probabilistic trust model is to estimate the probability distribution over potential outcomes of the next interaction with the principal under consideration (trustee). We refer to this probability distribution as the *predictive probability distribution*. In the basic Beta trust model (Section 3.2.1), the behaviour of a trustee is represented by a fixed probability distribution Θ over potential interaction outcomes. While such a representation of a trustee's behaviour simplifies the inference of trust values, this representation is not always realistic. Namely, the probability that a trustee interacts positively with its partners may change over time depending on the trustee's internal state(s); that is, a principal may exhibit dynamic behaviour when it interacts with others.

For this reason, the exponential decay (described in Section 3.2.5) has been proposed by Jøsang and Ismail (2002) to cope with principals dynamic behaviours. In this chapter we aim to address the issue of whether and when exponential decay may be an optimal technique for reasoning about principals dynamic behaviours. For this purpose, we model a principal's dynamic behaviour by a Hidden Markov Model (HMM) (described by Section 2.5), where the probability distribution Θ over observables is allowed to change according to state transitions rather then being fixed. Using this representation, we study the estimation error induced by using the Beta model enhanced with exponential decay to estimate the predictive probability distribution.

Under certain conditions on principals' behaviours, namely that their probabilistic state transition matrices are *ergodic*, we derive an analytic formula for the estimation error, which provides a first formal tool to assess precision and usefulness of the decay technique. Also, we illustrate the obtained results by deploying it in an experimental setting to show how the system *stability*, as a measure for state transition frequency, has a dramatic impact on the precision of the Beta trust model enhanced by exponential

decay. The analysis and results obtained in this chapter are also presented in our paper ElSalamouny et al. (2009).

5.1 The exponential decay principle

One purpose of the *exponential decay* principle is to improve the responsiveness of the Beta model to principals exhibiting dynamic behaviours. The idea is to scale by a constant $0 < r < 1$ the information about past behaviour, viz., $\#_o(h) \mapsto r \cdot \#_o(h)$, each time a new observation is made. This yields an exponential decay of the weight of past observations, as in fact the contribution of an event n steps in the past will be scaled down by a factor r^n. Qualitatively, this means that picking a reasonably small r will make the model respond quickly to behavioural changes. Suppose for instance that a sequence of five positive and no negative events has occurred. The unmodified Beta model would yield a Beta pdf with parameters $\alpha = 6$ and $\beta = 1$, predicting the next event to be positive with probability higher than 0.85. In contrast, choosing $r = 0.5$, the Beta model with exponential decay would set $\alpha = 1+31/16$ and $\beta = 1$. This assigns probability 0.75 to the event that the next interaction is positive, as a reflection of the fact that some of the weight of early positive events has significantly decayed. Suppose however that a single negative event occurs next. Then, in the unmodified Beta model the parameters are updated to $\alpha = 6$ and $\beta = 2$, which still assign a probability 0.75 to 'positive' events, reflecting the relative unresponsiveness of the model to change. On the contrary, the model with decay assigns 63/32 to α and 2 to β, which yields a probability just above 0.5 that the next event is *again* negative. So despite having observed five positive events and a negative one, the model with decay yields an approximately uniform distribution, i.e., it considers positive and negative almost equally likely in the next interaction.

Of course, this may or may not be appropriate depending on the application and the hypotheses made on principals behaviours. If on the one hand the specifics of the application are such to suggest that principals do indeed behave according to a single, immutable probability distribution Θ, then discounting the past is clearly not the right thing to do, because all past outcomes are sampled from the same distribution Θ, and therefore discounting past outcomes imposes inefficiency in learning Θ. If otherwise one assumes that principals may behave according to different Θs as they switch their internal state, then exponential decay for a suitable r may make prediction more accurate, because old outcomes don't reflect the current distribution Θ as significantly as the more recent ones. The assumption in this chapter is precisely the latter, and the main objective is to analyse the properties and qualities of the Beta model with exponential decay in dynamic applications, where the dynamic behaviour of a principal is modelled by a hidden Markov model.

5.2 Modelling the real system

As mentioned earlier, the objective of any probabilistic trust model is to 'estimate' the predictive probability distribution, that is the probability distribution over the outcomes of next interaction with the trustee. In order to find an expression for the estimation error, we need to model the outcome generating system by a suitable probabilistic model which we call the *real model*. In this thesis we are interested in studying systems which exhibit a dynamic behaviour, that is changing their behaviour over time. We mathematically model the behaviour of the system at any time by a particular probability distribution over possible outcomes. A system with a dynamic behaviour can therefore be modelled by a multiple state transition system where each state exhibits a particular behaviour (probability distribution). This naturally leads to choosing a generic Hidden Markov Model (HMM) λ as the real model.

Following the results from the theory of Markov chains recalled in Section 2.4, we shall work under the hypothesis that λ is *ergodic*. This corresponds to demanding that all the states of λ remain 'live' (i.e., probabilistically possible) at all times. It then follows by general reasons that λ admits a *stationary probability distribution* over its states S_λ (cf. Theorem 2.5); we denote it by the row vector

$$\boldsymbol{\pi}_\lambda = \begin{bmatrix} \pi_1 & \pi_2 & \dots & \pi_n \end{bmatrix},$$

where π_q denotes the stationary probability of the state q. If $\boldsymbol{A}_\lambda$ is the stochastic state transition matrix representing the Markov chain underlying λ, vector $\boldsymbol{\pi}_\lambda$ satisfies the stationary equation

$$\boldsymbol{\pi}_\lambda = \boldsymbol{\pi}_\lambda \boldsymbol{A}_\lambda. \tag{5.1}$$

As we are only interested λ's *steady-state* behaviour, and as the state distribution of the process is guaranteed to converge to $\boldsymbol{\pi}_\lambda$ after a transient period (cf. Theorem 2.7), without loss of generality in the following we shall assume that $\boldsymbol{\pi}_\lambda$ is indeed λ's initial distribution. Observe too that as λ is finite and irreducible, all components of $\boldsymbol{\pi}_\lambda$ are *strictly* positive and can be computed easily from matrix $\boldsymbol{A}_\lambda$.

For simplicity, we maintain here the restriction to binary outcomes (`s` or `f`), yet our derivation of the estimation error can be generalised to multiple outcomes cases (i.e., replacing beta with Dirichlet pdfs, cf. Jøsang and Haller, 2007; Nielsen et al., 2007).

5.3 Beta model with a decay factor

Consider observation sequences $h_\ell = o_0\, o_1 \cdots o_{\ell-1}$ of arbitrary length ℓ, where o_0 and $o_{\ell-1}$ are respectively the least and the most recent observed outcomes. Then, for r a decay factor ($0 < r < 1$), the beta estimate for the probability distribution on the next

outcomes $\{\mathsf{s}, \mathsf{f}\}$ is given by $(\mathcal{B}_r(\mathsf{s} \mid h_\ell), \mathcal{B}_r(\mathsf{f} \mid h_\ell))$, where

$$\begin{aligned} \mathcal{B}_r(\mathsf{s} \mid h_\ell) &= \frac{m_r(h_\ell) + 1}{m_r(h_\ell) + n_r(h_\ell) + 2} \\ \mathcal{B}_r(\mathsf{f} \mid h_\ell) &= \frac{n_r(h_\ell) + 1}{m_r(h_\ell) + n_r(h_\ell) + 2} \end{aligned} \tag{5.2}$$

and

$$m_r(h_\ell) = \sum_{i=0}^{\ell-1} r^i \delta_{\ell-i-1}(\mathsf{s}) \qquad n_r(h_\ell) = \sum_{i=0}^{\ell-1} r^i \delta_{\ell-i-1}(\mathsf{f}) \tag{5.3}$$

for

$$\delta_i(X) = \begin{cases} 1 & \text{if } o_i = X; \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

Under these conditions, obviously from Equations (5.3) and (5.4), the sum $m_r(h_\ell) + n_r(h_\ell)$ forms a geometric series, and therefore

$$m_r(h_\ell) + n_r(h_\ell) = \frac{1 - r^\ell}{1 - r} \ . \tag{5.5}$$

5.4 The error function

We call the real probability that the next outcome will be s, the *real* predictive probability, and denote it by σ. In contrast, we call the estimated probability that the next outcome will be s the *estimated* predictive probability. The deviation of the estimated predictive probability, given by Equation (5.2) from the real predictive probability σ is expressed by the *Beta estimation error*, which we define by the following equation.

$$D^2(\sigma \mid\mid \mathcal{B}_r(\mathsf{s} \mid h_\ell)) = (\mathcal{B}_r(\mathsf{s} \mid h_\ell) - \sigma)^2 \tag{5.6}$$

That is the Beta estimation error is simply the *squared* difference (which we also refer to as the *quadratic distance*) between the real and predictive probabilities, and hence is denoted above by $D^2(. \mid\mid .)$. Although, in this chapter, we are interested in binary observation case (s or f), the quadratic distance measure can be in general extended to the cases of multiple observations by summing the squared differences between the corresponding probabilities of individual observations. The quadratic distance measure shares some properties with other measures of the statistical distance between probability distributions. Specifically, the measure is 0 if and only if the real and estimated predictive probabilities are equal. Additionally, the measure is symmetric, i.e. $D^2(\mu \mid\mid \nu) = D^2(\nu \mid\mid \mu)$. Choosing the quadratic distance measure for formulating the Beta estimation error is motivated by its advantage that its expected value, in the case of Beta trust estimation, can be expressed analytically as will be shown in the next section.

Observe that whilst the real predictive probability σ depends on λ, the chosen representation of principal's behaviour, and its current state, the estimated predictive probability $\mathcal{B}_r(\mathtt{s} \mid h_\ell)$ depends on the interaction history h_ℓ and the fixed decay parameter r. Thus the Beta estimation error given by Equation (5.6) is basically a random variable. As we aim to assessing the average 'goodness' of the Beta trust model at estimating the predictive probability, we evaluate the *expected Beta estimation error* as the expected value of the Beta estimation error. Note that the expected Beta estimation error depends therefore only on the assumptions about the real model, the decay factor as the parameter for the Beta trust model, and a specified length ℓ of sequences, upon which estimation is performed.

5.4.1 Analysis of the expected Beta estimation error

In the following an expression is derived for the expected Beta estimation error parametric in ℓ as a step towards computing its limit for $\ell \to \infty$, and thus obtain the required formula for the *asymptotic* estimation error. Here we start by expressing the expected beta estimation error as a function of the behaviour model λ and the decay r. Formally,

$$\begin{aligned} \mathit{Error}_\ell(\lambda, r) &= \mathbf{E}\left[D^2(\sigma \mid\mid \mathcal{B}_r(\mathtt{s} \mid h_\ell))\right] \\ &= \mathbf{E}\left[(\mathcal{B}_r(\mathtt{s} \mid h_\ell) - \sigma)^2\right]. \end{aligned} \tag{5.7}$$

As shown earlier in Section 5.4, the real predictive probability σ depends on the current state of the real model λ, whereas the estimated predictive probability $\mathcal{B}_r(\mathtt{s} \mid h_\ell)$ depends on the observation sequence h_ℓ. Thus the expectation in Eq. (5.7) is taken over the current state of λ and the sequence h_ℓ.

Using the definition in (5.2) for $\mathcal{B}_r(\mathtt{s} \mid h_\ell)$, and writing $a = m_r(h_\ell) + n_r(h_\ell) + 2$ for brevity, the error function is rewritten as:

$$\begin{aligned} \mathit{Error}_\ell(\lambda, r) = \mathbf{E}\left[\left(\frac{m_r(h_\ell)+1}{a} - \sigma\right)^2\right] = \\ \mathbf{E}\left[\frac{1}{a^2}\left(m_r(h_\ell)^2 + 2m_r(h_\ell) + 1\right) - \frac{2\sigma}{a}\left(1 + m_r(h_\ell)\right) + \sigma^2\right]. \end{aligned} \tag{5.8}$$

Using (5.5), it holds that

$$a = \frac{3 - 2r - r^\ell}{1 - r}. \tag{5.9}$$

Observe now that a depends on the decay parameter r and the sequence length ℓ. Using

the linearity property of expectation, Equation (5.8) can be rewritten as:

$$\begin{aligned} \mathit{Error}_\ell(\lambda, r) = \frac{1}{a^2}\mathbf{E}\left[m_r(h_\ell)^2\right] + \frac{2}{a^2}\mathbf{E}\Big[m_r(h_\ell)\Big] + \frac{1}{a^2} \\ - \frac{2}{a}\mathbf{E}\left[\sigma\right] - \frac{2}{a}\mathbf{E}\Big[\sigma m_r(h_\ell)\Big] + \mathbf{E}\left[\sigma^2\right]. \end{aligned} \tag{5.10}$$

In order to express the above error in terms of the real model λ and the decay r, it is needed to express $\mathbf{E}\big[m_r(h_\ell)^2\big]$, $\mathbf{E}\big[m_r(h_\ell)\big]$, $\mathbf{E}\big[\sigma m_r(h_\ell)\big]$, $\mathbf{E}\big[\sigma\big]$, and $\mathbf{E}\big[\sigma^2\big]$ in terms of the parameters of the real model λ and r. We start with evaluating $\mathbf{E}\big[m_r(h_\ell)\big]$.

Using the definition of $m_r(h_\ell)$ given by (5.3) and the linearity of the expectation operator, we have

$$\mathbf{E}\left[m_r(h_\ell)\right] = \sum_{i=0}^{\ell-1} r^i \cdot \mathbf{E}\left[\delta_{\ell-i-1}(\mathsf{s})\right]. \tag{5.11}$$

Then, by Equation (5.4), we find that

$$\mathbf{E}\left[\delta_{\ell-i-1}(\mathsf{s})\right] = P\left(\delta_{\ell-i-1}(\mathsf{s}) = 1\right). \tag{5.12}$$

Denoting the system state at the time of observing o_i by q_i we have

$$\begin{aligned} P\left(\delta_{\ell-i-1}(\mathsf{s}) = 1\right) &= \sum_{x \in S_\lambda} P\big(q_{\ell-i-1} = x, \delta_{\ell-i-1}(\mathsf{s}) = 1\big) \\ &= \sum_{x \in S_\lambda} P\left(q_{\ell-i-1} = x\right) P\big(\delta_{\ell-i-1}(\mathsf{s}) = 1 \mid q_{\ell-i-1} = x\big). \end{aligned} \tag{5.13}$$

where S_λ is the set of states in the real model λ.

Let the *state success probabilities vector*, $\boldsymbol{\Theta}_\lambda$, be defined as the column vector

$$\boldsymbol{\Theta}_\lambda = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \tag{5.14}$$

where θ_q is the probability of observing s given the system is in state q. Notice that these probabilities are given together with λ, viz., $B_q(\mathsf{s})$ from Definition 2.5.1. As we focus on steady state behaviours, exploiting the properties of the stationary distribution $\boldsymbol{\pi}_\lambda$, Equation (5.13) can be rewritten as the scalar product of $\boldsymbol{\pi}_\lambda$ and $\boldsymbol{\Theta}_\lambda$:

$$P\left(\delta_{\ell-i-1}(\mathsf{s}) = 1\right) = \sum_{x \in S_\lambda} \pi_x \theta_x = \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda \tag{5.15}$$

Substituting in Equation (5.12) yields

$$\mathbf{E}\big[\delta_{\ell-i-1}(\mathsf{s})\big] = \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda \ , \tag{5.16}$$

and substituting in (5.11) yields

$$\mathbf{E}\big[m_r(h_\ell)\big] = \sum_{i=0}^{\ell-1} r^i \cdot \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda \, . \tag{5.17}$$

Since $\boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda$ is independent of r, the *geometric series* summation rule is used to evaluate the sum in the above equation, and obtain:

$$\mathbf{E}\big[m_r(h_\ell)\big] = \left(\frac{1-r^\ell}{1-r}\right) \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda \, . \tag{5.18}$$

Isolating the dependency on ℓ, the above equation is written as follows

$$\mathbf{E}\big[m_r(h_\ell)\big] = \frac{\boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda}{1-r} + \epsilon_1(\ell), \tag{5.19}$$

where

$$\epsilon_1(\ell) = -r^\ell \frac{\boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda}{1-r} \, . \tag{5.20}$$

We now move on to simplify $\mathbf{E}\big[m_r(h_\ell)^2\big]$, the next addend to $\mathit{Error}(\lambda, r)$. By the definition of $m_r(h_\ell)$ in Equation (5.3), and using the linearity of expectation, it is obtained that

$$\begin{aligned}
\mathbf{E}\left[m_r(h_\ell)^2\right] &= \mathbf{E}\left[\left(\sum_{i=0}^{\ell-1} r^i \delta_{\ell-i-1}(\mathsf{s})\right)^2\right] \\
&= \mathbf{E}\left[\sum_{i_1=0}^{\ell-1}\sum_{i_2=0}^{\ell-1} r^{i_1+i_2}\, \delta_{\ell-i_1-1}(\mathsf{s})\, \delta_{\ell-i_2-1}(\mathsf{s})\right] \\
&= \sum_{i_1=0}^{\ell-1}\sum_{i_2=0}^{\ell-1} r^{i_1+i_2} \cdot \mathbf{E}\left[\delta_{\ell-i_1-1}(\mathsf{s})\, \delta_{\ell-i_2-1}(\mathsf{s})\right] .
\end{aligned} \tag{5.21}$$

In fact, from the definition of $\delta_i(\mathsf{s})$ given by (5.4) above, it is obvious that

$$\mathbf{E}\big[\delta_{\ell-i_1-1}(\mathsf{s})\, \delta_{\ell-i_2-1}(\mathsf{s})\big] = P\big(\delta_{\ell-i_1-1}(\mathsf{s}) = 1, \delta_{\ell-i_2-1}(\mathsf{s}) = 1\big).$$

Substituting in Equation (5.21) we get

$$\begin{aligned}
\mathbf{E}\left[m_r(h_\ell)^2\right] &= \sum_{i_1=0}^{\ell-1}\sum_{i_2=0}^{\ell-1} r^{i_1+i_2} P\left(\delta_{\ell-i_1-1}(\mathsf{s})=1, \delta_{\ell-i_2-1}(\mathsf{s})=1\right) \\
&= \sum_{i=0}^{\ell-1} r^{2i} P\big(\delta_{\ell-i-1}(\mathsf{s})=1\big) + 2\sum_{i_1=0}^{\ell-2}\sum_{i_2=i_1+1}^{\ell-1} r^{i_1+i_2} P\big(\delta_{\ell-i_1-1}(\mathsf{s})=1, \delta_{\ell-i_2-1}(\mathsf{s})=1\big) \\
&= \sum_{i=0}^{\ell-1} r^{2i} P\big(\delta_{\ell-i-1}(\mathsf{s})=1\big) + 2\sum_{i=0}^{\ell-2}\sum_{k=1}^{\ell-1-i} r^{2i+k} P\big(\delta_{\ell-i-1}(\mathsf{s})=1, \delta_{\ell-(i+k)-1}(\mathsf{s})=1\big) \\
&= \sum_{i=0}^{\ell-1} r^{2i} P\left(\delta_{\ell-i-1}(\mathsf{s})=1\right) + 2\sum_{i=0}^{\ell-2} r^{2i}\sum_{k=1}^{\ell-1-i} r^{k} P\big(\delta_{\ell-i-1}(\mathsf{s})=1, \delta_{\ell-i-1-k}(\mathsf{s})=1\big).
\end{aligned} \tag{5.22}$$

Using the notation $\hat{\imath} = \ell - i - 1$, the above equation is written as follows

$$\mathbf{E}\left[m_r(h_\ell)^2\right] = \sum_{i=0}^{\ell-1} r^{2i} P\left(\delta_{\hat{\imath}}(\mathsf{s})=1\right) + 2\sum_{i=0}^{\ell-2} r^{2i}\sum_{k=1}^{\ell-1-i} r^{k} P\big(\delta_{\hat{\imath}}(\mathsf{s})=1, \delta_{\hat{\imath}-k}(\mathsf{s})=1\big). \tag{5.23}$$

Note now that $P\left(\delta_{\hat{\imath}}(\mathsf{s})=1, \delta_{\hat{\imath}-k}(\mathsf{s})=1\right)$ is the joint probability of observing s at times $\hat{\imath}$ and $\hat{\imath}-k$. This probability can be expressed as

$$\begin{aligned}
P\big(\delta_{\hat{\imath}}(\mathsf{s})=1, \delta_{\hat{\imath}-k}(\mathsf{s})=1\big) &= \sum_{x\in S_\lambda}\sum_{y\in S_\lambda} P\big(q_{\hat{\imath}}=x, \delta_{\hat{\imath}}(\mathsf{s})=1, q_{\hat{\imath}-k}=y, \delta_{\hat{\imath}-k}(\mathsf{s})=1\big) \\
&= \sum_{x\in S_\lambda} P\big(q_{\hat{\imath}}=x\big)\, P\left(\delta_{\hat{\imath}}(\mathsf{s})=1 \mid q_{\hat{\imath}}=x\right)\cdot \\
&\qquad \cdot \sum_{y\in S_\lambda} P\big(q_{\hat{\imath}-k}=y \mid q_{\hat{\imath}}=x\big)\, P\big(\delta_{\hat{\imath}-k}(\mathsf{s})=1 \mid q_{\hat{\imath}-k}=y\big).
\end{aligned} \tag{5.24}$$

We can rewrite (5.24) in terms of the state stationary probabilities vector $\boldsymbol{\pi}_\lambda$ and the state success probabilities vector $\boldsymbol{\Theta}_\lambda$, given by Equations (5.1) and (5.14), respectively.

$$P\big(\delta_{\hat{\imath}}(\mathsf{s})=1,\ \delta_{\hat{\imath}-k}(\mathsf{s})=1\big) = \sum_{x\in S_\lambda} \pi_x\theta_x \sum_{y\in S_\lambda} P\big(q_{\hat{\imath}-k}=y \mid q_{\hat{\imath}}=x\big)\ \theta_y\,. \tag{5.25}$$

The above equation can be simplified further by making use of the *time reversal* model of λ (cf. Brémaud, 1998; Norris, 1997) which, informally speaking, represents the same model λ when time runs 'backwards.' If λ's state transition probability matrix is $\boldsymbol{A}_\lambda = (A_{ij} \mid i,j = 1,\ldots,n)$ then λ's *reverse state transition* probability matrix is:

$$\boldsymbol{A}'_\lambda = \begin{bmatrix} A'_{11} & A'_{12} & \ldots & \ldots \\ A'_{21} & \ddots & \ldots & \ldots \\ \vdots & \ldots & A'_{xy} & \vdots \\ \ldots & \ldots & \ldots & A'_{nn} \end{bmatrix} \tag{5.26}$$

where A'_{xy} is the probability that the previous state is y given that current state is x. Clearly, $\boldsymbol{A}'_\lambda$ is derived from $\boldsymbol{A}_\lambda$ by the identity:

$$A'_{xy} = \frac{\pi_y}{\pi_x} A_{yx} \tag{5.27}$$

which exist as, by the irreducibility of λ, all π_x are strictly positive. It is easy to prove that $\boldsymbol{A}'_\lambda$ is a stochastic matrix, and is irreducible when $\boldsymbol{A}_\lambda$ is such. Now, observing that $P\left(q_{i-k} = y \mid q_i = x\right)$ is the probability that the kth previous state is y given that the current state is x, Eq. (5.25) can be written in terms of $\boldsymbol{\pi}_\lambda$, $\boldsymbol{\Theta}_\lambda$ and $\boldsymbol{A}'_\lambda$ as

$$P\left(\delta_i(\mathsf{s}) = 1, \delta_{i-k}(\mathsf{s}) = 1\right) \;=\; \left(\boldsymbol{\pi}_\lambda \times {\boldsymbol{\Theta}_\lambda}^T\right) {\boldsymbol{A}'_\lambda}^k \, \boldsymbol{\Theta}_\lambda \,, \tag{5.28}$$

where we use symbol $\times$ to denote the '*entry-wise*' product of matrices. Let us now return to Equation (5.23) and replace $P\big(\delta_i(\mathsf{s}) = 1\big)$ and $P\left(\delta_i(\mathsf{s}) = 1,\, \delta_{i-k}(\mathsf{s}) = 1\right)$ in it using expressions (5.15) and (5.28), respectively.

$$\mathbf{E}\left[m_r(h_\ell)^2\right] = \sum_{i=0}^{\ell-1} r^{2i} \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda + 2 \sum_{i=0}^{\ell-2} r^{2i} \sum_{k=1}^{\ell-i-1} \left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right) \left(r \boldsymbol{A}'_\lambda\right)^k \; \boldsymbol{\Theta}_\lambda \tag{5.29}$$

Using the summation rule for geometric series, Equation (5.29) can be simplified to the following expression

$$\begin{aligned} \mathbf{E}\big[m_r(h_\ell)^2\big] = \left(\frac{1 - r^{2\ell}}{1 - r^2}\right) \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda + \\ 2 \sum_{i=0}^{\ell-2} r^{2i} \left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right) \left(r \boldsymbol{A}'_\lambda - \left(r \boldsymbol{A}'_\lambda\right)^{\ell-i}\right) \left(\boldsymbol{I} - r \boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta} \,, \end{aligned} \tag{5.30}$$

where $\boldsymbol{I}$ is the identity matrix of size n. Applying the geometric series rule again, the above equation can be rewritten as

$$\begin{aligned} \mathbf{E}\big[m_r(h_\ell)^2\big] = \left(\frac{1 - r^{2\ell}}{1 - r^2}\right) \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda + 2r \left(\frac{1 - r^{2\ell-2}}{1 - r^2}\right) \left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right) \boldsymbol{A}'_\lambda \left(\boldsymbol{I} - r \boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta} \\ - 2r^\ell \sum_{i=0}^{\ell-2} r^i \left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right) \left({\boldsymbol{A}'_\lambda}^{\ell-i}\right) \left(\boldsymbol{I} - r \boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta} \,. \end{aligned} \tag{5.31}$$

Isolating the terms which depend on ℓ, we write the above equation as follows

$$\mathbf{E}\big[m_r(h_\ell)^2\big] = \frac{\boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda}{1 - r^2} + \frac{2r}{1 - r^2} \left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right) \boldsymbol{A}'_\lambda \left(\boldsymbol{I} - r \boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta}_\lambda + \epsilon_2(\ell), \tag{5.32}$$

where

$$\epsilon_2(\ell) = \left(\frac{-r^{2\ell}}{1-r^2}\right)\boldsymbol{\pi}_\lambda\boldsymbol{\Theta}_\lambda + 2\left(\frac{-r^{2\ell-1}}{1-r^2}\right)\left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right)\left(\boldsymbol{A}'_\lambda\right)\left(\boldsymbol{I} - r\boldsymbol{A}'_\lambda\right)^{-1}\boldsymbol{\Theta} - 2r^\ell\sum_{i=0}^{\ell-2} r^i\left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right)\left(\boldsymbol{A}'^{\,\ell-i}_\lambda\right)\left(\boldsymbol{I} - r\boldsymbol{A}'_\lambda\right)^{-1}\boldsymbol{\Theta} \tag{5.33}$$

Notice that in the formulation above we use an inverse matrix, whose existence is proved by the following lemma.

Lemma 5.1. *For $\boldsymbol{A}$ a stochastic matrix and $0 < r < 1$, matrix $(\boldsymbol{I} - r\boldsymbol{A})$ is invertible.*

Proof. We prove equivalently that

$$Det\Big(\boldsymbol{I} - r\boldsymbol{A}\Big) \neq 0 \tag{5.34}$$

By multiplying (5.34) by the scalar $-r^{-1}$, we reduce it to the equivalent condition

$$-\frac{1}{r}\cdot Det\Big(\boldsymbol{I} - r\boldsymbol{A}\Big) \quad = \quad Det\Big(\boldsymbol{A} - \frac{1}{r}\boldsymbol{I}\Big) \quad \neq \quad 0$$

Observe that $Det\big(\boldsymbol{A} - r^{-1}\boldsymbol{I}\big)$ is the characteristic polynomial of $\boldsymbol{A}$ evaluated on r^{-1}, which is zero if and only if r^{-1} is an eigenvalue of $\boldsymbol{A}$. Since $\boldsymbol{A}$ has no negative entry, it follows from Perron-Frobenius theorem (cf., e.g., Horn and Johnson, 1985) that all its eigenvalues u are such that

$$|\,u\,| \leq \max_i \sum_{k=1}^{n} A_{ik}.$$

As $\boldsymbol{A}$ is stochastic and $r^{-1} > 1$, this concludes our proof. □

We now turn our attention to $\mathbf{E}\left[\sigma m_r(h_\ell)\right]$, with σ the probability that the next outcome is s. As σ depends on the current state $q_{\ell-1}$, the expectation $\mathbf{E}\big[\sigma m_r(h_\ell)\big]$ can be expressed as

$$\mathbf{E}\left[\sigma m_r(h_\ell)\right] = \mathbf{E}\left[R(x)\right], \tag{5.35}$$

with $R(x)$ defined for $x \in S_\lambda$ by

$$R(x) = \mathbf{E}\left[\sigma m_r(h_\ell) \mid q_{\ell-1} = x\right]. \tag{5.36}$$

In other words, $R(x)$ is the conditional expected value of $\sigma m_r(h_\ell)$ given that the current state is x.

We define the *state predictive success probabilities vector* $\boldsymbol{\Phi}_\lambda$ as the following column

vector.

$$\mathbf{\Phi}_\lambda = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{bmatrix} \tag{5.37}$$

where ϕ_x is the probability that the next outcome after a state transition is s, given that the current state is x. The entries of $\mathbf{\Phi}_\lambda$ can be computed by

$$\phi_x = \sum_{y \in S_\lambda} A_{xy}\theta_y,$$

and therefore

$$\mathbf{\Phi}_\lambda = \boldsymbol{A}_\lambda \mathbf{\Theta}_\lambda \,. \tag{5.38}$$

Using the above, we can rewrite Equation (5.36) as

$$R(x) = \mathbf{E}\Big[\phi_x m_r(h_\ell) \;\Big|\; q_{\ell-1} = x\Big], \tag{5.39}$$

for $x \in S_\lambda$. Substituting $m_r(h_\ell)$ with its definition in (5.3), we obtain

$$\begin{aligned} R(x) &= \mathbf{E}\Big[\phi_x \sum_{i=0}^{\ell-1} r^i \delta_{\ell-i-1}(\mathsf{s}) \;\Big|\; q_{\ell-1} = x\Big] \\ &= \phi_x \mathbf{E}\Big[\sum_{i=0}^{\ell-1} r^i \delta_{\ell-i-1}(\mathsf{s}) \;\Big|\; q_{\ell-1} = x\Big]. \end{aligned} \tag{5.40}$$

Using the linearity of expectation, we then get

$$R(x) = \phi_x \sum_{i=0}^{\ell-1} r^i \mathbf{E}\Big[\delta_{\ell-i-1}(\mathsf{s}) \;\Big|\; q_{\ell-1} = x\Big]. \tag{5.41}$$

Since the possible values of $\delta_{\ell-i-1}(\mathsf{s})$ are only 0 and 1, we have

$$\mathbf{E}\Big[\delta_{\ell-i-1}(\mathsf{s}) \;\Big|\; q_{\ell-1} = x\Big] = P\Big(\delta_{\ell-i-1}(\mathsf{s}) = 1 \;\Big|\; q_{\ell-1} = x\Big).$$

Thus Equation (5.41) can be written as

$$\begin{aligned} R(x) &= \phi_x \sum_{i=0}^{\ell-1} r^i P\Big(\delta_{\ell-i-1}(\mathsf{s}) = 1 \;\Big|\; q_{\ell-1} = x\Big) \\ &= \phi_x \sum_{i=0}^{\ell-1} r^i \sum_{y \in S_\lambda} P\Big(q_{\ell-i-1} = y \;\Big|\; q_{\ell-1} = x\Big)\, P\Big(\delta_{\ell-i-1}(\mathsf{s}) = 1 \;\Big|\; q_{\ell-i-1} = y\Big) \\ &= \phi_x \sum_{i=0}^{\ell-1} r^i \sum_{y \in S_\lambda} P\Big(q_{\ell-i-1} = y \;\Big|\; q_{\ell-1} = x\Big)\, \theta_y \,. \end{aligned} \tag{5.42}$$

We now return to Equation (5.35) which expresses $\mathbf{E}\big[\sigma m_r(h_\ell)\big]$ and, making use again of the stationary distribution, substitute the expression above for $R(x)$

$$\begin{aligned}\mathbf{E}\big[\sigma m_r(h_\ell)\big] &= \sum_{x\in S_\lambda} P\left(q_{\ell-1}=x\right) R(x) = \sum_{x\in S_\lambda} \pi_x R(x) \\ &= \sum_{x\in S_\lambda} \pi_x\phi_x \sum_{i=0}^{\ell-1} r^i \sum_{y\in S_\lambda} P\left(q_{\ell-i-1}=y \mid q_{\ell-1}=x\right)\ \theta_y\,. \end{aligned} \tag{5.43}$$

Exchanging the summations in the above equation, we get

$$\mathbf{E}\big[\sigma m_r(h_\ell)\big] = \sum_{i=0}^{\ell-1} r^i \sum_{x\in S_\lambda} \pi_x\phi_x \sum_{y\in S_\lambda} P\left(q_{\ell-i-1}=y \mid q_{\ell-1}=x\right)\ \theta_y\,. \tag{5.44}$$

Comparing the above with Equations (5.25) and (5.28), we similarly obtain

$$\begin{aligned}\mathbf{E}\big[\sigma m_r(h_\ell)\big] &= \sum_{i=0}^{\ell-1} r^i \left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Phi}_\lambda^T\right) {\boldsymbol{A}'_\lambda}^i\, \boldsymbol{\Theta}_\lambda \\ &= \left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Phi}_\lambda^T\right) \left(\sum_{i=0}^{\ell-1} (r\boldsymbol{A}'_\lambda)^i\right) \boldsymbol{\Theta}_\lambda. \end{aligned} \tag{5.45}$$

As before, by Lemma 5.1, we can simplify the above formula as

$$\mathbf{E}\big[\sigma m_r(h_\ell)\big] = \left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Phi}_\lambda^T\right) \left(\boldsymbol{I} - (r\boldsymbol{A}'_\lambda)^\ell\right) \left(\boldsymbol{I} - r\boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta}_\lambda. \tag{5.46}$$

Isolating the term which depends on ℓ, we rewrite the above equation as follows

$$\mathbf{E}\big[\sigma m_r(h_\ell)\big] = \left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Phi}_\lambda^T\right) \left(\boldsymbol{I} - r\boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta}_\lambda + \epsilon_3(\ell), \tag{5.47}$$

where

$$\epsilon_3(\ell) = -r^\ell \left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Phi}_\lambda^T\right) (\boldsymbol{A}'_\lambda)^\ell \left(\boldsymbol{I} - r\boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta}_\lambda\,. \tag{5.48}$$

Let us now consider $\mathbf{E}\left[\sigma\right]$

$$\mathbf{E}\left[\sigma\right] = \sum_{x\in S_\lambda} P\left(q_{\ell-1}=x\right) \phi_x = \sum_{x\in S_\lambda} \pi_x\phi_x = \boldsymbol{\pi}_\lambda \boldsymbol{\Phi}_\lambda.$$

Substituting $\boldsymbol{\Phi}$ in the above equation by its definition in (5.38), we get

$$\mathbf{E}\left[\sigma\right] = \boldsymbol{\pi}_\lambda \boldsymbol{A}_\lambda \boldsymbol{\Theta}_\lambda\,. \tag{5.49}$$

Using the eigenvector property of $\boldsymbol{\pi}_\lambda$ in Equation (5.1) we obtain

$$\mathbf{E}\left[\sigma\right] = \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda. \tag{5.50}$$

Finally, let us evaluate $\mathbf{E}\left[\sigma^2\right]$

$$\mathbf{E}\left[\sigma^2\right] = \sum_{x \in S_\lambda} P\left(q_{\ell-1} = x\right) {\phi_x}^2 = \sum_{x \in S_\lambda} \pi_x {\phi_x}^2 = \boldsymbol{\pi}_\lambda \left(\boldsymbol{\Phi}_\lambda \times \boldsymbol{\Phi}_\lambda\right). \tag{5.51}$$

We can now in the end return to the error formula (5.10) and substitute the expressions we have so derived for its various components, viz., Equations (5.32), (5.19), (5.47), (5.50) and (5.51). We therefore obtain the following formula for the expected Beta estimation error

$$\begin{aligned} \mathit{Error}_\ell\left(\lambda, r\right) = & \frac{1}{a^2}\left(\frac{\boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda}{1-r^2} + \frac{2r}{1-r^2}\left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right) \boldsymbol{A}'_\lambda \left(\boldsymbol{I} - r\boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta}_\lambda\right) \\ & + \frac{2}{a^2}\left(\frac{\boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda}{1-r}\right) - \frac{2}{a}\left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Phi}_\lambda^T\right)\left(\boldsymbol{I} - r\boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta}_\lambda \\ & - \frac{2}{a}\boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda + \boldsymbol{\pi}_\lambda \left(\boldsymbol{\Phi}_\lambda \times \boldsymbol{\Phi}_\lambda\right) + \frac{1}{a^2} \\ & + \frac{2}{a^2}\epsilon_1(\ell) + \frac{1}{a^2}\epsilon_2(\ell) - \frac{2}{a}\epsilon_3(\ell), \end{aligned} \tag{5.52}$$

where $\epsilon_1(\ell)$, $\epsilon_2(\ell)$, and $\epsilon_3(\ell)$ are given by equations (5.20), (5.33), and (5.48) respectively. Also a is given by (5.9). Now, as we are interested in the asymptotic error, we evaluate the limit of the above error when $\ell \to \infty$.

$$\mathit{Error}\left(\lambda, r\right) = \lim_{\ell \to \infty} \mathit{Error}_\ell\left(\lambda, r\right). \tag{5.53}$$

Since $r < 1$, it is obvious that

$$\lim_{\ell \to \infty} \epsilon_1(\ell) = \lim_{\ell \to \infty} \epsilon_2(\ell) = \lim_{\ell \to \infty} \epsilon_3(\ell) = 0,$$

and

$$\lim_{\ell \to \infty} a = \frac{3-2r}{1-r}.$$

Therefore, and using a few algebraic manipulations we get our final asymptotic error formula for the beta model with exponential decay.

$$\begin{aligned} \mathit{Error}\left(\lambda, r\right) = & \frac{\left(1-r\right)\left(4r^2-3\right)}{\left(1+r\right)\left(3-2r\right)^2}\boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda + \left(\frac{1-r}{3-2r}\right)^2 \\ & + \frac{2\left(1-r\right)r}{\left(3-2r\right)^2\left(1+r\right)}\left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right) \boldsymbol{A}'_\lambda \left(\boldsymbol{I} - r\boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta}_\lambda \\ & - 2\left(\frac{1-r}{3-2r}\right)\left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Phi}_\lambda^T\right)\left(\boldsymbol{I} - r\boldsymbol{A}'_\lambda\right)^{-1} \boldsymbol{\Theta}_\lambda + \boldsymbol{\pi}_\lambda \left(\boldsymbol{\Phi}_\lambda \times \boldsymbol{\Phi}_\lambda\right). \end{aligned} \tag{5.54}$$

5.5 System stability

The stability of a system is, informally speaking, its tendency to remain in the same state. In this section we describe the effect of system stability on the expected Beta estimation error derived in Section 5.4. In particular, we show that if a system is very stable, then the expected Beta estimation error tends to 0 as the decay r tends to 1; as the limit of the decay model for $r \to 1$ is indeed the unmodified Beta model, this means that when systems are very stable, the unmodified Beta model achieves better prediction than any decay model.

We introduce the notion of *state stability* which we define as the probability of transition to the same state. Formally, given a HMM λ with set of states S_λ, the stability of a state $x \in S_\lambda$ is defined as

$$\mathit{Stability}\,(x) = P\,(q_{t+1} = x \mid q_t = x) = A_{xx}.$$

Building on that, we define the *system stability* of λ at time t, as

$$\mathit{Stability}_t\,(\lambda) = P\,(q_{t+1} = q_t)\,,$$

that is the probability that the system remains at time $t+1$ in the same state where it has been at time t. System stability can therefore be expressed as

$$\mathit{Stability}_t\,(\lambda) = \sum_{x \in S_\lambda} P\,(q_t = x)\,A_{xx}. \tag{5.55}$$

Note that the system stability depends on the diagonal elements of the transition matrix $\boldsymbol{A}_\lambda$. It also depends on the probability distribution over system states at the time t. Assuming as before that the system is ergodic (cf. Definitions 2.1 and 2.6), when t tends to ∞ the probability distribution over the system states converges to the stationary probability distribution $\boldsymbol{\pi}_\lambda$. We call the system stability when $t \to \infty$ the *asymptotic* system stability, and denote it by $\mathit{Stability}_\infty(\lambda)$.

$$\mathit{Stability}_\infty\,(\lambda) = \sum_{x \in S_\lambda} \pi_x A_{xx}. \tag{5.56}$$

As the stationary probability distribution $\boldsymbol{\pi}_\lambda$ over states depends on the state transition matrix $\boldsymbol{A}_\lambda$ — see Equation (5.1) — the asymptotic system stability of λ is thus determined by the transition matrix $\boldsymbol{A}_\lambda$.

Regarding the analysis of the effect of the system stability on the estimation, obviously the error formula (5.54) is too complex to allow an analytical study of its curve. However, given a particular system model with a specific stability, the beta estimation error can be evaluated for different values of the decay factor r, which allows us to build sound

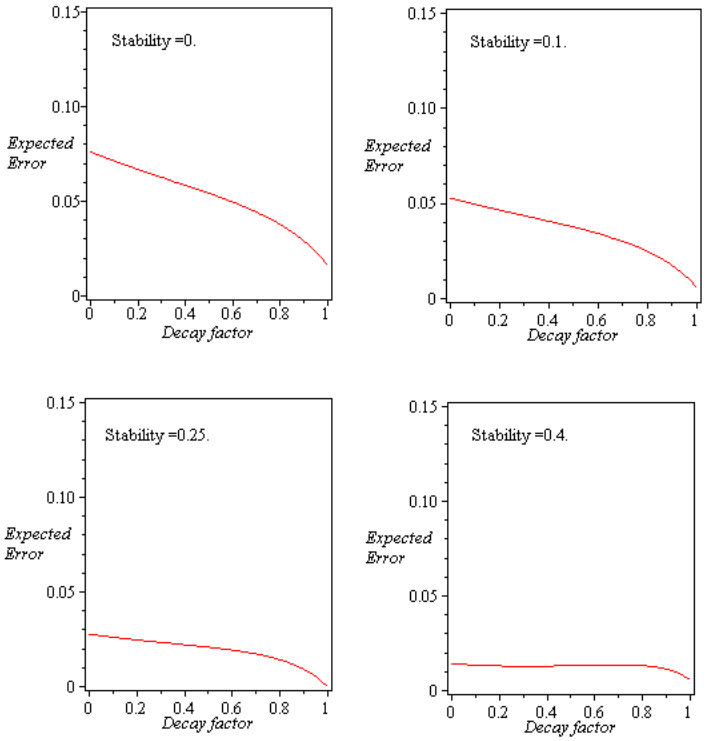


FIGURE 5.1: Expected Beta estimation error versus decay factor given stability < 0.5

intuitions about the impact of stability on the beta estimation mechanism.

Consider the model λ with the stability s where

$$\boldsymbol{A}_\lambda = \begin{bmatrix} s & \frac{1-s}{3} & \frac{1-s}{3} & \frac{1-s}{3} \\ \frac{1-s}{3} & s & \frac{1-s}{3} & \frac{1-s}{3} \\ \frac{1-s}{3} & \frac{1-s}{3} & s & \frac{1-s}{3} \\ \frac{1-s}{3} & \frac{1-s}{3} & \frac{1-s}{3} & s \end{bmatrix} \tag{5.57}$$

Given the above transition matrix, it can be easily verified that

$$\boldsymbol{\pi}_\lambda = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}. \tag{5.58}$$

Let the success probabilities vector $\boldsymbol{\Theta}_\lambda$ be defined by

$$\boldsymbol{\Theta}_\lambda = \begin{bmatrix} 1.0 \\ 0.7 \\ 0.3 \\ 0.0 \end{bmatrix} \tag{5.59}$$

Figure 5.1 shows the expected Beta estimation error when the system λ is unstable ($s < 0.5$). It is obvious that the minimum error value is obtained when the decay

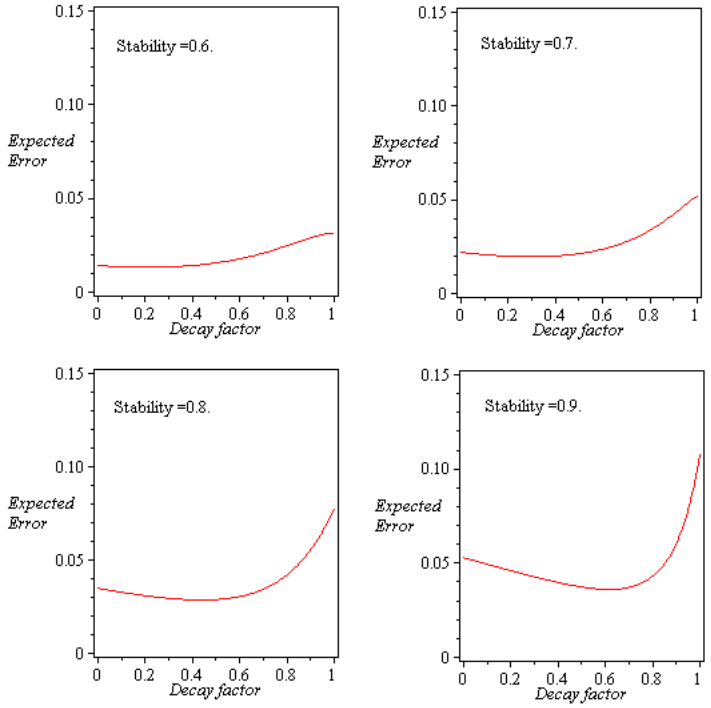


FIGURE 5.2: Expected Beta estimation error versus decay factor given stability > 0.5

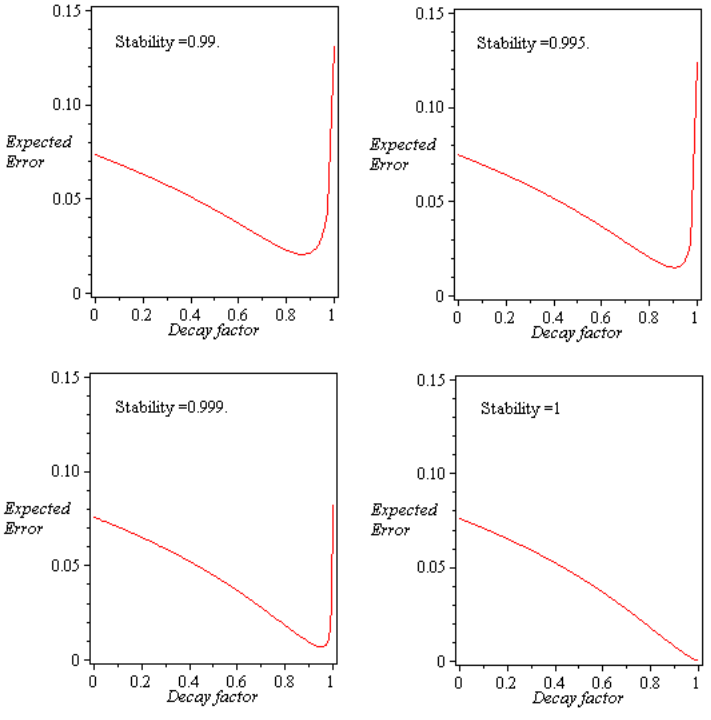


FIGURE 5.3: Expected Beta estimation error versus decay factor given stability > 0.9

r tends to 1. The reason for this is that an unstable system is relatively unlikely to stay in the same state, and therefore unlikely to preserve the previous distribution over observations. If the estimation uses low values for the decay, then the resulting estimate for the predictive probability distribution is close to the previous distribution; this is unlikely to be the same as in the next time instant, due to instability. On the other hand, using a decay r tending to 1 favours equally all previous observations, and according to the following theorem the resulting probability distribution is almost surely (with probability 1) the average of the distributions exhibited by the model states. Such an average provides a better estimate for the predictive probability distribution than approximating the distribution of the most recent set of states using low decay values.

Theorem 5.2. *Given unbounded sequences generated by a HMM λ, the beta estimate for the predictive probability with decay $r \to 1$ is given by $\boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda$ almost surely, where $\boldsymbol{\pi}_\lambda$ and $\boldsymbol{\Theta}_\lambda$ are the stationary probability distribution and success probabilities vectors of λ, respectively.*

Proof. Given an ℓ-length sequence h_ℓ, let the 'nondecayed' beta estimate, denoted by $\mathcal{B}(\mathsf{s} \mid h_\ell)$, be defined as follows.

$$\mathcal{B}(\mathsf{s} \mid h_\ell) = \lim_{r \to 1} \mathcal{B}_r(\mathsf{s} \mid h_\ell)$$

By Equations (5.2), (5.3), and (5.4), the above Beta estimate can be expressed as follows

$$\mathcal{B}(\mathsf{s} \mid h_\ell) = \frac{\left(\sum_{i=0}^{\ell-1} \delta_{\ell-i-1}(\mathsf{s})\right) + 1}{\ell + 2} = \frac{\left(\sum_{k=0}^{\ell-1} \delta_k(\mathsf{s})\right) + 1}{\ell + 2}\,. \tag{5.60}$$

Since the HMM λ is assumed to be ergodic, the proof is now completed by applying the ergodic theorem 2.15. Let the real valued function $f(h_\ell)$ be defined on sequences h_ℓ as follows

$$f(h_\ell) = \delta_0(\mathsf{s}).$$

Then, using the time shift operator described by Equation (2.27), and the notation in (2.28), we can write

$$f\left(\tau^k(h_\ell)\right) = \delta_k(\mathsf{s}).$$

Since it holds that

$$\mathbf{E}\left[f(h_\ell)\right] = \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda < \infty$$

by Equation (5.16), it follows by the ergodic theorem that

$$\frac{1}{\ell} \sum_{k=0}^{\ell-1} \delta_k(\mathsf{s}) \to \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda \qquad \text{almost surely.} \tag{5.61}$$

Using the above equation to express the convergence of $\mathcal{B}(\mathsf{s} \mid h_\ell)$ given by (5.60), we get

$$\mathcal{B}(\mathsf{s} \mid h_\ell) \to \boldsymbol{\pi}_\lambda \boldsymbol{\Theta}_\lambda \qquad \text{almost surely.}$$

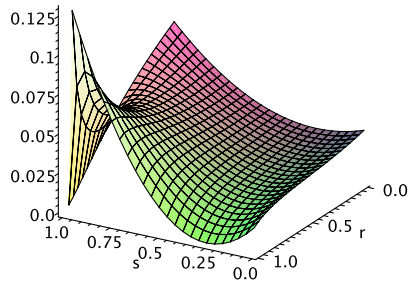
□

It is worth noticing that when $s = 1/|S_\lambda|$, the minimum expected beta error is 0, when $r \to 1$. In this case all elements of $\boldsymbol{A}_\lambda$ are equal and therefore, the predictive probability of success is $\sum_{x \in S_\lambda} \theta_x / |S_\lambda|$, regardless of the current state. In other words, the whole behaviour can effectively be modelled by a single probability distribution over observations. The best approximation for this probability distribution is achieved by considering the entire history using decay $r \to 1$, because in this way the beta estimate converges to the correct predictive distribution according to Theorem 5.2.

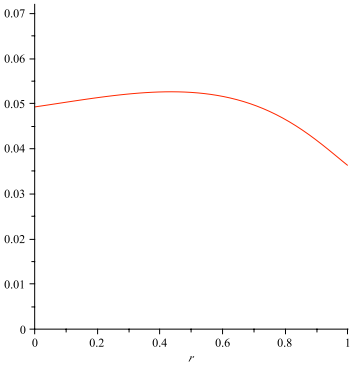
Systems which are relatively stable (i.e., with $s > 0.5$) are more likely to stay in the same state rather than transitioning to a new state. In such case, approximating the probability distribution of a state by observing systems interactions provides a good estimate for the predictive probability distribution. However, the quality of the approximation depends heavily on the choice of an optimum value for decay. If the decay is too small, the sequence of observation considered in the computation will prove too short to reflect the correct distribution precisely. If otherwise the decay is too large (i.e., too close to 1), then the resulting estimate approaches the average probability distribution as described above. Figure 5.2 above shows the expected beta estimation error when the system λ is relatively stable.

Figure 5.3 shows the expected beta estimation error for very stable systems, i.e., systems with $s > 0.9$. In such case, observe that the expected estimation error is very sensitive to the choice of the decay value. In fact, regarded as a function of s and r, the error formula is pathological around point $(1, 1)$. Observe that the formula is undefined for $r = 1$, because in such a case all matrices $(\boldsymbol{I} - r\boldsymbol{A}')$ are singular. Worse than that, there is *no* limit as s and r tend to 1, as the limiting value depends on the relative speed of s and r. This is illustrated in Figure 5.4, which plots $\mathit{Error}(\lambda, r)$ over the open unit square for our running four-state model. A simple inspection of (5.54), with the support of Lemma 5.1, shows that *Error* is continuous and well behaved on its domain, as illustrated by the top-left plot. Yet, the cusp near $(1, 1)$ –which is also noticeable in graphs of Figure 5.3– reflects its erratic behaviour in that neighbourhood. The remaining three graphs of Figure 5.4 show that the error function for $s \mapsto 1$ and $r \mapsto 1$ tends to different values along different lines, and therefore prove that it admits no limit at $(1, 1)$. However, if stability is actually 1, the minimum expected estimation error tends to 0, and the optimum decay value (which corresponds to the minimum expected estimation error) tends to 1. The following theorem proves this observation formally.

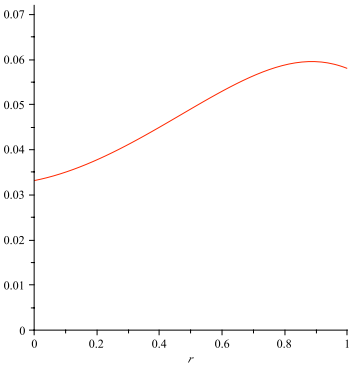
Theorem 5.3. *Let λ be a HMM. If $\mathit{Stability}_\infty(\lambda)$ tends to 1, then the asymptotic beta*



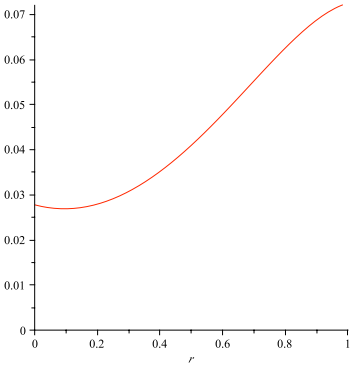
The error as a function of s and r



The error along the line $4s = r + 3$



The error along the line $2s = r + 1$



The error along the line $4s = 3r + 1$

FIGURE 5.4: Expected Beta estimation error for the four-states model

estimation error tends to 0 *when the decay* r *tends to* 1.

Proof. The asymptotic stability of a given system λ tends to 1 (i.e., a perfectly stable system) if and only if all the diagonal elements of $\boldsymbol{A}_\lambda$ tend to 1; this means that $\boldsymbol{A}_\lambda$ tends to the identity matrix $\boldsymbol{I}$. As the latter is *not* irreducible, we first need to prove that the error formula (5.54) remains valid for $s = 1$. In fact, irreducibility plays its role in our assumption that the initial state distribution $\boldsymbol{\pi}_\lambda$ is stable, which is obviously true in the case of $\boldsymbol{I}$ because for any initial vector $\boldsymbol{\pi}$, it holds

$$\boldsymbol{\pi}\, \boldsymbol{I} = \boldsymbol{\pi}.$$

All the steps in the derivation can then be repeated verbatim, with the exception of (5.27), which is undefined. Yet, it can easily be verified that $\boldsymbol{I}'_\lambda$ exists and is the identity matrix. We can therefore evaluate the expected beta estimation error in this case by replacing $\boldsymbol{A}'_\lambda$ by the identity matrix $\boldsymbol{I}$ in (5.54), while remembering that

$(\boldsymbol{I} - r\boldsymbol{I})^{-1} = \boldsymbol{I}(1-r)^{-1}$ and $\boldsymbol{\Phi}_\lambda = \boldsymbol{I}\boldsymbol{\Theta}_\lambda = \boldsymbol{\Theta}_\lambda$. We get,

$$\begin{aligned} \mathit{Error}\,(\lambda, r) = & \frac{(1-r)\left(4r^2-3\right)}{(1+r)\,(3-2r)^2}\boldsymbol{\pi}_\lambda\boldsymbol{\Theta}_\lambda + \left(\frac{1-r}{3-2r}\right)^2 \\ & + \frac{2\,(1-r)\,r}{(3-2r)^2\,(1+r)}\left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right)\frac{1}{1-r}\boldsymbol{\Theta}_\lambda \\ & - 2\left(\frac{1-r}{3-2r}\right)\left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right)\frac{1}{1-r}\boldsymbol{\Theta}_\lambda + \boldsymbol{\pi}_\lambda\left(\boldsymbol{\Theta}_\lambda \times \boldsymbol{\Theta}_\lambda\right) \end{aligned} \tag{5.62}$$

Then, observing that

$$\left(\boldsymbol{\pi}_\lambda \times \boldsymbol{\Theta}_\lambda^T\right)\boldsymbol{\Theta}_\lambda = \boldsymbol{\pi}_\lambda\left(\boldsymbol{\Theta}_\lambda \times \boldsymbol{\Theta}_\lambda\right)\ ,$$

we obtain

$$\begin{aligned} \mathit{Error}\,(\lambda, r) = & \frac{(1-r)\left(4r^2-3\right)}{(1+r)\,(3-2r)^2}\boldsymbol{\pi}_\lambda\boldsymbol{\Theta}_\lambda + \left(\frac{1-r}{3-2r}\right)^2 \\ & + \left(\frac{2r}{(3-2r)^2\,(1+r)} - \frac{2}{3-2r} + 1\right)\boldsymbol{\pi}_\lambda\left(\boldsymbol{\Theta}_\lambda \times \boldsymbol{\Theta}_\lambda\right) \end{aligned} \tag{5.63}$$

and thus

$$\begin{aligned} \mathit{Error}\,(\lambda, r) = & \frac{(1-r)\left(4r^2-3\right)}{(1+r)\,(3-2r)^2}\boldsymbol{\pi}_\lambda\boldsymbol{\Theta}_\lambda + \left(\frac{1-r}{3-2r}\right)^2 \\ & + \frac{(1-r)\left(3-4r^2\right)}{(1+r)\,(3-2r)^2}\boldsymbol{\pi}_\lambda\left(\boldsymbol{\Theta}_\lambda \times \boldsymbol{\Theta}_\lambda\right)\ . \end{aligned} \tag{5.64}$$

By inspection of the error formula above, when $r \to 1$, the expected beta estimation error obviously tends to 0. That is, when the given system is stable, zero expected estimation error is achieved by choosing the decay r tending to 1, which is the same as saying dropping the decay altogether and using the unmodified Beta model. □

5.6 Discussion

This chapter has focussed on the exponential decay principle in the context of probabilistic trust as a way to endow the well-known and widely-used Beta model with appropriate mechanisms to account for dynamic behaviours. The main conclusion is that, despite the attention the Beta model has received in the literature and its undoubted success 'on-the-ground,' the assumption that principals can be represented by a single immutable probability distribution is untenable in the real world.

Although this thesis in general advocates fully-fledged 'stateful' models, such as the hidden Markov models, the purpose in this chapter was to ascertain to what extent the decay principle put forward by some authors can provide the required support for principals whose behaviour changes according to their (discrete) state transitions. In

doing so, this chapter has described some mathematical properties of the Beta model with exponential decay scheme, which suggest that the scheme will not be ideal in all scenarios.

A formula has been then derived for the expected error of the Beta scheme with respect to a representation of the 'real model' as a hidden Markov model, which can be used by algorithm developers to understand the implications of choosing a decay factor. Finally, we have exemplified one such analysis by plotting the error formula as a function of the decay parameter r according to a notion of system stability. The evidence obtained for the exercise, can be roughly summarised by saying that the choice of the 'right' parameter r remains highly sensitive and critical, and that anyway the choice of a decay scheme over the unmodified Beta model appears sensible only when systems are relatively stable, so that state changes happen rather infrequently.

The analysis is valid under the assumption of the ergodicity of the underlying Markov chain, which in the case of finite-state systems reduces to just irreducibility and aperiodicity. Observe that the states of the model can be grouped in maximal classes –known in the literature as 'communicating'– whereby each state is reachable from any other state in the same class. By definition, reducible chains admit multiple maximal classes; every run of the system will eventually be 'trapped' in one of such classes, after which its steady-state behaviour will be described by the irreducible (sub)chain consisting of only the states in that class. As the given analysis focusses on asymptotic behaviours only, this indicates that when the chain is reducible it may be sufficient to analyse each of the (sub)models determined by the maximal irreducible communicating classes in the model. The situation is more complex if the model fails to be aperiodic, as this indicates cyclic asymptotic behaviours and, potentially, causal dependencies between events, whereby a probabilistic analysis may anyway not be the best option.

Chapter 6

HMM-based trust model

In this chapter, we introduce the *HMM-based trust model* as an approach to evaluating trust in principals exhibiting dynamic behaviour, i.e. changing their behaviour over time. This model is also presented in our recent paper (ElSalamouny et al. (2010)). The HMM-based trust model is based on approximating the behaviour of any given principal p by a finite-state HMM η_p, called the *approximate behaviour model* for the given principal p. Given any sequence h_T of outcomes of interactions with p, the probability distribution over the potential outcomes of the next interaction with p can be estimated using the p's approximate behaviour model η_p. We call this estimated probability distribution the *estimated predictive probability distribution* of p. Following the existing notion of probabilistic trust (described in Section 3.2), the estimated predictive probability distribution defines the trust in the principal p.

In order to precisely define the HMM-based trust model, it is required to define a method for computing the approximate behaviour model η for principal p, and also a method for estimating the predictive probability distribution given a sequence h_T of length T. As a general notation which will be used in these definitions we will write the probability of any random variable ζ, under a given probabilistic model R, as $P(\zeta \mid R)$.

The process of computing η for a principal p is basically *learning* the behaviour of p. This is naturally performed by finding the probabilistic model which best 'fits' a given sample sequence of outcomes of interactions with p. Here, the *maximum likelihood criterion*, described by Section 2.2.2, is adopted as the criterion for the model fitness. Specifically, let $y = y_1\, y_2 \cdots y_T$ be an observed sequence of outcomes of interactions with a given principal, where T is an arbitrary length. Let also $\mathcal{R}_n$ denote any n-state HMM. Then, using the sequence y, the n-state approximate behaviour model η is obtained by the following equation.

$$\eta = \underset{\mathcal{R}_n}{\operatorname{argmax}}\, P\left(h_T = y \mid \mathcal{R}_n\right). \tag{6.1}$$

That is η is the n-state HMM under which the probability of the given sequence y is

maximised. The HMM model η can be therefore obtained by the Baum-Welch algorithm which is described in Section 2.5.4 and detailed by Baum et al. (1970); Rabiner (1989).

Now we address the problem of estimating the predictive probability distribution given a particular sequence of outcomes. Let $h_T = o_1\, o_2 \cdots o_T$ be a random variable representing any sequence of observed outcomes of interaction with the principal p, where o_1 and o_T represent respectively the least and the most recent outcomes, and T is an arbitrary length. Extending this notation to future outcomes, the outcome of the next interaction with p is denoted by o_{T+1}. Note that each outcome o_t is therefore a random variable representing the outcome at time t. Let also $V = \{1, 2, \ldots, K\}$ be the alphabet of each single outcome. Using the n-state approximate behaviour HMM η defined by Equation (6.1), the estimated predictive probability distribution given a particular sequence of outcomes w is denoted by $\mathcal{H}_\eta(. \mid w)$ and defined by the following equation.

$$\mathcal{H}_\eta\left(z \mid w\right) = P\left(o_{T+1} = z \mid h_T = w, \eta\right) = \frac{P\left(h_T = w, o_{T+1} = z \mid \eta\right)}{P\left(h_T = w \mid \eta\right)}\,. \tag{6.2}$$

where $z \in V$. The above probabilities are efficiently evaluated by the forward-backward algorithm described in Section 2.5.3, and detailed by Rabiner (1989).

Like other existing probabilistic trust models, the objective of the HMM-based trust model is to estimate the predictive probability distribution for any given principal p, that is the probability of each possible outcome in the next interaction with p. Therefore it is a fundamental requirement that the approximate behaviour model η, computed for the principal p is chosen such that the divergence of the estimated predictive distribution from the real predictive distribution, also called the *estimation error*, is minimised.

To analyse this error, we need to model the real behaviour of the principal p. This allows expressing the *real* predictive probability distribution of p. Given a particular sequence h of observations about p, the estimation error can be therefore evaluated as the statistical difference between the real and estimated predictive probability distributions.

The next section describes modelling the real behaviour of a principal in a network of multiple principals, and distinguishes between two observable behaviours of a principal. In Section 6.2, we turn our attention to defining and analysing the estimation error incurred by applying the HMM-based trust evaluation described above, and discuss the consistency of this trust evaluation approach. Section 6.3 describes a simulation-based comparison between the HMM-based and Beta based trust models. We follow in Section 6.4 by discussing the likelihood convergence property of HMMs upon which the reliability of the HMM-based trust model is based.

6.1 Modelling the real system

Given the same reasons described in Section 5.2 we model the real behaviour of any principal p by a HMM λ, which we call the *real behaviour model* of p. Possible behaviour states p are modelled by corresponding states in λ. The transition of p from one state to another is modelled by a state transition between the corresponding states in λ.

The state of the real model λ at the time of observing o_t is denoted by the random variable q_t. Thus, for the real HMM λ, given that the current underlying state is x, i.e. $q_T = x$, we can compute the real predictive probability distribution, denoted by $P(. \mid x, \lambda)$, that is the probability of each possible next observation, $z \in V$, using the following equation.

$$\begin{aligned} P(z \mid x, \lambda) &= P(o_{T+1} = z \mid q_T = x, \lambda) \\ &= \sum_{y \in S_\lambda} P(q_{T+1} = y \mid q_T = x, \lambda)\, P(o_{T+1} = z \mid q_{T+1} = y, \lambda) \\ &= \sum_{y \in S_\lambda} (\boldsymbol{A}_\lambda)_{xy} (\boldsymbol{B}_\lambda)_{yz} \ . \end{aligned} \tag{6.3}$$

where S_λ, $\boldsymbol{A}_\lambda$, and $\boldsymbol{B}_\lambda$ are respectively the set of states, the state transition matrix, and the emission matrix of λ. The reader is referred to Section 2.5.1 for details about these parameters. We shall also work under the hypothesis that λ is *ergodic*. This corresponds to demanding that the Markov chain underlying λ is irreducible and aperiodic (more details on these properties are given by Grimmet and Stirzaker (2001); Norris (1997); Brémaud (1998), and also by Section 2.4). In the following we distinguish between two types of behaviours, the *general* and *relative* behaviours of a trustee.

6.1.1 General and relative behaviour

In many cases, the behaviour of a principal *te* depends on its internal state which is determined by a combination of *te*'s internal attributes. These attributes may include *te*'s security (whether or not *te* is compromised by an attacker), and also *te*'s reliability which is determined by available computational resources, e.g. processing power and memory. The change of the principal's security attribute is governed by the robustness of its defence system against external attacks, while the change of its reliability attribute depends on the performance of its operating system which determines the likelihood of overloading and crashes, and also the likelihood of recovery given that the system is overloaded.

Example 6.1. *Figure 6.1 depicts a markov chain modelling the overall state transitions of a principal, considering only its security S and reliability R attributes. If the principal is secure ($S = 1$) at a particular interaction, it can become insecure ($S = 0$) at the next interaction with probability* 0.1. *Given it is insecure in a particular interaction, it can*

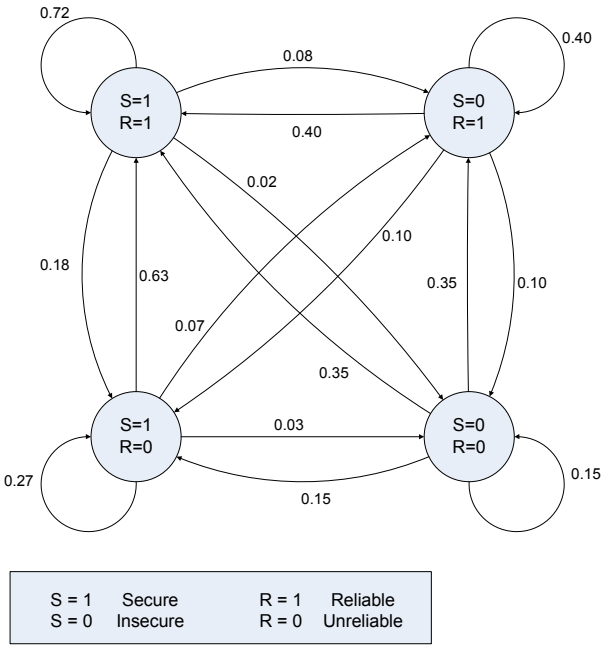


FIGURE 6.1: State transition system for Example 6.1

be recovered to be secure in the next interaction with probability 0.5. *If the principal is reliable (has sufficient resources to handle an interaction) at a particular interaction* ($R = 1$)*, it can become unreliable* ($R = 0$) *in the next interaction with probability* 0.2. *It can also recover from unreliability to reliability with probability* 0.7. *The overall state transition probabilities are computed assuming the transitions of* S *and* R *attributes are mutually independent.*

The long term dynamic behaviour of *te* is hence driven by an assumed probabilistic evolution of its internal state, where the probability distribution over possible outcomes of a particular interaction involving *te* is determined by the current state of *te*. If for instance, the principal is insecure (compromised by an external attack) or it is unreliable due to overloading, it may exhibit Denial of Service (DoS) where it responds successfully to requests from other principals with a low probability, whereas under normal conditions (secure and reliable), it responds successfully to the same requests with high probability. In many systems including this example, the outcome of an interaction between *te* and another principal x depends only on the internal overall state of *te* regardless of the identity of the partner x, especially in anonymous transactions where the identity of the interaction partner x is hidden from *te*. We restrict our attention to these systems where the behaviour of a principal depends only on its internal state regardless of the partner. If the possible states of *te* are finite, then its dynamic behaviour can be then modelled by a finite-state HMM which we call the *te*'s *general behaviour model*. Note that the sequence of outcomes of all interactions involving *te* is a realisation of such *te*'s

general behaviour model.

Definition 6.1. The *general behaviour model* λ_{te} of a principal *te* is the HMM which is assumed to generate the sequence of outcomes of all interactions which involve *te*.

On the other hand, a particular truster *tr*, which aims at evaluating the trust in *te*, observes a possibly different behaviour, the *te*'s behaviour *relative* to *tr*. This behaviour is characterised by the sequence of outcomes resulting from only personal interactions between *tr* and *te*. Since we define the trust of *tr* in *te* as the probability distribution over possible outcomes of the next interaction between *tr* and *te*, the relative behaviour of *te* to *tr* is the basis of evaluating this trust, and therefore we need to describe the relative behaviour by the following theorem.

Theorem 6.2. *Let the HMM* $\lambda_{te} = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B})$ *be the general behaviour model of a principal te, where the parameters* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are independent of interacting partners. Let also* p_x *be the probability that a principal* x *is the te's partner at a given interaction. Then, the probability of any sequence of outcomes observed only by* x *is given by a HMM* $\lambda_{tex} = (\boldsymbol{\pi}'\boldsymbol{A}', \boldsymbol{B}')$*, called the relative behaviour model of te with respect to* x*, where*

$$\begin{aligned} \boldsymbol{A}' &= p_x \boldsymbol{A} \left(\boldsymbol{I} - (1-p_x)\,\boldsymbol{A}\right)^{-1}, \\ \boldsymbol{\pi}' &= \boldsymbol{\pi}.\boldsymbol{A}', \\ \boldsymbol{B}' &= \boldsymbol{B}. \end{aligned}$$

Proof. Let x interact with *te* at time τ. Then, the next interaction between x and *te* occurs at time $\tau+1$ with probability p_x, and at time $\tau+2$ with probability $(1-p_x)p_x$. In general, the next interaction between x and *te* occurs at time $\tau+k$ with probability $(1-p_x)^{k-1}p_x$. Let *te* be in state u at time τ (when interacting with x). Then at time $\tau+k$, *te* will be in state v with probability $(\boldsymbol{A}^k)_{uv}$, that is the uvth entry of the matrix $\boldsymbol{A}$ to the power k. As the state transition of *te* is independent of the identity of its interaction partner, we express the probability of the joint event $e_{te,x}(u,k,v)$ that the next interaction between x and *te* occurs at time $\tau+k$ and *te* makes a k-step state transition from u to v.

$$P\left(e_{te,x}\left(u,k,v\right)\right) = (1-p_x)^{k-1}p_x.(\boldsymbol{A}^k)_{uv}.$$

Since the events $e_{te,x}(u,1,v), e_{te,x}(u,2,v), \ldots$ are mutually exclusive, the probability $\boldsymbol{A'}_{uv}$ that *te* will be in state v when the next interaction between *te* and x occurs, is evaluated by summing the probabilities of the above events.

$$\begin{aligned} \boldsymbol{A'}_{uv} &= \sum_{k=1}^{\infty} P(e_{te,x}(u,k,v)) \\ &= \sum_{k=1}^{\infty} (1-p_x)^{k-1}p_x.(\boldsymbol{A}^k)_{uv} \end{aligned}$$

Moving from the matrix entry notation to the matrix notation, and using the summation rule of geometric series we obtain the following equation.

$$\boldsymbol{A}' = \sum_{k=1}^{\infty}(1-p_x)^{k-1}p_x.(\boldsymbol{A}^k) \tag{6.4}$$

$$= p_x\boldsymbol{A}\left(\boldsymbol{I}-(1-p_x)\boldsymbol{A}\right)^{-1}. \tag{6.5}$$

where $\boldsymbol{I}$ is the identity matrix. Assuming $p_x > 0$, note that the matrix $(\boldsymbol{I}-(1-p_x)\boldsymbol{A})$ in the above equation is invertable by Lemma 5.1. Indeed if $p_x = 0$, i.e. the principal x never interacts with te, the behaviour of te is entirely hidden from x. In this case the te's relative behaviour to x is undefined.

Given that $\boldsymbol{\pi}$ is the initial probability distribution over te's states, the probability distribution over the states when the first interaction with x occurs can be evaluated using $\boldsymbol{A}'$.

$$\boldsymbol{\pi}' = \boldsymbol{\pi}.\boldsymbol{A}'.$$

Lastly, since observation probabilities are dependent only on the current state of te, the observation probability matrix exhibited when interacting with x is the same, that is

$$\boldsymbol{B}' = \boldsymbol{B}.$$

□

It is worth noting that if $\boldsymbol{\pi}$ is the stationary distribution of $\boldsymbol{A}$, then $\boldsymbol{\pi}$ is also the stationary distribution of $\boldsymbol{A}'$. This is verifiable by Equation (6.4) and the fact that $\boldsymbol{\pi}\boldsymbol{A}^k = \boldsymbol{\pi}$:

$$\begin{aligned}\boldsymbol{\pi}\boldsymbol{A}' &= \boldsymbol{\pi}\sum_{k=1}^{\infty}(1-p_x)^{k-1}p_x.(\boldsymbol{A}^k)\\ &= \boldsymbol{\pi}\sum_{k=1}^{\infty}(1-p_x)^{k-1}p_x\\ &= \boldsymbol{\pi}. \end{aligned}\tag{6.6}$$

By Theorem 6.2, te's relative behaviour observed by a principal x in the environment can be described by a HMM λ_{tex} whose parameters depend on the general behaviour of te and also the probability p_x at which, the principal x interacts with te. Based on this result, the HMM-based trust framework is applicable to the sequence of outcomes observed only by the principal x when interacting with te. Specifically, within the scope of interaction between x and te, the real predictive probability distribution is computed by substituting the real behaviour model λ in (6.3) by λ_{tex}. Thus, in our analysis of the HMM-based trust, the real behaviour model λ will always refer to the relative behaviour model of the trustee te, seen by a particular principal x.

Another important implication of Theorem 6.2 is that the relative behaviour of *te* is the same to all principals who have the same probability p of interacting with *te*. That is the real model of *te* from the perspective of each of these principals is the same HMM λ. This is useful as the information carried by observation sequences collected by these principals can add together to give a better estimate about the real HMM model λ. This idea is employed to enhance the reliability of HMM-based trust evaluation by using multiple sequences as shown later in Section 6.4.1, and also to describe a reputation model in Chapter 7.

6.2 The estimation error

As mentioned earlier in this chapter, the *estimation error* incurred when estimating the predictive probability distribution is quantified as the statistical difference between the real and estimated predictive distributions. In general there exist different measures for quantifying the statistical difference between two probability distributions. A fundamental property of any measure of statistical difference between two probability distributions μ and ν is that it is equal to zero if and only if μ and ν are identical.

Some of these measures are symmetric, e.g. the *total variation distance* and the L_p *distance* (also known as the *p-norm*) (Rachev, 1991). Symmetric measures include also the *quadratic distance* defined in Section 5.4 for the Beta estimation error. Because these measures hold the property of symmetry, each of them is referred to as a *distance* between the two probability distributions, in analogy with the *euclidean distance* between two points in the euclidean space. That is the distance from the distribution μ to distribution ν is the same as the distance from ν to μ. More details on symmetric measures and their properties can be found in (Rachev, 1991) along with their applications in the probability theory.

There are also non-symmetric measures for the difference between two probability distributions. These measures are used in many applications to measure *'how far'* an estimated (or empirical) distribution $\hat{\mu}$ is from a theoretical or real distribution μ which is aimed to be estimated or approximated by $\hat{\mu}$. Therefore each of these measures is referred to as *divergence* measure. That is, it is important here to distinguish the real distribution from which the *'estimated'* distribution diverges. As it is required to evaluate the quality of HMM-based trust model in estimating the predictive probability, one of such measures, known as the *relative entropy divergence* (also *Kullback-Leibler divergence*) (cf. Cover and Thomas, 2006) is adopted to quantify the estimation error, that is the divergence of the estimated predictive distribution from the real predictive distribution.

With λ a real model, the real predictive probability distribution depends only on the current state of λ because of the *Markovian* property of λ (i.e. a state transition prob-

ability depends only on the current state). Thus, given the current state is x, the real predictive probability is denoted by $P(. \mid x, \lambda)$, and evaluated by Equation (6.3). Using the approximate behaviour model η as the parameter for the HMM-trust model, and given a sequence $h_T = w$ of observations, the estimated predictive probability distribution, is denoted by $\mathcal{H}_\eta(. \mid w)$, and given by Equation (6.2). Now the estimation error, expressed as the relative entropy divergence from the real predictive distribution to the estimated predictive probability distribution is written as follows

$$D_{KL}\left(P(. \mid x, \lambda) \mid\mid \mathcal{H}_\eta(. \mid w)\right) = \sum_{z \in V} P(z \mid x, \lambda) \log\left(\frac{P(z \mid x, \lambda)}{\mathcal{H}_\eta(z \mid w)}\right). \tag{6.7}$$

Observe that the above divergence depends on the current state q_T of λ, and the random sequence of outcomes h_T. The estimation error is therefore a random variable. To assess the long-term estimation quality of the HMM-based trust model for estimating the predictive probability distribution, we study in the next subsection the *expected estimation error*, which is invariant of both the current state q_T and the given sequence h_T, and depends only on the real model λ and the parameter η of the HMM-based trust model.

6.2.1 Analysis of the expected estimation error for HMM-based trust model

The *expected estimation error*, denoted by $Error_T(\lambda, \mathcal{H}_\eta)$ is the expected value of the estimation error given by Equation (6.7), where the expectation is evaluated on the underlying random variables q_T and h_T. Thus

$$Error_T(\lambda, \mathcal{H}_\eta) = \mathbf{E}\left[D_{KL}\left(P(. \mid q_T, \lambda) \mid\mid \mathcal{H}_\eta(. \mid h_T)\right)\right]. \tag{6.8}$$

In the following it is formally shown that adopting the maximum likelihood criterion, defined by Equation (6.1), for choosing the approximate behaviour model η minimises the expected estimation error of HMM-based trust model, and therefore is a consistent method for choosing the parameters of the HMM-based trust model.

Equation (6.8) can be written as follows

$$Error_T(\lambda, \mathcal{H}_\eta) = \sum_{w \in V^T} \sum_{x \in S_\lambda} P(h_T = w, q_T = x \mid \lambda) \cdot \cdot D_{KL}\left(P(. \mid x, \lambda) \mid\mid \mathcal{H}_\eta(. \mid w)\right). \tag{6.9}$$

Using Equation (6.7) we rewrite the above equation.

$$Error_T(\lambda, \mathcal{H}_\eta) = \sum_{w \in V^T} \sum_{x \in S_\lambda} P(h_T = w, q_T = x \mid \lambda) \cdot \cdot \sum_{z \in V} P(z \mid x, \lambda) \log \left(\frac{P(z \mid x, \lambda)}{\mathcal{H}_\eta(z \mid w)} \right). \tag{6.10}$$

Substituting $P(z \mid x, \lambda)$ and $\mathcal{H}_\eta(z \mid w)$ using Equations (6.3) and (6.2) respectively, we write the above equation as follows

$$\begin{aligned} Error_T(\lambda, \mathcal{H}_\eta) &= \sum_{w \in V^T} \sum_{x \in S_\lambda} P(h_T = w, q_T = x \mid \lambda) \cdot \\ &\quad \cdot \sum_{z \in V} P(o_{T+1} = z \mid q_T = x, \lambda) \log \left(\frac{P(o_{T+1} = z \mid q_T = x, \lambda)}{P(o_{T+1} = z \mid h_T = w, \eta)} \right) \\ &= \sum_{w \in V^T} \sum_{x \in S_\lambda} \sum_{z \in V} P(o_{T+1} = z \mid q_T = x, \lambda) \cdot \\ &\quad \cdot P(h_T = w, q_T = x \mid \lambda) \log \left(\frac{P(o_{T+1} = z \mid q_T = x, \lambda)}{P(o_{T+1} = z \mid h_T = w, \eta)} \right). \end{aligned} \tag{6.11}$$

Since the next outcome o_{T+1} depends only on the current state q_T regardless of the history sequence h_T, we have

$$P(o_{T+1} = z \mid q_T = x, \lambda) = P(o_{T+1} = z \mid h_T = w, q_T = x, \lambda). \tag{6.12}$$

Thus Equation (6.11) becomes

$$\begin{aligned} Error_T(\lambda, \mathcal{H}_\eta) &= \sum_{w \in V^T} \sum_{x \in S_\lambda} \sum_{z \in V} P(o_{T+1} = z \mid h_T = w, q_T = x, \lambda) \cdot \\ &\quad \cdot P(h_T = w, q_T = x \mid \lambda) \log \left(\frac{P(o_{T+1} = z \mid q_T = x, \lambda)}{P(o_{T+1} = z \mid h_T = w, \eta)} \right) \\ &= \sum_{w \in V^T} \sum_{x \in S_\lambda} \sum_{z \in V} P(o_{T+1} = z, h_T = w, q_T = x \mid \lambda) \cdot \\ &\quad \cdot \log \left(\frac{P(o_{T+1} = z \mid q_T = x, \lambda)}{P(o_{T+1} = z \mid h_T = w, \eta)} \right). \end{aligned} \tag{6.13}$$

The above equation can be simplified to the following equation

$$Error_T(\lambda, \mathcal{H}_\eta) = \mathbf{E}[\log P(o_{T+1} \mid q_T, \lambda)] - \mathbf{E}[\log P(o_{T+1} \mid h_T, \eta)]. \tag{6.14}$$

Observe that the first term in the above equation depends only on the real behaviour model λ, while the second term depends on both the real and approximate behaviour models λ and η. Denoting the first and second terms respectively by $C_T(\lambda)$ and

$H_T(\lambda, \eta)$, we rewrite the above equation as follows.

$$Error_T(\lambda, \mathcal{H}_\eta) = C_T(\lambda) - H_T(\lambda, \eta). \tag{6.15}$$

Assuming that $(\boldsymbol{A}_\eta)_{ij} > 0$ for any states i, j, that is the state transition probabilities of η are strictly positive, it has been proved by Baum and Petrie (1966) that the following limit exists.

$$\lim_{T\to\infty} H_T(\lambda, \eta) = H(\lambda, \eta). \tag{6.16}$$

Observe also that the limit $\lim_{T\to\infty} C_T(\lambda) = C(\lambda)$ exists. This is because the ergodicity of λ implies that the distribution of the random variable q_T converges to a stationary (fixed) distribution according to which the expectation $\mathbf{E}[\log P(o_{T+1} \mid q_T, \lambda)]$ is evaluated. The convergence of both $C_T(\lambda)$ and $H_T(\lambda, \eta)$ implies the convergence of the expected estimation error (as $T \to \infty$) to an *asymptotic estimation error* denoted by $Error(\lambda, \mathcal{H}_\eta)$, and expressed as follows

$$Error(\lambda, \mathcal{H}_\eta) = C(\lambda) - H(\lambda, \eta). \tag{6.17}$$

Also, by Theorem 3.2 in (Baum and Petrie, 1966) the log-probability of any observation sequence h_T is related to $H(\lambda, \eta)$ as follows

$$\frac{1}{T}\log P(h_T \mid \eta) \overset{a.s.}{\to} H(\lambda, \eta). \tag{6.18}$$

The above equation means that the log-probability of a random sequence h_T under the approximate model η, divided by its length converges *almost surely* to $H(\lambda, \eta)$. Here *'almost surely'* (also known as *'almost everywhere'* and *'with probability 1'*) convergence means that the probability that the function $\frac{1}{T}\log P(h_T \mid \eta)$ converges to the above limit is 1. That is

$$P\left(\lim_{T\to\infty} \frac{1}{T}\log P(h_T \mid \eta) = H(\lambda, \eta)\right) = 1.$$

Equation (6.18) implies that choosing an approximate model η which maximises the probability of a sufficiently long sequence h_T almost surely maximises $H(\lambda, \eta)$, and therefore reduces the asymptotic estimation error given by Equation (6.17). Thus, the maximum data likelihood criterion, expressed by Equation (6.1) is a consistent method to obtain the approximate behaviour model, which is then used to estimate the predictive probability distribution.

6.3 Comparison with Beta-based trust with decay principle

In this section we contrast HMM-based trust model described above against the existing Beta-based trust model with exponential decay, described by Jøsang and Ismail (2002)

and Sections 3.2.1,3.2.5 and analysed in Chapter 5, in terms of the expected estimation error. In order to perform this comparison, it is essential to unify the estimation error measure for the two models.

In the case of Beta-based trust model, we derived an analytical expression for the expected Beta estimation error, parametrized upon the decay factor. This estimation error is based on the *quadratic distance* between the real and estimated predictive probability distributions (see the definition in Section 5.4). The reason for choosing this particular distance measure is that each of the Beta estimated predictive probabilities (given by Equations (5.2)) is basically a sum of simple random variables ($\delta_i(X)$) weighted by the decay parameter.

Since the quadratic distance is a binomial expression, its expected value can be evaluated by propagating the expectation operator (due to its linearity) to products of the random variables ($\delta_i(X)$), and therefore the computation of the whole expectation of the quadratic distance between the real and estimated predictive distributions amounts to evaluation of the expected values for products of random variables $\delta_i(X)$ which are expressed using matrix algebra. The details of this derivation are given in Section 5.4. Although any other distance measure can be approximated by a sum of polynomials using the *Taylor's expansion*, the resulting expression for the expected Beta estimation error would be further complicated and also not exact.

In the case of HMM-based trust, on the other hand, we have no means to derive an analogous expression for the expected estimation error using the quadratic distance. This is because, each of the estimated predictive probabilities (given by Equations (6.2)), is not a sum of simple random variables as the case in Beta-trust model. It is, rather, a conditional probability of an outcome given a sequence of outcomes. By the results of Baum and Petrie (1966), the expected value of the logarithm of this conditional probability converges asymptotically to $H(\lambda, \eta)$, a characteristic of the approximate model η, given a real model λ.

Since the relative entropy divergence takes the form of the difference between log-probabilities (see Equation (6.7)), the expected value of the HMM estimation error can be expressed analytically in terms of $H(\lambda, \eta)$ by propagating the expectation operator to the log conditional probabilities. Thus the relative entropy divergence in the case of HMM-based trust is favourable as it enables analysing the expected estimation error in terms of the model parameter η. This analysis has been detailed in Section 6.2.1.

As a conclusion of the above discussion, it appears difficult to find a unified error metric which can be evaluated analytically, or even numerically, for both Beta-based and HMM-based trust models due to the nature of each of these models. So we use an HMM simulation framework to simulate the real model λ and adopt Monte Carlo methods to evaluate the expected estimation error in both models, and therefore perform the comparison between them. While we choose the relative entropy divergence as a unified error

metric for comparing between Beta and HMM based models in terms of the expected estimation errors, any other metric for the estimation error (e.g. the quadratic error) can be also evaluated using the same simulation framework because the Monte-Carlo framework is independent of the statistical difference measure as shown in the following section.

6.3.1 Monte-Carlo based evaluation of the expected estimation error

In general, any probabilistic trust model is described by an *estimating algorithm* A_σ, with a parameter σ, where the parameter (or the set of parameters) σ is specified or computed according to the trust model. For a given principal p, the estimating algorithm A_σ is fed with any observation sequence h generated by the p's real system λ and computes an estimated predictive probability distribution for p, denoted by $A_\sigma(. \mid h)$.

In the case of Beta trust model, the estimating algorithm is denoted by $\mathcal{B}_r$, where the parameter r is the decay factor, and the estimated predictive probability distribution $\mathcal{B}_r(. \mid h)$ is evaluated by Equations (5.2).

In the case of HMM-based trust model, on the other hand, the estimating algorithm is denoted by $\mathcal{H}_\eta$, where the parameter η is an approximate behaviour HMM. Note that the parameter η is obtained by maximising the probability of any sufficiently long sequence w generated by λ using Equation (6.1). Given any sequence of observations h, the estimated predictive probability distribution $\mathcal{H}_\eta(. \mid h)$ is evaluated using Equation (6.2).

Consider a HMM λ, the real model for a particular principal. Let the random variable h_T denote any generated sequence of observations of length T. Let also the random variable u_T denote the underlying hidden state sequence. Given an estimating algorithm A_σ (e.g. $\mathcal{B}_r$ or $\mathcal{H}_\eta$), the expected estimation error using A_σ is given by the following equation.

$$Error_T(\lambda, A_\sigma) = \mathbf{E}\left[D\left(P\left(. \mid u_T, \lambda\right) \mid\mid A_\sigma\left(. \mid h_T\right)\right)\right]. \tag{6.19}$$

In the above equation $D\left(P\left(. \mid u_T, \lambda\right) \mid\mid A_\sigma\left(. \mid h_T\right)\right)$ is basically the *divergence* of the estimated predictive probability distribution from the real predictive probability distribution. Plugging any of the divergence measures into Equation (6.19), the expected error can be approximated by the following Monte-Carlo procedure.

1. Simulate the real model λ_p to generate a large sample S_m of size m:

$$S_m = \{(w_1, u_1), (w_2, u_2), \ldots, (w_m, u_m)\}$$

where w_j and u_j are respectively the observation sequence, and the underlying state sequence generated in the jth simulation run.

2. For each pair (w_j, u_j),

   (a) compute both $P(. \mid u_j, \lambda)$ and $A_\sigma(. \mid w_j)$, which are the real and estimated predictive probability distributions, respectively.

   (b) Evaluate the estimation error, denoted by e_j, as

$$e_j = D\left(P(. \mid u_j, \lambda) \mid\mid A_\sigma(. \mid w_j)\right). \tag{6.20}$$

3. Approximate the required expected estimation error by evaluating the sample average:

$$Error_T(\lambda, A_\sigma) \approx \frac{1}{m} \sum_{j=1}^{m} e_j. \tag{6.21}$$

The above approximation of the expected estimation error by the sample average is based on the law of large numbers. Note that the approximation error can be made arbitrarily small by making the sample size m sufficiently large.

6.3.2 Experiments

In the following experiments, we study the effect of the system *stability*, described in Section 5.5, on both Beta estimation with a decay factor and HMM based estimation. Thus, we consider the same s-stability real model λ used in 5.5, with the observation alphabet $V = \{1, 2\}$, where the observation probability matrix is

$$\boldsymbol{B}_\lambda = \begin{bmatrix} 1.0 & 0.0 \\ 0.7 & 0.3 \\ 0.3 & 0.7 \\ 0.0 & 1.0 \end{bmatrix} \tag{6.22}$$

and the state transition matrix is

$$\boldsymbol{A}_\lambda = \begin{bmatrix} s & \frac{1-s}{3} & \frac{1-s}{3} & \frac{1-s}{3} \\ \frac{1-s}{3} & s & \frac{1-s}{3} & \frac{1-s}{3} \\ \frac{1-s}{3} & \frac{1-s}{3} & s & \frac{1-s}{3} \\ \frac{1-s}{3} & \frac{1-s}{3} & \frac{1-s}{3} & s \end{bmatrix} \tag{6.23}$$

Recall that the parameter s in the transition matrix $\boldsymbol{A}_\lambda$ is called the *system stability*, which indicates the tendency of the system to staying in the same state rather than transiting to a different one.

For simplicity, and without loss of generality, we confine our HMM-based trust model

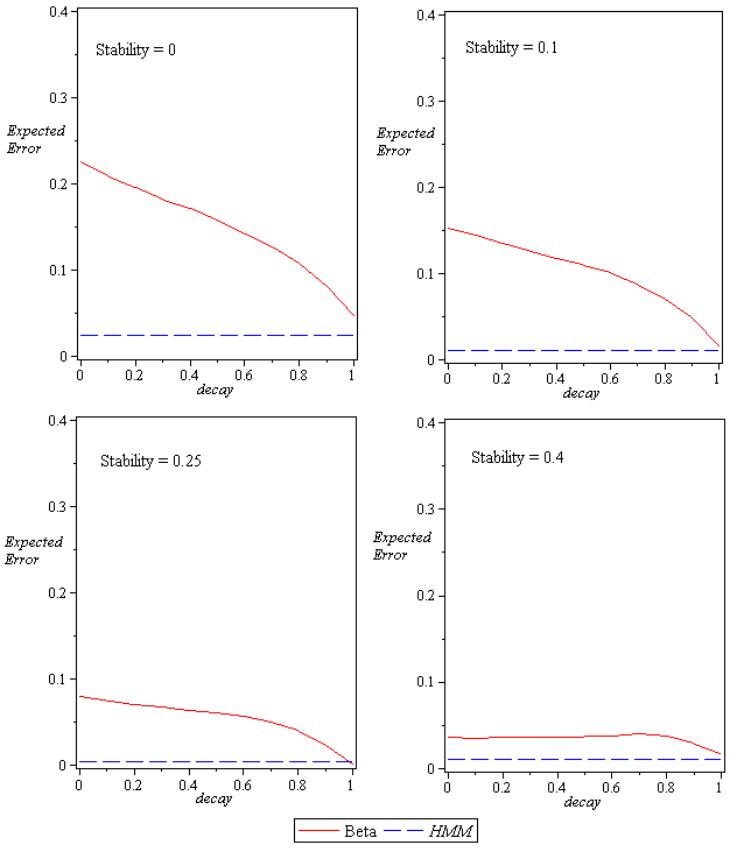


FIGURE 6.2: Beta and HMM expected estimation errors versus decay factor given stability < 0.5

to use only 2-state approximate behaviour models. In these experiments, we also base our trust estimation on sequences of length 200.

For different stability values $0 \leq s \leq 1$ and decay values $0 \leq r \leq 1$, we apply the Monte-Carlo procedure described above to evaluate the expected estimation error using both Beta ($\mathcal{B}_r$) and 2-state HMM ($\mathcal{H}_\eta$) trust algorithms. Each generated sample is of size 10000.

Figure 6.2 shows Beta and HMM expected estimation errors when the system λ is unstable ($s < 0.5$). It is obvious that the minimum error value for Beta error is obtained when the decay tends to 1. An informal explanation for this is given in Section 5.5. It is also obvious that the HMM expected estimation error is lower than Beta expected estimation error. The reason is that the 2-state HMM is a more flexible model to approximate the real HMM λ than the Beta model which is, with decay 1, equivalent to 1-state HMM model. It is worth noting that when stability is 0.25, the minimum expected beta error is 0, when the decay is 1. The HMM expected estimation error is also approximately 0. In this case all elements of the transition matrix $\boldsymbol{A}_\lambda$ are equal and therefore, the whole behaviour can effectively be modelled by a single probability distribution over observations. This single probability distribution is perfectly approximated by taking the whole history into account using Beta model with decay 1, and also with 2-state

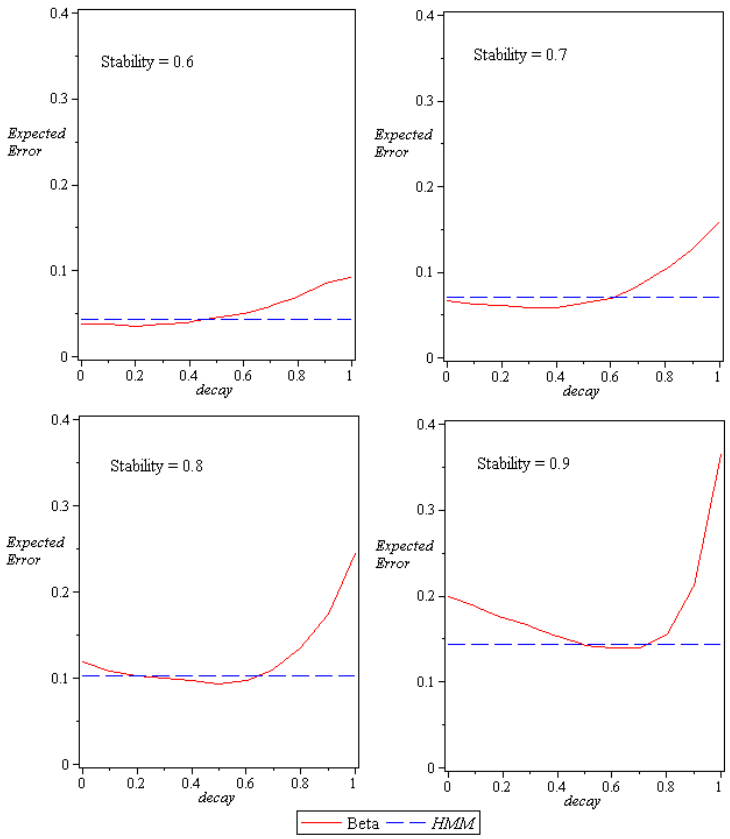


FIGURE 6.3: Beta and HMM estimation errors versus decay factor given stabilities 0.6, 0.7, 0.8, and 0.9

HMM where both states are equivalent, i.e. have the same probability distribution over observations.

Figure 6.3 shows Beta and HMM estimation errors when the system λ is stable (stability > 0.5). Observe that both Beta with decay 1 and HMM estimation errors are increasing as the stability is higher. The reason is that, at relatively high stability, old observations become irrelevant to the current behaviour which determines the real predictive probability distribution. Hence, the estimation based on the whole history using HMM or Beta with decay 1 is worse than the estimation with the same parameters when the system is unstable, where both old and recent outcomes are relevant to the current behaviour.

Observe also in the cases of high stability that HMM based estimation is better than Beta estimation for most values of decay. However, for a particular range of decay, Beta estimation is slightly better than HMM estimation. Using any decay value in this range for Beta estimation, has the effect of considering only relatively recent outcomes which characterise the current system behaviour and therefore give a better estimation for the predictive distribution. Although using any value from this specific range of decay makes Beta estimation better than HMM estimation, there is no formal means to determine this range without information about the real model λ. Figure 6.4 shows the expected

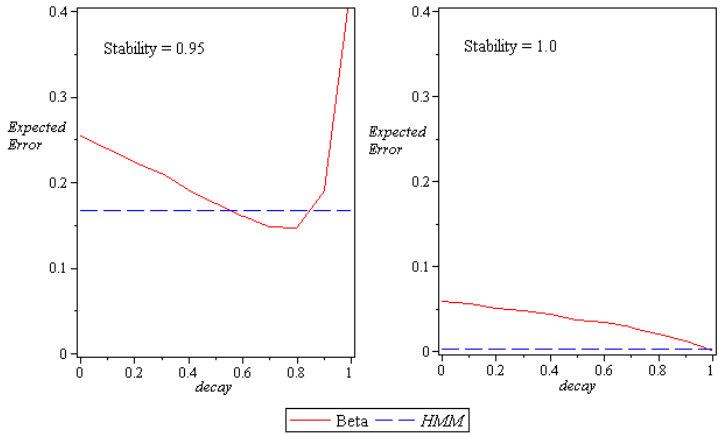


FIGURE 6.4: Beta and HMM estimation errors versus decay factor given stabilities 0.95, 1.0

estimation error when the system stability is close and equal to 1. As expected, at stability 0.95, Beta estimation still exhibits lower expected estimation error than HMM estimation only with optimal decay values. However, at stability 1, both Beta and HMM achieve 0 estimation error. In this case, the system λ does not change its state, and therefore any sequence is generated from one particular state in λ. This is equivalent to the fixed behaviour case which can be modelled by both Beta function with decay 1 and also by a HMM having equivalent states.

6.4 Likelihood convergence property

It was shown in Section 6.2.1, that the consistency of the HMM-based trust model relies on the *likelihood convergence* result for ergodic HMMs which was obtained by Baum and Petrie (1966), and stated by Equation (6.18), recalled here

$$\frac{1}{T} \log P\left(h_T \mid \eta\right) \overset{a.s.}{\to} H\left(\lambda, \eta\right).$$

For illustrating the above convergence property of ergodic HMMs, consider for instance a HMM λ having the same parameters as the s-stable HMM used in Section 5.5 with the stability parameter s set to 0.5. Let h be an observation sequence of length 2000 generated by λ. Let also η be a 2-state HMM trained on the sequence h using the Baum-Welch algorithm. Then Figure 6.5 shows the convergence of $\frac{1}{T} \log P\left(h_T \mid \eta\right)$, where h_T is a T-length sub-sequence of h and $5 \leq T \leq 2000$.

By the convergence property, the limit $H\left(\lambda, \eta\right)$ can be approximated by $\frac{1}{T} \log P\left(h_T \mid \eta\right)$ if the observation sequence h_T is sufficiently long to maintain an arbitrary approximation error margin for any model η. Using this approximation the HMM estimation error expressed by (6.17) is thus minimised by maximising $\log P\left(h_T \mid \eta\right)$, that is choosing the model η under which the probability of the given observation sequence h_T is maximised

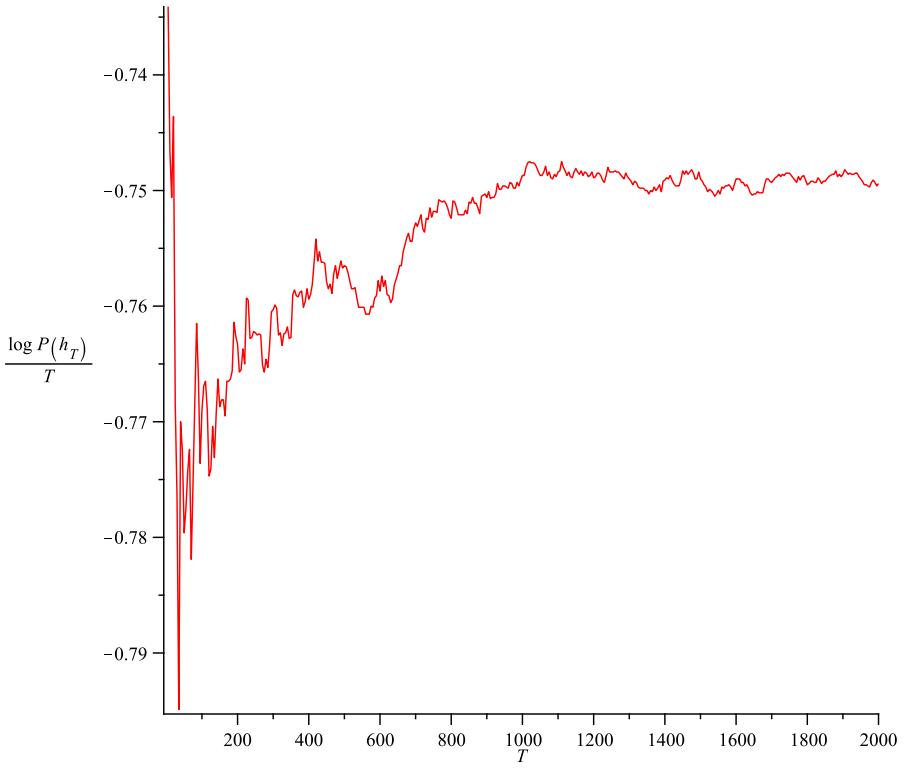


FIGURE 6.5: Convergence of $\frac{1}{T}\log P\left(h_T \mid \eta\right)$ to the limit $H(\lambda,\eta)$.

(See Criterion (6.1)).

Therefore, the reliability of the HMM-based trust model depends on the 'goodness' of approximating the limit $H\left(\lambda,\eta\right)$ by $\frac{1}{T}\log P\left(h_T \mid \eta\right)$ for any model η. In particular, if $\frac{1}{T}\log P\left(h_T \mid \eta\right)$ is not an appropriate approximation of $H\left(\lambda,\eta\right)$ (i.e. satisfying an arbitrary error margin), then Criterion (6.1) does not necessarily maximises $H\left(\lambda,\eta\right)$, and therefore, by (6.17), the estimation error is not necessarily minimised as required. In the following we describe a means of measuring the goodness of approximating the limit $H\left(\lambda,\eta\right)$ using the notion of the *mean squared error (MSE)*. In terms of this measure we will seek an approach to improving the approximation of $H\left(\lambda,\eta\right)$.

In the context of interacting principals, let λ be the (real) relative behaviour HMM of a trustee *te*, with respect to a truster *tr*. Let also η be an arbitrary approximate behaviour HMM for *te*. In a period of time of arbitrary length, the truster *tr* interacts with *te* and hence observes a sequence of outcomes h_T, where T denotes the (random) length of h_T. Here we define the random variable $G(h_T \mid \eta)$ as follows.

$$G(h_T \mid \eta) = \frac{1}{T}\log P\left(h_T \mid \eta\right)$$

The random variable $G(h_T \mid \eta)$ is simply a function of the random sequence h_T, where this function depends on the choice of the HMM η. The probability density function (pdf) of $G(h_T \mid \eta)$ is determined, therefore, by the real behaviour HMM λ which defines the probability distribution of the random sequence h_T.

By the almost sure convergence of $G(h_T \mid \eta)$ to $H(\lambda, \eta)$ in (6.18), the random variable $G(h_T \mid \eta)$ is thus seen as an estimator for $H(\lambda, \eta)$. The 'goodness' of this estimation (approximation) can be measured by the *mean squared error (MSE)* which is defined as the expected squared difference between the estimator and the parameter being estimated (See e.g. Bickel and Doksum, 2000). The MSE of $G(h_T \mid \eta)$ with respect to the limit $H(\lambda, \eta)$ is therefore expressed as follows.

$$MSE(G(h_T \mid \eta)) = \mathbf{E}\left[(G(h_T \mid \eta) - H(\lambda, \eta))^2\right] \tag{6.24}$$

In practise, the MSE of $G(h_T \mid \eta)$ highly depends on the length τ of the time period in which the sequence h_T is observed. Namely, the longer is the time period τ, the larger is the expected length T of the observation sequence h_T which results in a lower MSE, by the convergence property (6.18).

In the probabilistic setting of interaction between principals, it is likely (with a non zero probability) that the available sequence h_T of observations resulting from the personal interactions between the truster and the trustee is not sufficiently long to make $\frac{1}{T} \log P(h_T \mid \eta)$ an acceptable approximation for $H(\lambda, \eta)$. In this situation, another expression (estimator) which uses extra information, is therefore needed to better approximate $H(\lambda, \eta)$. The following section is devoted to describing this expression, which approximates $H(\lambda, \eta)$ using multiple observation sequences instead of one sequence.

6.4.1 Enhancing the convergence using multiple observation sequences

In the model for interacting principals, described in Section 6.1.1, it is assumed that each principal i has, intrinsically, a fixed probability p_i of being the te's partner in a given interaction. Informally, this probability indicates the tendency of peer i to interact with the trustee te. With respect to a trustee te, we call the members of a set of principals *symmetric peers* if they have the same tendency to interact with te.

Definition 6.3. The members of a set S of principals are said to be *symmetric peers* with respect to a trustee te if they have the same probability of being the te's partner in a given interaction. That is,

$$p_i = p_j \qquad \forall i, j \in S.$$

By Theorem 6.2, the symmetric peers observe the same relative behaviour of te, mod-

elled by a HMM λ. Thus they are expected to give consistent 'opinions' regarding the behaviour of *te*, which help better approximate the *te* relative behaviour λ.

Consider a set of symmetric peers $i = 1, 2, \ldots, M$, interacting with the trustee *te* randomly in an arbitrary period of time. Let h^i be the sequence of outcomes observed by principal i, where h^i has the length T_i. Accordingly, define the random variable $G^i(h_i \mid \eta)$ as follows.

$$G^i(h^i \mid \eta) = \frac{1}{T_i} \log P(h^i \mid \eta)$$

where η is an arbitrary HMM. In the above definition each random variable $G^i(h^i \mid \eta)$ is a function of the random T_i-length observation sequence observed by the peer i when interacting with the trustee *te*. It follows from the symmetry between principals $1, 2, \ldots, M$ that the variables $G^i(h^i \mid \eta)$ have the same probability distribution. Therefore, each of them can be seen as an approximation for $H(\lambda, \eta)$. This fact suggests taking the average of $G^i(h^i \mid \eta)$ as a better estimator for $H(\lambda, \eta)$. This improvement is expressed in terms of the mean squared error as confirmed by the following theorem.

Theorem 6.4. *Consider a set $S = \{1, 2, \ldots, M\}$ of symmetric principals with respect to a given trustee te, where each principal i observes the sequence h^i of length T_i during a fixed period of time. Let the average of the random variables $G^i(h^i \mid \eta)$ be defined as*

$$\bar{G}(S \mid \eta) = \frac{1}{M} \sum_{i=1}^{M} G^i(h^i \mid \eta). \tag{6.25}$$

Then, it holds for all $i \in S$ that

$$MSE(\bar{G}(S \mid \eta)) \leq MSE\left(G^i(h^i \mid \eta)\right),$$

where the equality holds if and only if

$$P\left(G^i(h^i \mid \eta) = G^j(h^j \mid \eta)\right) = 1 \qquad \forall i, j \in S. \tag{6.26}$$

Proof. In the following we write G^i, $\bar{G}$, and H as shorthands for $G^i(h^i \mid \eta)$, $\bar{G}(S \mid \eta)$, and $H(\lambda, \eta)$ respectively. By the symmetry of principals $\{i : i \in S\}$, the random variables $\{G^i : i \in S\}$ individually have the same probability distribution. Therefore, $MSE(G^i)$ is independent of i. That is

$$MSE(G^i) = MSE(G^j) \qquad \forall i, j \in S. \tag{6.27}$$

Using the above equation and the definition of $\bar{G}$ in (6.25), the MSE of the average $\bar{G}$

with respect to the limit H can be expanded as follows.

$$MSE(\bar{G}) = \mathbf{E}\left[(\bar{G} - H)^2\right] = \frac{1}{M^2}\,\mathbf{E}\left[\left(\sum_{i=1}^{M}(G^i - H)\right)^2\right]$$
$$= \frac{1}{M^2}\left(M\,MSE(G^i)\right) + \frac{2}{M^2}\sum_{u=1}^{M}\sum_{v>u}^{M}\mathbf{E}\left[(G^u - H)\,(G^v - H)\right]. \tag{6.28}$$

Applying the *triangle inequality*, we get

$$MSE(\bar{G}) \leq \frac{1}{M^2}\left(M\,MSE(G^i)\right) + \frac{2}{M^2}\sum_{u=1}^{M}\sum_{v>u}^{M}\left|\mathbf{E}\left[(G^u - H)\,(G^v - H)\right]\right|. \tag{6.29}$$

By the *Cauchy–Schwarz inequality* (see e.g. Theorem (9) in Grimmet and Stirzaker, 2001), it holds that

$$\left|\mathbf{E}\left[(G^u - H)\,(G^v - H)\right]\right| \leq \sqrt{\mathbf{E}\left[(G^u - H)^2\right]}\sqrt{\mathbf{E}\left[(G^v - H)^2\right]}, \tag{6.30}$$

where the equality holds if and only if

$$P\left(\alpha\,(G^u - H) = \beta\,(G^v - H)\right) = 1,$$

for some real α, β at least one of them is non-zero. Taking into account that $(G^u - H)$ and $(G^v - H)$ have the same mean, the above condition can be written equivalently as follows.

$$P\left(G^u = G^v\right) = 1 \tag{6.31}$$

Using Eq. (6.27), Inequality (6.30) can be written as follows for any $u, v, i \in S$.

$$\left|\mathbf{E}\left[(G^u - H)\,(G^v - H)\right]\right| \leq MSE(G^i) \tag{6.32}$$

The proof is then completed by substituting Inequality (6.32) in (6.29). □

According to Theorem 6.4, the average $\bar{G}(S \mid \eta)$ for a set S of symmetric principals is a better approximation for the limit $H(\lambda, \eta)$ than $G^i_T(\lambda, \eta)$ in terms of the mean squared error. This provides a formal basis of enhancing the approximation of $H(\lambda, \eta)$ using multiple sequences by choosing the approximate behaviour HMM η which maximises the average $\bar{G}(S \mid \eta)$ rather than maximising the probability of a single sequence (i.e. maximising $G^i(h^i \mid \eta)$). Using this result for a set of symmetric peers, the criterion (6.1) of choosing the approximate model η can be now generalised to use multiple sequences

rather than one sequence as follows.

$$\begin{aligned}\eta &= \operatorname*{argmax}_{\mathcal{R}_n} \bar{G}(S \mid \eta) \\ &= \operatorname*{argmax}_{\mathcal{R}_n} \sum_i \frac{1}{T_i} \log P\left(h^i \mid \mathcal{R}_n\right)\end{aligned} \tag{6.33}$$

Observe that the mean squared error of $\bar{G}(S \mid \eta)$ is not improved by using multiple sequences if and only if the condition (6.26) holds. In fact this condition holds in the following cases.

1. The observation sequences h^i are sufficiently long so that $G^i(h^i \mid \eta) = H(\lambda, \eta)$ (with respect to an arbitrary small error margin). The averaging does not provide a significant advantage in this case as $G^i(h^i \mid \eta)$, for any i, is already equal to $H(\lambda, \eta)$. Note that this case happens when the time period, during which observation sequences are observed, is large enough to allow for such long sequences.

2. The HMM η is defined such that at each state, the probability distribution over outcomes (the emission probability distribution) is uniform. In this case it holds, for any sequence $h = o_1 o_2 \dots o_T$, that

$$\begin{aligned}\log P(o_1 \mid \eta) = \log P(o_2 \mid o_1, \eta) = \log P(o_3 \mid o_1\, o_2, \eta) = \dots \\ = \log P(o_T \mid o_1\, o_2 \dots o_{T-1}, \eta) = H(\lambda, \eta),\end{aligned}$$

   which implies that

$$\frac{1}{T} \log P(h \mid \eta) = H(\lambda, \eta).$$

   That is the convergence to the limit $H(\lambda, \eta)$ is already achieved by any in-hand sequence h^i, and similarly to the previous case there is no need for averaging to approximate $H(\lambda, \eta)$.

6.5 Discussion

This chapter has introduced the foundations for the HMM-based trust model. This model is based on approximating the behaviour of any trustee by the n-states HMM η which maximises the likelihood of the available history of observations. The approximate behaviour model η is then used to evaluate the estimated predictive probability distribution given any sequence of observations. With modelling the real dynamic behaviour of principals by hidden Markov models, and using the results obtained by Baum and Petrie (1966), we justified the consistency of the HMM-based trust model. This justification relies on showing that maximising the likelihood of a given observation sequence minimises the relative entropy between the real and estimated predictive probability distributions.

To assess the estimation quality of a particular trust algorithm, we use the notion of expected estimation error that is the expected difference (divergence) between the real and estimated predictive probability distributions. Since we have no means yet to evaluate the expected estimation error expressed by Equation (6.17) for the HMM-based trust model using analytical or numerical methods, a Monte-Carlo algorithm, described in Section 6.3.1, has been used for evaluating the expected estimation error.

Using an implementation of this algorithm, and adopting the relative entropy (Kullback-Leibler divergence) as a measure for the estimation error, an experimental comparison between HMM-based trust algorithm and the Beta-based trust algorithm with an exponential decay scheme was performed. The results of this comparison are given in Section 6.3.2. These results shows that HMM-based trust algorithm gives a better estimation for the predictive probability distribution when the trustee's behaviour is highly dynamic. When the real behaviour is more stable (less dynamic), the Beta-based algorithm with the optimal value of decay gives slightly better estimation than the HMM-based algorithm.

The HMM-based trust model relies on observing a sequence of interaction outcomes rather than individual (and independent) outcomes. The order of the outcomes in a single sequence, in fact, provides valuable statistical information about the trustee's underlying evolution of states, and hence helps better estimate the predictive probability distribution. This 'sequencing' information is basically the reason why the HMM-based trust model incurs a lower estimation error than the Beta trust model when the the trustee's behaviour is dynamic.

Nevertheless, the HMM-based estimation using a single observation sequence requires that this sequence be sufficiently long as explained in Section 6.4. In practical cases where such a long observation sequence is unavailable, Section 6.4.1 describes an approach to reliable HMM-based estimation using multiple sequences. This approach compensates the lack of a single long sequence by information from multiple sequences. This results in a general criterion for choosing the approximate HMM model given multiple sequences. This criterion is expressed by (6.33) and plays the basic role of extending the HMM-trust model to encompass a model of reputation which is detailed in the following chapter.

Chapter 7

HMM-based reputation model

In Chapter 6, we described the foundations of our proposed HMM-based trust model which forms the basic pillar in a probabilistic trust framework aimed at evaluating the trust in a principal (trustee) exhibiting dynamic behaviour. Underlying this framework, the dynamic behaviour of a trustee *te* is represented by a set of (behaviour) states where each state is associated with a probability distribution over possible interaction outcomes. The dynamic behaviour of *te* is thus reflected by probabilistic transitions from one state to another.

As described earlier in Section 6.1.1, two descriptive behaviours of the trustee *te* are distinguished. The first is called the *general behaviour* of *te* in which the probabilistic state transition is modelled by a finite-state Markov chain (MC). This MC describes the probabilistic evolution of *te*'s internal states from one interaction to a successive one irrespective of its partners. Since each state is associated with a probability distribution over potential interaction outcomes, the general behaviour of *te* is then modelled by a finite-state HMM, called the *general behaviour HMM*. This HMM defines a probability distribution over the sequences of outcomes of interactions involving *te* regardless of the *te* partners. This representation involves an implicit assumption that the trustee does not discriminate between different interaction partners, and the probability distribution over interaction outcomes depends only on the trustee's internal state which depends on e.g. the security and integrity properties of the trustee as shown in Example 6.1.

On the other hand, a truster *tr* is usually interested in the outcomes of its 'personal' interactions with *te*, rather than all interactions of *te*. That is *te* defines its trust in *te* as an estimated probability distribution over the potential outcomes of *tr*'s next interaction with *te*. This requires modelling the so called *relative behaviour* of *te* with respect to *tr*. Unlike the general behaviour of *te* which considers all interactions involving *te*, the relative behaviour (with respect to *tr*) considers only the interactions between *te* and *tr*. Specifically, a model for the relative behaviour of *te* with respect to *tr* is required to define a probability distribution over sequences of outcomes of interactions between *te*

and tr. To meet this requirement, it is assumed that, at a given interaction involving te, the partner of te in this interaction is tr with a fixed probability p_{tr}. Under this assumption, Theorem 6.2 proves that the relative behaviour of te can be represented by a finite-states HMM, called the *relative behaviour HMM* and denoted by λ, whose parameters depend on te's general behaviour HMM, and also on the probability p_{tr} associated with tr.

In order to evaluate the trust of tr in the trustee te, we seek approximating the (hidden) relative behaviour HMM of te with respect to tr. While the parameters of te's relative behaviour HMM are hidden from tr, the sequence h of outcomes of interactions between tr and te is observable to tr. The objective of the HMM-based trust model is therefore to provide the truster tr a means for evaluating its trust in te using the given sequence of observations h, that is to estimate the probability distribution over possible outcomes of the next interaction with te. We also call this distribution an *estimated predictive probability distribution*.

The HMM-based trust model employs the Baum-Welch algorithm to approximate the relative behaviour HMM λ of the trustee by an arbitrary size finite-state HMM η, called the approximate behaviour HMM. The estimated predictive probability is then evaluated using Eq. (6.2). Following the representation of the (hidden) relative behaviour of te by a HMM λ, it was shown that the trust evaluation algorithm is consistent in the sense that the asymptotic error of estimating the predictive probability distribution is minimised by choosing the approximate HMM model η which maximises the probability of the historical sequence of observations h. This process of choosing the approximate model which fits the historical sequence, is referred to as *learning* the trustee's behaviour.

In many practical situations, the sequence of personal observations available to the truster is not sufficiently long to learn the behaviour of a trustee. In these cases, *learning* the behaviour of the considered trustee using such a short sequence would not be reliable for obtaining an approximate model η for the trustee. A traditional approach to handle this shortage of information is to incorporate the *reputation information* in the behaviour learning process. Reputation information can be simply described as the feedback information collected from other principals about the trustee. This feedback information is aimed to enrich the available information about the trustee's behaviour to the truster, and therefore enables the truster to more reliably approximate the trustee's behaviour and accordingly better estimate the predictive probability distribution over possible outcomes of the next interaction with the trustee. Addressing the problem of reputation requires answering two main questions:

- In which format should a principal, called a *reputation source*, phrase its reputation report about a particular trustee ?
- How can a truster use the reports collected from different reputation sources, to enhance the process of learning the trustee's behaviour ?

Clearly, the answer to these questions depends on the assumptions made about the behaviour model of a principal. For example many existing frameworks are based on the assumption that a trustee's behaviour is modelled by a fixed probability distribution over the interaction outcomes (observables). This is apparent in the Beta Reputation System (Jøsang and Ismail, 2002) which is followed in TRAVOS (Teacy et al., 2006), where the assumed fixed distribution considers only two possible outcomes (success, fail). This assumption of a fixed distribution over interaction outcomes is also adopted by Nielsen et al. (2007) and Jøsang and Haller (2007), where multiple outcomes are considered instead of only two. According to this assumption of fixed distribution, a single reputation report (feedback) given by a reputation source includes the count of each outcome experienced by the source out of its interaction with the trustee (See (3.2)). Mixing multiple reputation reports is performed in these systems by adding the counts of corresponding outcomes in the given reports (See Eqs. (3.3)).

This solution to the reputation problem is consistent under the assumption of the fixed behaviour (probability distribution) of the trustee, since the outcomes of interactions are independent of each other, and therefore only the counts of occurrences of each outcome (e.g. success or failure) are important to estimate this distribution.

Since our novel HMM-based trust model is based on representing the dynamic behaviour of principals by a HMM rather than a fixed probability distribution over observables, we ask for reputation information which reflects the 'dynamicity' of the trustee's behaviour. A reputation report is therefore required to provides information about individual trustee's states. This is not trivial because the internal state transitions of the trustee is hidden from the reputation source when it interacts with the trustee. However, using the Baum-Welch algorithm, the truster can estimate some statistics about such hidden states and their associated probability distributions over observables. Following this idea we seek, in this chapter, a framework which answers the above two questions, and therefore augments the basic HMM-based trust model by a reputation handling mechanism.

7.1 General assumptions

The HMM-based trust model is based on representing the relative behaviour of any trustee *te* by a stationary finite-state HMM λ. This representation involves the assumption that a trustee has always a finite number of states, where each state is described by a probability distribution (the emission distribution) over possible outcomes, and also an immutable probability distribution (the state transition distribution) governing the transition from the given state to other states.

In order for a set of reputation sources to give consistent opinions about the behaviour of a trustee *te* behaviour, we assume that *te* interacts similarly with these sources. That is

te has the same relative behaviour HMM λ with respect to the reputation sources. This allows the principal mixing the reputation reports to better approximate the 'unique' hidden behaviour λ of *te* rather than concluding a model η which averages different hidden behaviours. In the setting of general and relative behaviour models described in Section 6.1.1, the relative behaviour of *te* with respect to a reputation source *rs* depends on the probability p_{rs} at which the reputation source *rs* interacts with *te*. Thus according to this setting, the proposed reputation model requires collecting reputation from principals which interact with the trustee at the same probability p.

As shown in Chapter 6, the HMM-based trust evaluation is based on approximating the relative behaviour of any trustee by a finite-state HMM η with an arbitrary number of states. For the sake of unifying the form of reputation reports, it is assumed that reputation sources agree on the structure of the approximate behaviour model η for a trustee; namely the set of states S and the set of possible interaction outcomes V. Each single reputation report can therefore have the same representation, whose semantics depend on the agreed structure of η. The objective of the reputation mixing algorithm is then to determine the parameter values of η from the given reputation reports.

Example 7.1. *A set of peers can confine the approximate HMM of any trustee te to the states 'Honest(H)' and 'Corrupt(C)', that is* $S = \{H, C\}$. *The outcome of any interaction with te is also confined to the set* $V = \{\mathsf{s}, \mathsf{f}\}$, *which indicates successful and unsuccessful interactions respectively. At state H, the trustee interacts successfully with higher probability than interacting unsuccessfully, while vice versa in the state C.*

7.2 Reputation framework

In the HMM based trust framework, consider a truster *tr* aiming at evaluating its trust in a trustee *te*. Let h be the sequence of outcomes of interaction between *tr* and *te*, observed by *tr*. The truster *tr* therefore tries to approximate the behaviour of *te*, by estimating the parameters of the optimal HMM model η which maximises the probability of the observation sequence h seen only by *tr*, as indicated by Eq. (6.1). This parameter estimation process is performed by applying the Baum-Welch algorithm which is described in Section 2.5.4.

As shown by Section 2.5.4 and Rabiner (1989), the Baum-Welch algorithm is basically an instance of the *Expectation-Maximization* algorithm where maximising the data likelihood amounts to iteratively maximising the *expected complete data likelihood*. In each iteration of the EM algorithm, an *a priori* model is used to evaluate the expected complete data likelihood function, which is then optimised to obtain an *a posteriori* model. This process is then repeated in the next iteration with replacing the a priori model by the obtained a posteriori one. More details about the EM algorithm can be found in Section 2.2.3 and Dempster et al. (1977).

In the case of HMM parameter estimation, the expected complete data likelihood, also called the *Baum's auxiliary function*, is given by Eq. (2.11), which we rewrite in the following form

$$\mathcal{Q}\left(\eta', h, \eta\right) = \sum_{q} P\left(q \mid h, \eta'\right) \log P\left(h, q \mid \eta\right), \tag{7.1}$$

where η' and η are respectively the a priori and the a posteriori models. h and q are respectively the observed sequence of outcomes and the corresponding (hidden) sequence of states. More details about maximising the above function are given by Section 2.5.4, and Rabiner and Juang (1993).

To handle the reputation problem, consider multiple principals $1, 2, \ldots, M$ which interact with a trustee *te*. If these principals interact with *te* with the same probability, the relative behaviour *te* with respect to each of them is the same according to Theorem 6.2. However, each principal u can obtain a different approximate model η^u for *te* which maximises the probability of its own observation sequence h^u.

Rather than having multiple approximate models which individually maximise different sequences, it is required to obtain a single model which tries to maximise the 'overall' likelihood of the observation sequences h^u. For this purpose, Criterion (6.33) is proposed for choosing the optimal approximate model. That is we aim at finding the approximate model η^* which maximises a *likelihood objective function* $\mathcal{G}\left(h^1, h^2, \ldots, h^M \mid \eta\right)$ defined as

$$\mathcal{G}\left(h^1, h^2, \ldots, h^M \mid \eta\right) = \sum_{u} \frac{1}{T_u} \log P\left(h^u \mid \eta\right), \tag{7.2}$$

where T_u is the length of the sequence h^u. Although the sequences h^u are not independent, maximising the above objective function tends to give better estimation results as shown in Section 6.4.1. If all such sequences are available to one principal u, it would be able to estimate η^* iteratively using the EM algorithm. Indeed, it is not practical for principals to exchange their whole observation sequences as reputation information since each of these sequences is getting longer over time. So, we seek in the following an alternative approach to estimate η^* associated with the trustee *te*, using partial and bounded information about the observation sequences.

Consider a single sequence h of observations. The log probability of h given a model η is related to the auxiliary function, (7.1), by the following lemma.

Lemma 7.1. *Given an observation sequence h, and a model η',*

$$\log P\left(h \mid \eta\right) \geq \mathcal{Q}\left(\eta', h, \eta\right) + \mathcal{R}\left(\eta', h\right),$$

where

$$\mathcal{R}\left(\eta', h\right) = -\sum_{q} P\left(q \mid h, \eta'\right) \log P\left(q \mid h, \eta'\right).$$

Proof.

$$\begin{aligned}
\log P(h \mid \eta) &= \log \left\{ \sum_q P(h, q \mid \eta) \right\} \\
&= \log \left\{ \sum_q P(q \mid h, \eta') \frac{P(h, q \mid \eta)}{P(q \mid h, \eta')} \right\} \\
&= \log \left\{ \mathbf{E}_q \left[\frac{P(q, h \mid \eta)}{P(q \mid h, \eta')} \,\middle|\, h, \eta' \right] \right\} \\
&\geq^{(1)} \mathbf{E}_q \left[\log \left(\frac{P(q, h \mid \eta)}{P(q \mid h, \eta')} \right) \,\middle|\, h, \eta' \right] \\
&= \sum_q P(q \mid h, \eta') \log \frac{P(q, h \mid \eta)}{P(q \mid h, \eta')} \\
&= \sum_q P(q \mid h, \eta') \log P(q, h \mid \eta) - \sum_q P(q \mid h, \eta') \log P(q \mid h, \eta') \\
&= \mathcal{Q}(\eta', h, \eta) + \mathcal{R}(\eta', h) .
\end{aligned}$$

The inequality (1) is obtained by applying Jensen's inequality (see e.g. Theorem 2.6.2 in Cover and Thomas, 2006) to the *log* function. The equality holds when $\eta = \eta'$. □

The above lemma expresses a lower bound for the log probability of h under any model η, in terms of an a priori model η'. From the above proof, if $\eta' \neq \eta$ the exact difference between this bound and the log probability of h depends clearly on the observation sequence h, and the choice of η', η. If full information about h is not available, the best that can be done for maximising $\log P(h \mid \eta)$ is to maximise its lower bound with respect to η. This amounts to maximising $\mathcal{Q}(\eta', h, \eta)$ since $\mathcal{R}(\eta', h)$ is independent of the variable model η. While the auxiliary function $\mathcal{Q}(\eta', h, \eta)$ still depends on the observation sequence h, it will be shown that only partial, and also bounded information about h is needed to maximise $\mathcal{Q}(\eta', h, \eta)$.

7.2.1 Deriving a reputation algorithm

For any particular trustee, each reputation source u observes an observation sequence $h^u = o_1^u o_2^u \ldots o_{T_u}^u$ resulting from its 'personal' interactions with the trustee. A reputation source therefore maintains an approximate behaviour model η^u for the trustee's behaviour. For maximising the likelihood objective function $\mathcal{G}(h^1, h^2, \ldots, h^M \mid \eta)$ defined by (7.2), it is necessary to have full information about the sequences as shown above. Since any particular principal has no access to the sequences observed by other principals, we aim at finding the model η^* which rather maximises a lower bound of

$\mathcal{G}\left(h^1, h^2, \ldots, h^M \mid \eta\right)$. Using Lemma 7.1, this lower bound is given by

$$\sum_u \frac{1}{T_u} \mathcal{Q}\left(\eta^u, h^u, \eta\right) + \sum_u \frac{1}{T_u} \mathcal{R}\left(\eta^u, h^u\right).$$

Regarding the approximate models η^u as a priori models for the trustee's behaviour, the problem of reputation mixing can be expressed as finding the *optimal a posteriori HMM* η^* which maximises the above lower bound of the likelihood objective function. Since $\sum_u \frac{1}{T_u} \mathcal{R}\left(\eta^u, h^u\right)$ is independent of the a posteriori model η, we write

$$\eta^* = \underset{\eta}{\operatorname{argmax}} \sum_u \frac{1}{T_u} \mathcal{Q}\left(\eta^u, h^u, \eta\right). \tag{7.3}$$

It will be shown in this section that full information about the sequences h_u is not necessary to maximise the above expression, while partial information about these sequences are sufficient to perform the maximisation in (7.3). The derivation for the necessary information is inspired by the derivation of the Baum-Welch reestimation equations given in (Rabiner and Juang, 1993).

Let $q^u = q_1^u, q_2^u, \ldots, q_{T_u}^u$ denote the (hidden) sequence of states underlying the observation sequence h^u. For performing the maximisation in 7.3, we need to formulate the function $\mathcal{Q}\left(\eta^u, h^u, \eta\right)$ using its definition in (7.1). We start by formulating $\log P\left(h^u, q^u \mid \eta\right)$ in terms of the parameters of η as follows

$$\log P\left(h^u, q^u \mid \eta\right) = \log \pi_{q_1^u} + \sum_{t=2}^{T_u} \log A_{q_{t-1}^u q_t^u} + \sum_{t=1}^{T_u} \log B_{q_t^u}\left(o_t^u\right), \tag{7.4}$$

where π_i denotes the probability that the initial state (q_1^u) is i. A_{ij} is the probability of transition from state i to state j. $B_i\left(z_k\right)$ is the probability of observing the outcome z_k at state i. Refer to the description of the HMM elements in Section 2.5.1 for more details about these notations.

Substituting Expression (7.4) in (7.1), the function $\mathcal{Q}\left(\eta^u, h^u, \eta\right)$ can be written as follows.

$$\begin{aligned}\mathcal{Q}\left(\eta^u, h^u, \eta\right) = & \sum_{i=1}^{N} P\left(q_1^u = i \mid h^u, \eta^u\right) \log \pi_i + \\ & \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T_u} P\left(q_{t-1}^u = i, q_t^u = j \mid h^u, \eta^u\right) \log A_{ij} + \\ & \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{t=1}^{T_u} P\left(q_t^u = i \mid h^u, \eta^u\right) \delta\left(o_t^u, z_k\right) \log B_i\left(z_k\right),\end{aligned} \tag{7.5}$$

where N is the number of states, K is the number of possible observation symbols, and

the δ-function $\delta\left(o_t^u, z_k\right)$ is defined as by:

$$\delta\left(o_t, z_k\right) = \begin{cases} 1 & \text{if } o_t = z_k; \\ 0 & \text{otherwise.} \end{cases} \tag{7.6}$$

Now we are ready to expressing the sum $\sum_{u=1}^{M} \frac{1}{T_u} \mathcal{Q}\left(\eta^u, h^u, \eta\right)$ in (7.3) by scaling Eq. (7.5) by $\frac{1}{T_u}$, and then summing over the available reputation sources $\{1, 2, \ldots, M\}$. For convenience, we write the resulting sum as follows

$$\sum_{u=1}^{M} \frac{1}{T_u} \mathcal{Q}\left(\eta^u, h^u, \eta\right) = Q_\pi\left(\boldsymbol{\pi}\right) + \sum_{i=1}^{N} Q_{A_i}\left(\boldsymbol{A_i}\right) + \sum_{i=1}^{N} Q_{B_i}\left(\boldsymbol{B_i}\right), \tag{7.7}$$

where $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_N]$ is the vector representing the initial state probability distribution, $\boldsymbol{A_i} = [A_{i1}, A_{i2}, \ldots, A_{iN}]$ is the vector representing the probability distribution over state transitions from state i to other states, and $\boldsymbol{B_i} = [B_i(z_1), B_i(z_2), \ldots, B_i(z_K)]$ is the vector representing the emission probability distribution over outcomes given state i. The functions $Q_\pi(\boldsymbol{\pi})$, $Q_{A_i}(\boldsymbol{A_i})$, and $Q_{B_i}(\boldsymbol{B_i})$ in the above equation are defined as follows

$$Q_\pi(\boldsymbol{\pi}) = \sum_{i=1}^{N} \left(\sum_{u=1}^{M} \frac{1}{T_u} P\left(q_1^u = i \mid h^u, \eta^u\right) \right) \log \pi_i, \tag{7.8}$$

$$Q_{A_i}(\boldsymbol{A_i}) = \sum_{j=1}^{N} \left(\sum_{u=1}^{M} \frac{1}{T_u} \sum_{t=2}^{T_u} P\left(q_{t-1}^u = i, q_t^u = j \mid h^u, \eta^u\right) \right) \log A_{ij}, \tag{7.9}$$

$$Q_{B_i}\left(\boldsymbol{B_i}\right) = \sum_{k=1}^{K} \left(\sum_{u=1}^{M} \frac{1}{T_u} \sum_{t=1}^{T_u} P\left(q_t^u = i \mid h^u, \eta^u\right) \delta\left(o_t^u, z_k\right) \right) \log B_i\left(z_k\right). \tag{7.10}$$

Observe that each term in Equation (7.7) is a function of a probability distribution which parametrises the HMM η. These distributions ($\boldsymbol{\pi}$, $\boldsymbol{A_i}$, $\boldsymbol{B_i}$ $\forall i : 1 \leq i \leq N$) are independent of each other, that is the choice of one of them does not affect the choice of the others. Therefore the overall sum (7.7) is maximised by maximising each term in (7.7) separately. Observe furthermore that each of equations (7.8),(7.9), and (7.10) is in the following form

$$F\left(y_1, y_2, \ldots, y_V\right) = \sum_{v=1}^{V} w_v \log y_v \qquad \text{where} \qquad \sum_{v=1}^{V} y_v = 1. \tag{7.11}$$

Using the *Lagrange multiplier* technique for optimising a function subject to a constraint, the constrained function F defined above can be easily proved to have a global maximum at the point $(\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_V)$, where $\bar{y}_v$ is given by

$$\bar{y}_v = \frac{w_v}{\sum_{v=1}^{V} w_v}.$$

Using the above fact, the parameters of the optimal a posteriori model η^* are given as follows

$$\bar{\pi}_i = \frac{\sum_{u=1}^{M} \frac{1}{T_u} P\left(q_1^u = i \mid h^u, \eta^u\right)}{\sum_{u=1}^{M} \frac{1}{T_u}}, \tag{7.12}$$

$$\bar{A}_{ij} = \frac{\sum_{u=1}^{M} \frac{1}{T_u} \sum_{t=2}^{T_u} P\left(q_{t-1}^u = i, q_t^u = j \mid h^u, \eta^u\right)}{\sum_{u=1}^{M} \frac{1}{T_u} \sum_{t=2}^{T_u} P\left(q_{t-1}^u = i \mid h^u, \eta^u\right)}, \tag{7.13}$$

$$\bar{B}_i\left(z_k\right) = \frac{\sum_{u=1}^{M} \sum_{t=1}^{T_u} P\left(q_t^u = i \mid h^u, \eta^u\right) \delta\left(o_t^u, z_k\right)}{\sum_{u=1}^{M} \frac{1}{T_u} \sum_{t=1}^{T_u} P\left(q_t^u = i \mid h^u, \eta^u\right)}. \tag{7.14}$$

In the context of Baum-Welch algorithm description (Section 2.5.4, Rabiner, 1989; Rabiner and Juang, 1993), $P\left(q_t = i \mid h, \eta\right)$, the probability of visiting state i at time t given an observation sequence h and a HMM η is denoted by the variable $\gamma_t(i)$. Also $P\left(q_{t-1} = i, q_t = j \mid h, \eta\right)$, the probability of visiting states i and j at times $t-1$ and t respectively is denoted by the variable $\xi_{t-1}(i,j)$. In the same manner, we use the variables $\gamma_t^u(i)$ and $\xi_{t-1}^u(i,j)$ to denote the probabilities $P\left(q_t^u = i \mid h^u, \eta^u\right)$, and $P\left(q_{t-1}^u = i, q_t = j \mid h^u, \eta^u\right)$ respectively. Using these variables, Eqs (7.12),(7.13), and (7.14) expressing the parameters of η^* can be written as follows

$$\bar{\pi}_i = \frac{\sum_{u=1}^{M} \frac{1}{T_u} \gamma_1^u(i)}{\sum_{u=1}^{M} \frac{1}{T_u}}, \tag{7.15}$$

$$\bar{A}_{ij} = \frac{\sum_{u=1}^{M} \frac{1}{T_u} \sum_{t=2}^{T_u} \xi_{t-1}^u(i,j)}{\sum_{u=1}^{M} \frac{1}{T_u} \sum_{t=2}^{T_u} \gamma_{t-1}^u(i)}, \tag{7.16}$$

$$\bar{B}_i(z_k) = \frac{\sum_{u=1}^{M} \frac{1}{T_u} \sum_{t=1, o_t^u = z_k}^{T_u} \gamma_t^u(i)}{\sum_{u=1}^{M} \frac{1}{T_u} \sum_{t=1}^{T_u} \gamma_t^u(i)}. \tag{7.17}$$

The above derivation for Eqs. (7.15- 7.17) proves therefore the following theorem.

Theorem 7.2. *The parameters of the optimal a posteriori HMM η^*, defined by Eq. (7.3), are given by Eqs. (7.15- 7.17).*

From Eqs. (7.15- 7.17), it is noteworthy that the computation of η^* parameters does not require full specification of the observation sequences h^u, where $1 \leq u \leq M$. It requires however only some statistical functions of these sequences. Based on this fact, a concise form of a reputation report and also a reputation mixing scheme can be devised as shown below.

7.2.2 Reputation mixing

By Eqs. (7.15), (7.16), and (7.17) we are ready to formulate the novel HMM-based reputation framework. Starting by an initial HMM, each principal u applies the Baum-Welch learning algorithm to its own sequence of observations h^u to obtain an approximate HMM for the trustee. Referring to our assumptions in Section 7.1, the reputation sources must agree on the structure of the approximate HMM; that is the number of states N and the number of observables K. The Baum-Welch training performed by each principal u results in an approximate behaviour HMM η^u for the trustee and also the variables $\gamma_t^u(i)$, $\xi_t^u(i,j)$ for all $1 \leq t \leq T_u$, and all $i, j \in \{1, 2, \ldots, N\}, k \in \{1, 2, \ldots, K\}$. In terms of these variables, a principal u formulates its reputation report about the trustee as the tuple

$$\left(T_u\,,\,\bar{\boldsymbol{\gamma}}_1^u\,,\,\bar{\boldsymbol{\gamma}}_{T_u}^u\,,\,\bar{\boldsymbol{\gamma}}^u\,,\,\bar{\boldsymbol{\xi}}^u\,,\,\bar{\boldsymbol{\omega}}^u\right).$$

While T_u is clearly the length of h^u, each other element in the above reputation report is basically a matrix defined as follows

$$\bar{\boldsymbol{\gamma}}_1^u = \left[\begin{array}{cccc} \bar{\gamma}_1^u(1) & \bar{\gamma}_1^u(2) & \ldots & \bar{\gamma}_1^u(N) \end{array}\right] \qquad \text{where} \quad \bar{\gamma}_1^u(i) = \frac{1}{T_u}\,\gamma_1^u(i),$$

$$\bar{\boldsymbol{\gamma}}_{T_u}^u = \left[\begin{array}{cccc} \bar{\gamma}_{T_u}^u(1) & \bar{\gamma}_{T_u}^u(2) & \ldots & \bar{\gamma}_{T_u}^u(N) \end{array}\right] \qquad \text{where} \quad \bar{\gamma}_{T_u}^u(i) = \frac{1}{T_u}\,\gamma_{T_u}^u(i),$$

$$\bar{\boldsymbol{\gamma}}^u = \left[\begin{array}{cccc} \bar{\gamma}^u(1) & \bar{\gamma}^u(2) & \ldots & \bar{\gamma}^u(N) \end{array}\right] \qquad \text{where} \quad \bar{\gamma}^u(i) = \frac{1}{T_u}\sum_{t=1}^{T_u-1}\gamma_t^u(i),$$

$$\bar{\boldsymbol{\xi}}^u = \left[\begin{array}{cccc} \bar{\xi}^u(1,1) & \bar{\xi}^u(1,2) & \ldots & \bar{\xi}^u(1,N) \\ \bar{\xi}^u(2,1) & \bar{\xi}^u(2,2) & \ldots & \bar{\xi}^u(2,N) \\ \vdots & \ldots & \ddots & \vdots \\ \bar{\xi}^u(N,1) & \bar{\xi}^u(N,2) & \ldots & \bar{\xi}^u(N,N) \end{array}\right] \qquad \text{where} \quad \bar{\xi}^u(i,j) = \frac{1}{T_u}\sum_{t=2}^{T_u}\xi_{t-1}^u(i,j),$$

$$\bar{\boldsymbol{\omega}}^u = \left[\begin{array}{cccc} \bar{\omega}^u(1,1) & \bar{\omega}^u(1,2) & \ldots & \bar{\omega}^u(1,K) \\ \bar{\omega}^u(2,1) & \bar{\omega}^u(2,2) & \ldots & \bar{\omega}^u(2,K) \\ \vdots & \ldots & \ddots & \vdots \\ \bar{\omega}^u(N,1) & \bar{\omega}^u(N,2) & \ldots & \bar{\omega}^u(N,K) \end{array}\right] \qquad \text{where} \quad \bar{\omega}^u(i,k) = \frac{1}{T_u}\sum_{t=1,\,o_t^u=z_k}^{T_u}\gamma_t^u(i).$$

Now for describing the reputation mixing algorithm, consider the following set of reputation reports provided by M reputation sources.

$$\left\{\left(T_u\,,\,\bar{\boldsymbol{\gamma}}_1^u\,,\,\bar{\boldsymbol{\gamma}}_{T_u}^u\,,\,\bar{\boldsymbol{\gamma}}^u\,,\,\bar{\boldsymbol{\xi}}^u\,,\,\bar{\boldsymbol{\omega}}^u\right) \quad : \quad 1 \leq u \leq M\right\}$$

In terms of the elements of these reputation reports, Eqs. (7.15)-(7.17), evaluating the

parameters of the approximate behaviour HMM $\eta*$, can be rewritten as follows.

$$\bar{\pi}_i = \frac{\sum_{u=1}^{M} \bar{\gamma}_1^u(i)}{\sum_{u=1}^{M} \frac{1}{T_u}}, \tag{7.18}$$

$$\bar{A}_{ij} = \frac{\sum_{u=1}^{M} \bar{\xi}^u(i,j)}{\sum_{u=1}^{M} \bar{\gamma}^u(i)}, \tag{7.19}$$

$$\bar{B}_i(z_k) = \frac{\sum_{u=1}^{M} \bar{\omega}^u(i,k)}{\sum_{u=1}^{M} \left(\bar{\gamma}^u(i) + \bar{\gamma}_{T_u}^u(i)\right)}. \tag{7.20}$$

Using the above equations, a truster having a set of reputation reports (including its own report) can compute an optimal approximate HMM for the trustee. This approximate HMM is then used for evaluating the trust in the trustee using Eq. (6.2).

7.3 Performance analysis

It is important to highlight the additional computation cost required to extend the basic HMM-based trust model, described in Chapter 6, to incorporate reputation information. In other words, we ask the question what is the additional cost of using the 'opinions' of other peers along with the personal opinion of the truster. In fact, this additional cost includes the cost of computing, exchanging, and mixing the reputation reports.

The elements of a reputation report are evaluated by normalising the variables $\gamma^u(i)$, $\xi^u(i,j)$, and $\omega^u(i,k)$ by the length T_u of the observation sequence h^u. These variables are essential for the reputation source u to construct an approximate behaviour HMM η^u for the trustee, even if the trust evaluation is based only on personal interactions between u and the trustee. Thus the additional cost of formulating the reputation report is just paid for the normalisation by T_u. Note also that the cost of the reputation mixing process using Eqs. (7.18)-(7.20) is very insignificant compared to the cost of learning an approximate model for the trustee using the Baum-Welch algorithm.

Therefore the significant cost of extending the HMM-trust model to incorporate reputation, almost entirely lies in transporting the reputation reports between network principals. Note that these reputation reports have to be updated on a regular basis to incorporate new observations experienced by the reputation sources. Therefore the mechanism of exchanging reputation reports highly impacts the performance of the system. An independent line of research can be directed for optimising this cost of communicating the reputation reports.

7.4 Experimental evaluation

In this section we evaluate the performance of the HMM-based reputation model in terms of the expected estimation error. For this purpose, we adopt an experimental approach based on simulating the general behaviour of a trustee. In this simulation, the general behaviour of the trustee *te* is assumed to follow a given HMM. The other principals (peers) in the network interact with *te* probabilistically in a fixed period of time τ. For the purpose of simulation, it is assumed that interactions of *te* take place at an average rate (interactions per unit time). Thus the total number of interactions of *te* in a time interval τ tends to be proportional with τ. This allows for using the total number of *te*'s interactions instead of a continuous measure for the observation time period τ.

At any interaction involving *te*, the partner of *te* is principal i with a fixed probability (interaction probability) p_i. Assuming the trustee *te* experiences a number T of interactions with its network peers, each network peer i observes a sequence h^i of the outcomes of its interactions with *te*, where h^i is of length T_i. Each peer can therefore formulate its own reputation report for *te*. Symmetric peers with respect to *te* can exchange their reputation reports to approximate the *te*'s behaviour, and estimate their trust in *te* in the form of an estimated predictive probability distribution (over observables). The estimation error can be then evaluated by comparing the estimated predictive probability distribution to the real predictive distribution using Eq. (6.7).

Given the above simulation procedure, the expected estimation error can be obtained using the Monte-Carlo approach. Namely an m-size sample of estimation errors is obtained by running the above simulation procedure m times, where at each run the estimation error is evaluated. Making the sample size m arbitrarily large, the expected estimation error is then obtained as the sample average, according to the law of large numbers.

It is remarkable that this evaluation procedure is independent of the used trust and reputation models. In other words, the same procedure can be used for assessing different probabilistic observation-based reputation models. Taking the advantage of this feature we will compare between our adopted HMM-based reputation model and the traditional Beta reputation model (Jøsang and Ismail, 2002) in terms of the expected estimation error (6.19), where the estimation error, at each simulation run, is defined as the relative entropy divergence (given by Eq. (6.7)) from the real to the estimated predictive probability distributions.

7.4.1 Impact of multiple reputation reports

Here we consider a set of two symmetric peers with respect to a trustee *te*. This set includes a truster *tr* wanting to evaluate its trust in *te*, and a reputation source *rs*. Both

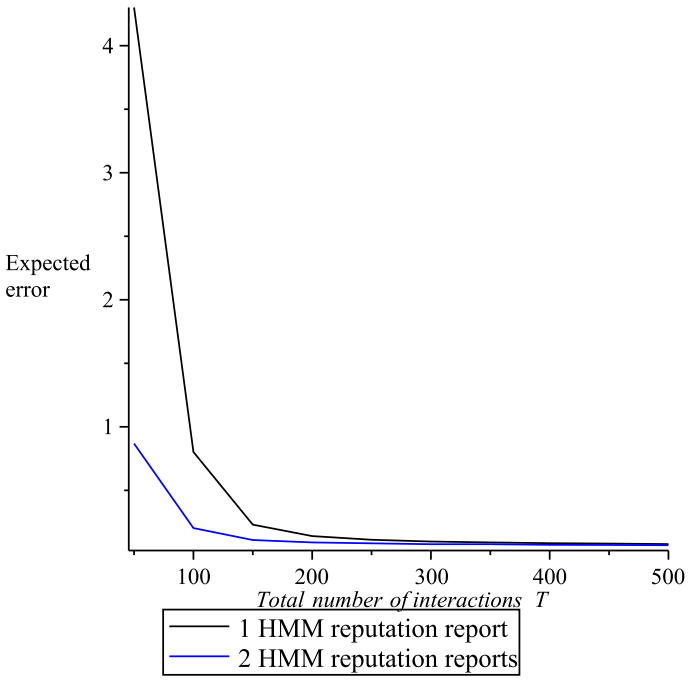


FIGURE 7.1: The expected estimation error using the HMM-based reputation model.

peers *tr* and *rs* formulate their reputation reports for *te*. Assuming that the truster *tr* is capable of receiving the *rs*'s reputation report, it is able to evaluate its trust in *te* either using its own single reputation report, or using the two available reports, by the HMM reputation mixing algorithm described in Section 7.2.2. In the following we compare these two methods in terms of the expected estimation error.

Let the general behaviour of *te* be modelled by the *s*-stable HMM defined in Section 5.5 with the stability parameter *s* set to 0.9. In this modelling, the outcomes of any interaction are confined to the binary set $\{\mathsf{s}, \mathsf{f}\}$. The interaction probability of *tr* (and also *rs*) is set to 0.2. This means that at any interaction with *te*, both the truster *tr*, and the reputation source *rs* are individually likely to be the *te*'s partner with probability 0.2, while it remains a probability of 0.6 that any other principal is the *te*'s partner.

With these settings, the interaction between *te* and other network peers is simulated as described earlier in Section 7.4. Figure 7.1, shows the impact of the described HMM-based reputation model on the expected estimation error using multiple reputation reports. One curve in this figure shows the estimation error resulting from using *tr*'s single reputation report. The other curve shows the expected estimation error when the truster *tr* uses its own reputation report along with the additional reputation report collected from the reputation source *rs*.

In both cases note that the expected estimation error is getting lower, as the number of interactions with *te* is getting higher. Indeed, this is because more observed interactions add more information about the relative behaviour of the trustee *te* with respect to

symmetric peers (*tr* abd *rs*). Therefore, learning the behaviour of the trustee, using this accumulated information tends to enhance estimating the predictive probability distributions, and hence reduces the estimation error.

Looking at the two curves in Figure 7.1, observe also that the proposed HMM reputation mixing provides a lower expected estimation error when multiple reputation reports are used. The improvement which results from mixing the two reputation reports is indicated by the vertical difference between the two curves. Observe that this improvement is getting less significant as the total number of interactions is getting larger. This is because when the number of *te*'s interactions is larger, the individual observation sequences observed by *tr* and *rs* tend to be consequently longer. This makes one sequence closer to be sufficient for learning the trustee's behaviour with no need to additional reputation reports.

7.4.2 Comparison with Beta reputation model

In the following, the simulation framework described earlier in Section 7.4 is used to contrast the HMM-based reputation model against the traditional Beta reputation model described by Section 3.2.1 and also by Jøsang and Ismail (2002). Consider a truster *tr* wanting to evaluate its trust in the trustee *te* using its own reputation report together with another reputation report collected from the reputation source *rs*. Again it is assumed that *tr* and *rs* are symmetric with respect to *te*, and have the same interaction probability 0.2. The general behaviour HMM for the trustee is again set to the same 0.9-stable HMM, used in Figure 7.1. With these settings, Figure 7.2 shows the expected estimation error when the HMM-reputation model is used, and when the Beta reputation model is used. From Figure 7.2, observe that for a relatively low number of total interactions T with *te*, the beta model outperforms the HMM reputation model by exhibiting a lower estimation error. In this case, the lengths of observation sequences (observed by *tr* and *rs*) are not sufficiently long to capture the 'dynamicity' of the behaviour. This makes learning the approximate behaviour HMM based on such relatively short sequences easily resulting in a HMM which is not a proper approximation for the trustee's relative behaviour, and hence a large estimation error compared to using Beta reputation reports.

However, for large number of interactions (T), the HMM reputation model exhibits a lower expected estimation error than the Beta model. In this case, the sequences of observations are long enough to reflect the dynamic behaviour of the trustee. Taking advantage of the sequencing information in these observed sequences, the HMM reputation model results in a reliable multiple-state HMM which approximates the hidden dynamic behaviour of the trustee. The Beta model, on the other hand, ignores the dynamic behaviour of the trustee and uses only the counts of observables to learn an 'average' probability distribution over possible outcomes as described in Section 3.2.1.

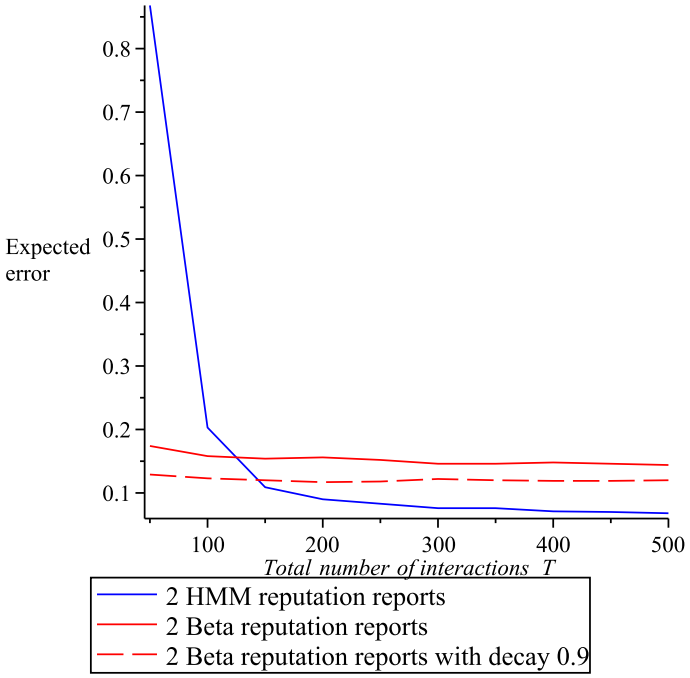


FIGURE 7.2: The expected estimation error using HMM and Beta reputation models.

It is also apparent in Figure 7.2, that incorporating the decay factor 0.9 in the beta model reduces the expected estimation error. With this respect, we studied in Chapter 5 the effect of the decay principle on the Beta estimation error, and found that the decay factor does not necessarily reduce the estimation error. It was also concluded that the proper values for the decay factor depend heavily on the stability of real behaviour model of *te* which is assumed to be hidden from its interacting partners. Here, the HMM-based reputation holds the advantage that no knowledge about the real behaviour is needed.

In conclusion, we advise using the beta reputation model when the number of interactions with the trustee is relatively small, and using the proposed HMM-reputation model when this number is large. One advantage in the proposed format of a HMM-reputation report, is that it includes the length of the observed sequence. Based on this length information, a truster can decide whether to use the simple beta model or the HMM model. Any truster can arbitrarily define the least number of interactions with a trustee so that the HMM-reputation model is used.

Consider, for example, a computer network where each peer provides a service (e.g. routing) to other network peers. Assuming that each peer has a fixed identity (e.g. IP address), reputation reports about a trustee can be accumulated based on interactions with this trustee. The beta reputation model can be used initially when the sequences of mutual interactions between peers are relatively short. Over the time, the sequences of mutual interactions between peers become long enough such that the HMM reputation model is more precise for evaluating trust in the network peers than the beta model.

If the network peers are however mobile such that they leave and re-enter the network with different identities, long sequences of interactions with a trustee are not always available in this case and hence using beta reputation model is preferable for evaluating the mutual trust based on the 'short-term' interactions between the network peers.

Chapter 8

Conclusion

In networks of interacting principals, each principal aims at protecting its resources from being abused or destroyed during an interaction with others. This is the reason why each principal usually tries to choose carefully its partners such that the risk of abusing its resources by its chosen partners is minimised. In other words, a principal bases its security policies on an estimated level of trust in each of its network peers. Therefore, the general problem of achieving security between interacting principals amounts essentially to formalising and computing trust between them.

In modern open-ended networks (e.g. the Internet) each principal can have autonomously different behaviours and intentions which are incompletely known by other principals and therefore can affect their security. Moreover, it is not practical in such networks to use third parties to issue and verify credentials (certificates) because this again raises the question of whether these parties are always trusted or not. Given these attributes, the credential-based trust is not entirely appropriate as a basis of interaction in these networks simply because no principal is assumed to be perfectly trusted; there is always a risk of experiencing unsatisfactory outcomes of a given interaction with a partner.

Given the above characteristics of modern open-ended networks, the aim of security decisions is then directed to minimising the risk of unsatisfactory interaction outcomes rather than avoiding the risk at all. This requires assessing the 'likelihood' of the risk associated with interacting with each potential partner. This leads to a modern notion of trust called *probabilistic trust* which is the main concern of this thesis.

The probabilistic trust of a principal (truster) in a particular trustee is defined as the probability distribution, estimated by the truster, over possible outcomes of an interaction between the truster and the trustee. An important advantage of formalising the trust in a trustee as a probability distribution is that mathematical tools provided by the probability theory (e.g. Bayesian inference and maximum likelihood estimators) can be used as formal means to estimate such a distribution based on the available observations

about the trustee. Another advantage of this probabilistic notion of trust is related to its usage as the core of security policies. Namely with the interpretation of trust as a probability distribution over possible interaction outcome, a security policy/protocol can be mathematically optimised so that the probability (likelihood) of experiencing unsatisfactory interaction outcomes is minimised.

In this thesis, we addressed two questions. The first is how the probabilistic trust information can be employed in security protocols to satisfy the requirement of minimising the risk of abusing resources and violating the protocols. The other question is how to evaluate the trust in principals exhibiting dynamic behaviours, given past observations about them. That is using a *hidden Markov models* (HMM) to model the hidden dynamic behaviour of a trustee *te*, it is required to equip a truster *tr* with trust and reputation models for estimating the probability distribution over possible outcomes of the *tr*'s next interaction with *te*.

8.1 Main contributions

8.1.1 Using probabilistic trust in Crowds protocol

In Chapter 4 it is described how the probabilistic trust information about principals can be used to optimise the Crowds anonymity protocol (Reiter and Rubin, 1998) aiming to protecting the identity of the protocol members when they initiate web transactions. In this chapter, each member i of the crowd is associated with a probabilistic trust value t_i denoting its probability of being 'honest' (i.e. conforming with the protocol) when it receives a message from another member. The principal i is therefore assumed to be 'corrupted' (i.e. violating the protocol) with probability $1 - t_i$.

For allowing using such trust values, the original Crowds protocol is extended such that each user i is associated with a preference level of forwarding q_i denoting the probability of choosing it as the next forwarder in the routing process. This allows for specifying the probabilities q_i as the interaction policy, according to the participants trust values.

Given a probability of forwarding p_f, a level of anonymity α, and trust levels $t_1, t_2, \cdots, t_n$ for crowd members, we identified the necessary conditions on the probabilities of forwarding q_i which are necessary to achieve a level of anonymity called α-*probable innocence*. Thus, in presence of untrusted members, the users of the protocol can exploit these results to derive an interaction policy $q_1, q_2, \cdots, q_n$, if there exists any, that ensures them a level of satisfactory anonymity.

8.1.2 Analysis of the decay principle

Since the exponential decay principle was proposed in the literature as an enhancement of the Beta and Dirichlet trust models to cope with principals dynamic behaviour, it is important to study the usefulness and limitations of this approach. Chapter 5 is devoted to this purpose as it provides a detailed analysis of the Beta trust model with exponential decay.

In this chapter, the dynamic behaviour of a trustee *te* is modelled by an ergodic finite-state HMM λ_{te}, called *te*'s 'real model'. Given a current state of λ_{te}, the probability distribution over the outcomes of the next interaction with *te* is computed, and called the *predictive probability distribution*. Applying the Beta trust model along with the decay scheme to a sequence of previous outcomes of interacting with *te*, results in another probability distribution, called the *estimated predictive distribution*. The quality of the trust model is then measured by an *estimation error* defined as the statistical difference between the (real) predictive distribution and the estimated predictive distribution.

A formula has been then derived for the expected estimation error of the Beta model with a decay factor. This formula can be used to understand the implications of choosing a decay factor. To study the effectiveness of the decay technique with respect to the trustee's dynamic behaviour, the expected Beta error is plotted as a function of the decay parameter r according to a notion of system stability. It is found that the optimal value of the parameter r is highly sensitive to the trustee's stability, and the decay technique appears effective only when systems are relatively stable, so that state changes happen infrequently.

8.1.3 HMM-based trust model

In Chapter 6, a novel HMM-based trust model is introduced. Specifically, a trust algorithm is proposed which evaluates an estimated predictive probability distribution given a history of observations about the trustee. This algorithm is based on approximating the behaviour of the trustee by the n-state HMM η which maximises the likelihood of the available history of observations. The approximate model η is then used by the truster to compute the estimated predictive probability distribution.

To assess the quality of the HMM-based trust model, the hidden behaviour of the trustee *te* is assumed to follow a finite-state HMM λ_{te} (called the 'real model'), which determines the (real) *predictive probability distribution* over the outcomes of next interaction between the truster and the trustee. The HMM-based trust model is then formally justified by showing that maximising the likelihood of the given observation sequence minimises the expected relative entropy divergence between the real and estimated predictive probability distributions.

Using a Monte-Carlo algorithm to evaluate the expected relative entropy divergence between the real and estimated predictive probability distributions, an experimental comparison between the HMM-based trust algorithm and the Beta-based trust algorithm with an exponential decay scheme was performed. The results of this comparison show that the HMM-based trust algorithm gives a better estimation for the predictive probability distribution when the trustee's behaviour is highly dynamic. When the real behaviour of the trustee is more stable (less dynamic), the Beta-based algorithm with the optimal value of the decay parameter gives slightly better estimation than the HMM-based algorithm. However this improvement is still tied to the optimal choice of the decay parameter which, as concluded earlier, depends on the 'hidden' behaviour of the trustee.

Nevertheless, it is identified that computing the approximate behaviour model η using only one observation sequence about the trustee *te* (representing direct interactions between the truster and the trustee) requires that this sequence be sufficiently long. With respect to this issue, an approach is also proposed to compute η based on multiple sequences of observations about *te*, where these sequences are observed by other principals (peers) as a result of their interactions with *te*. A fundamental condition of using this approach is that these principals and the truster form a set of 'symmetric peers' S, where the probability distribution over sequences observed by any peer $x \in S$ about *te* is the same regardless of the identity of x.

8.1.4 HMM-based reputation model

In Chapter 7, a model for reputation has been proposed as a supplement to the basic HMM-based trust model. This model is intended to enhance the reliability of the trust evaluation process by using feedback information about the trustee. Specifically, this reputation model provides a formalism of *reputation reports*, i.e. the ratings given by principals (*reputation sources*) about the trustee *te*. The reputation model also provides a *mixing algorithm* which is used by the truster to combine the reputation reports collected from different reputation sources together with its own report in order to evaluate the trust in the trustee *te*.

The experimental approach employed in Chapter 6 for evaluating and comparing trust models in terms of the expected estimation error has also been used in Chapter 7 to investigate the impact of the HMM-reputation model on trust evaluation and also to compare between this model and the existing beta reputation model.

It is found that the estimation error is significantly reduced when multiple reputation reports are used in the trust evaluation process rather than using a single reputation report (which corresponds to the truster's personal experience about the trustee). It is also shown that this improvement due to using reputation reports is getting less sig-

nificant as the total number of interactions with the trustee is getting larger. This is because larger number of total interactions with *te* implies longer sequences of observations experienced by reputation sources (including the truster), and hence a single sequence tends to suffice for learning the trustee's behaviour.

By comparison with the Beta reputation model, using the same number of reputation sources, it is found that the Beta reputation model outperforms the HMM-based reputation when the total number T of interactions with the trustee is relatively small. As T gets larger, the HMM-based reputation model gradually improves in terms of the estimation error, and eventually outperforms the Beta model when T is relatively large. In fact, larger T implies longer sequences of observations about the trustee; such sequences reflect more information about the trustee's dynamic behaviour and therefore yield a better approximation for such a dynamic behaviour which, in contrast, is ignored by the Beta reputation model.

8.2 Future Work

8.2.1 Implementation

While this work has addressed the theoretical foundations of trust models, a promising continuation can be directed to implementing such trust and reputation models in existing Internet access applications to evaluate the trust in e.g., web sites, FTP servers, and web services. Technically, such models can be implemented as a *plug-in* which is integrated with the existing web browsers. Such a 'trust evaluation' plug-in can automatically log the outcomes of interactions with different web sites, and therefore by processing such outcomes, can provide the user estimates about the reliability of these web sites. Reputation reports can be also phrased out of this logged information and exchanged with other Internet users who also use instances of the 'trust evaluation' plug-in.

Another potential goal can be set to provide the *service oriented architectures* (SOAs) a means to evaluate the probabilistic trust in the underlying web services. While trust and reputation functions can be implemented as separate components and integrated with the SOAs, these functions can be rather implemented in the SOA *implementation framework* (SOAIF), e.g. JAVA EE and .NET, where the *runtime environment*, in which the SOA is running, automatically logs the outcomes of calls to the web services, and therefore phrases reputation reports about these services. Implementing the trust and reputation models on the SOAIF level provides a transparent means of exchanging reputation reports between different SOAs which use the same web services.

However, an important issue to be handled in the above implementation ideas is the mechanism of exchanging reputation reports between principals. A simplistic solution

can be utilising a central server whose function is gathering reputation reports and relaying them to the requesting principals. Despite the simplicity of this approach, it suffers from two problems. The first is that the central server represents a single point of failure; if this server fails to receive or deliver reputation reports, all principals relying on this server are dramatically affected. The second problem is the possibility that the server itself attacks the reputation of the web sites/services by altering the reputation reports. In this respect, encryption mechanisms can provide a means to preserve the confidentiality of exchanged reputation reports, by hiding the reports from the server while allowing end principals to read them by decryption. A more promising approach to exchanging reputation reports is to adopt peer-to-peer protocols which can typically allow communicating the reputation reports between principals without the need to a central coordination server.

8.2.2 Handling inaccurate reputation reports

In HMM-based reputation model, we had an implicit assumption that each reputation source is honest. This means that each reputation source bases its opinion (reputation reports) on actual outcomes between it and the trustee. Practically, some reputation sources can provide corrupt reputation reports about the trustee to satisfy hidden selfish attitudes. This problem is addressed in other reputation models (e.g. Teacy et al., 2006) by using a mechanism for discounting the effect of the reports given by these 'lying' sources according to the reputation of these sources themselves. Researching analogous approaches to tackle this problem in the HMM-based reputation model is a useful extension.

Bibliography

H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer. Estimating hidden markov model parameters so as to maximize speech recognition accuracy. *Speech and Audio Processing, IEEE Transactions on*, 1(1):77 –83, January 1993. ISSN 1063-6676.

C. Baier, B. Engelen, and M. E. Majster-Cederbaum. Deciding bisimilarity and similarity for probabilistic processes. *Journal of Computer and System Sciences*, 60(1): 187–231, 2000.

L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite-state Markov chains. *Annals of Mathematical Statistics*, 37(6):1554–1563, Dec 1966.

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

M.Y. Becker and P. Sewell. Cassandra: distributed access control policies with tunable expressiveness. In *Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on*, pages 159 – 168, 2004.

M. Bhargava and C. Palamidessi. Probabilistic anonymity. In Martín Abadi and Luca de Alfaro, editors, *CONCUR*, volume 3653 of *Lecture Notes in Computer Science*, pages 171–185. Springer, 2005. ISBN 3-540-28309-9.

M. Bicego, A. Dovier, and V. Murino. Designing the minimal structure of hidden markov model by bisimulation. In *Proceedings of the Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, EMMCVPR '01, pages 75–90, London, UK, 2001. Springer-Verlag. ISBN 3-540-42523-3.

P. J. Bickel and K. A. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*, volume 1. Prentice Hall, 2 edition, September 2000. ISBN 013850363X.

P. Billingsley. *Ergodic Theory and Information*. John Wiley, 1965.

P. Billingsley. *Probability and Measure*. Wiley-Interscience, 3 edition, April 1995. ISBN 0471007102.

J. Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, University of Berkeley ICSI-TR-97-021, 1997.

M. Blaze, J. Feigenbaum, and A. D. Keromytis. KeyNote: Trust management for public-key infrastructures. In *Proceedings from Security Protocols: 6th International Workshop, Cambridge, UK, April 1998*, volume 1550, pages 59–63, 1999.

M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings from the 17th Symposium on Security and Privacy*, pages 164–173. IEEE Computer Society Press, 1996.

M. Blaze, J. Feigenbaum, and M. Strauss. Compliance checking in the policymaker trust management system. In *Proceedings from Financial Cryptography: Second International Conference (FC'98), Anguilla, British West Indies, February 1998*, pages 254–274, 1998.

P. Bonatti and D. Olmedilla. Driving and monitoring provisional trust negotiation with metapolicies. In *Policies for Distributed Systems and Networks, 2005. Sixth IEEE International Workshop on*, pages 14 – 23, 2005.

P. Brémaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer, 1998.

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

V. Cahill, E. Gray, J. M. Seigneur, C. D. Jensen, Y. Chen, B. Shand, N. Dimmock, A. Twigg, J. Bacon, C. English, W. Wagealla, S. Terzis, P. Nixon, G. Serugendo, C. Bryce, M. Carbone, K. Krukow, and M. Nielsen. Using trust for secure collaboration in uncertain environments. *IEEE Pervasive Computing*, 2(3):52–61, 2003. ISSN 1536-1268.

M. Carbone, M. Nielsen, and V. Sassone. A formal model for trust in dynamic networks. In *Proceedings from Software Engineering and Formal Methods (SEFM'03)*. IEEE Computer Society Press, 2003.

G. Casella and R. Berger. *Statistical Inference*. Duxbury Resource Center, June 2001. ISBN 0534243126.

K. Chatzikokolakis and C. Palamidessi. Probable innocence revisited. *Theor. Comput. Sci.*, 367(1-2):123–138, 2006.

K. Chatzikokolakis, C. Palamidessi, and P. Panangaden. Probability of error in information-hiding protocols. In *CSF*, pages 341–354. IEEE Computer Society, 2007.

K. Chatzikokolakis, C. Palamidessi, and P. Panangaden. Anonymity protocols as noisy channels. *Inf. Comput.*, 206(2-4):378–401, 2008a.

K. Chatzikokolakis, C. Palamidessi, and P. Panangaden. On the Bayes risk in information-hiding protocols. *Journal of Computer Security*, 16(5):531–571, 2008b.

D. Chaum. Security without identification: transaction systems to make big brother obsolete. *Commun. ACM*, 28:1030–1044, October 1985. ISSN 0001-0782.

T. M. Cover and J. A. Thomas. *Elements of Information Theory 2nd Edition*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, July 2006. ISBN 0471241954.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society, Series B*, 39(1), 1977.

P. Dupont, F. Denis, and Y. Esposito. Links between probabilistic automata and hidden markov models: probability distributions, learning models and induction algorithms. *Pattern Recognition*, 38(9):1349–1371, September 2005.

S. Eickeler, A. Kosmala, and G. Rigoll. Hidden Markov Model Based Continuous Online Gesture Recognition. In *Int. Conference on Pattern Recognition (ICPR)*, pages 1206–1208, Brisbane, 1998.

C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomsa, and T. Ylonen. SPKI Certificate Theory. RFC 2693, ftp-site: ftp://ftp.rfc-editor.org/in-notes/rfc2693.txt, September 1999.

E. ElSalamouny, K. Krukow, and V. Sassone. An analysis of the exponential decay principle in probabilistic trust models. *Theoretical Computer Science*, 410(41):4067 – 4084, 2009. ISSN 0304-3975.

E. ElSalamouny, V. Sassone, and M. Nielsen. Hmm-based trust model. In Pierpaolo Degano and Joshua Guttman, editors, *Formal Aspects in Security and Trust*, volume 5983 of *Lecture Notes in Computer Science*, pages 21–35. Springer Berlin / Heidelberg, 2010.

G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

E. Gabber, P. B. Gibbons, Y. Matias, and A. Mayer. How to make personalized web browsing simple, secure, and anonymous. *Lecture Notes in Computer Science*, 1318: 17–32, 1997.

D. Gambetta. *Can We Trust Trust?* Basil Blackwell, 1988.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman and Hall/CRC, 2 edition, July 2003. ISBN 158488388X.

G. Grimmet and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, third edition, 2001.

J. Y. Halpern and K. R. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 13(3):483–512, 2005.

S. Hamadou, C. Palamidessi, V. Sassone, and E. ElSalamouny. Probable innocence in the presence of independent knowledge. In Pierpaolo Degano and Joshua Guttman, editors, *Formal Aspects in Security and Trust*, volume 5983 of *Lecture Notes in Computer Science*, pages 141–156. Springer Berlin / Heidelberg, 2010.

R. A. Horn and C. A. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, UK, 1985.

R. Hughey and A. Krogh. Hidden Markov models for sequence analysis: extension and analysis of basic method. *Comp. Appl. BioSci*, 12(2):95–108, 1996.

ITU-T. *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, Recommendation X.509*. International Telecommunication Union, 1993. update.

W. J. Hu; Brown, M.K.; Turin. HMM based online handwriting recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, Oct 1996. ISSN 0162-8828.

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

A. Jøsang and J. Haller. Dirichlet reputation systems. In *The Second International Conference on Availability, Reliability and Security, 2007. ARES 2007.*, pages 112–119, 2007.

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

K. Krukow, M. Nielsen, and V. Sassone. Trust models in Ubiquitous Computing. *Philosophical Transactions of the Royal Society A*, 366(1881):3781–3793, 2008.

K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, September 1991.

P. Malacaria and H. Chen. Lagrange multipliers and maximum information leakage in different observational models. In Úlfar Erlingsson and Marco Pistoia, editors, *PLAS*, pages 135–146. ACM, 2008. ISBN 978-1-59593-936-4.

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

I. J. Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90 – 100, 2003. ISSN 0022-2496.

W. Nejdl, D. Olmedilla, and M. Winslett. Peertrust: Automated trust negotiation for peers on the semantic web. In Willem Jonker and Milan Petkovic, editors, *VLDB Workshop on Secure Data Management (SDM)*, volume 3178 of *Lecture Notes in Computer Science*, pages 159–182. Springer Berlin / Heidelberg, 2004.

B.C. Neuman and T. Ts'o. Kerberos: an authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33 –38, sep. 1994. ISSN 0163-6804.

M. Nielsen and K. Krukow. Towards a formal notion of trust. In *Proceedings from the 5th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming (PPDP'03)*, pages 4–7. ACM Press, 2003.

M. Nielsen, K. Krukow, and V. Sassone. A bayesian model for event-based trust. *Electronic Notes in Theoretical Computer Science (ENTCS)*, 172:499–521, April 2007. ISSN 1571-0661.

J. R. Norris. *Markov chains*. Cambridge University Press, 1997.

S. M. Omohundro. Best-first model merging for dynamic learning and recognition. In John E. Moody, Steve J. Hanson, and Richard P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 958–965. Morgan Kaufmann Publishers, Inc., 1992.

J. S. Park and R. Sandhu. Binding identities and attributes using digitally signed certificates. In *Proceedings of the 16th Annual Computer Security Applications Conference*, ACSAC '00, pages 120–, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0859-6.

K. Petersen. *Ergodic theory*. Cambridge University Press, 1990.

A. Pfitzmann and M. Waidner. Networks without user observability. *Comput. Secur.*, 6:158–166, May 1987. ISSN 0167-4048.

PGPi website. An introduction to cryptography, in pgp user's guide 7.0. ftp: ftp://ftp.pgpi.org/pub/pgp/7.0/docs/english/IntroToCrypto.pdf website: http://www.pgpi.org/doc, 2000.

L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall PTR, united states ed edition, April 1993. ISBN 0130151572.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989. ISSN 0018-9219.

S. T. Rachev. *Probability Metrics and the Stability of Stochastic Models*. John Wiley & Sons, 1991. ISBN 0-471-92877-1.

M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and Systems Security*, 1(1):66–92, 1998.

R. L. Rivest and B. Lampson. SDSI - A Simple Distributed Security Infrastructure. Presented at CRYPTO'96 Rumpsession, 1996.

R. Sandhu. Rationale for the RBAC96 family of access control models. In *Proceedings of the first ACM Workshop on Role-based access control*, RBAC '95, New York, NY, USA, 1996. ACM. ISBN 0-89791-759-6.

V. Sassone, E. ElSalamouny, and S. Hamadou. Trust in crowds: Probabilistic behaviour in anonymity protocols. In Martin Wirsing, Martin Hofmann, and Axel Rauschmayer, editors, *Trustworthly Global Computing*, volume 6084 of *Lecture Notes in Computer Science*, pages 88–102. Springer Berlin / Heidelberg, 2010.

V. Sassone, K. Krukow, and M. Nielsen. Towards a formal framework for computational trust. In Frank S. de Boer, Marcello M. Bonsangue, Susanne Graf, and Willem P. de Roever, editors, *FMCO*, volume 4709 of *Lecture Notes in Computer Science*, pages 175–184. Springer, 2006. ISBN 978-3-540-74791-8.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464, 1978.

K. Seymore, A. McCallum, and R. Rosenfeld. Learning hidden Markov model structure for information extraction. In *AAAI 99 Workshop on Machine Learning for Information Extraction*, 1999.

V. Shmatikov and M. Wang. Measuring relationship anonymity in mix networks. In Ari Juels and Marianne Winslett, editors, *WPES*, pages 59–62. ACM, 2006. ISBN 1-59593-556-8.

D. S. Sivia. *Data Analysis: A Bayesian Tutorial (Oxford Science Publications)*. Oxford University Press, July 1996. ISBN 0198518897.

G. Smith. On the foundations of quantitative information flow. In Luca de Alfaro, editor, *Proceedings of the Twelfth International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2009)*, volume 5504 of *Lecture Notes in Computer Science*, pages 288–302. Springer Berlin / Heidelberg, 2009.

A. Stolcke and S. Omohundro. Hidden Markov Model induction by bayesian model merging. In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, pages 11–18. Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-274-7.

W. Teacy, J. Patel, N. Jennings, and M. Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12:183–198, 2006. ISSN 1387-2532.

A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lott. KAoS policy and domain services: toward a description-logic approach to policy representation, deconfliction, and enforcement. In *Proceedings of*

the 4th IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY '03, pages 93–96, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1933-4.

D. Varacca, H. Völzer, and G. Winskel. Probabilistic event structures and domains. In Philippa Gardner and Nobuko Yoshida, editors, *CONCUR*, volume 3170 of *Lecture Notes in Computer Science*, pages 481–496. Springer, 2004. ISBN 3-540-22940-X.

E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. C. Carrasco. Probabilistic finite-state machines-part i. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1013–1025, 2005a. ISSN 0162-8828.

E. Vidal, F. Thollard, C. de la Higuera, F. Casacuberta, and R. C. Carrasco. Probabilistic finite-state machines-part ii. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1026–1039, 2005b. ISSN 0162-8828.

W.H. Winsborough, K.E. Seamons, and V.E. Jones. Automated trust negotiation. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings*, 2000.

M. Winslett, T. Yu, K. E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and L. Yu. Negotiating trust on the web. *IEEE Internet Computing*, 6(6):30–37, 2002. ISSN 1089-7801.

L. Xiong and L. Liu. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on knowledge and data engineering*, 16 (7):843–857, 2004.