

Advisor Networks and Referrals for Improved Trust Modelling in Multi-Agent Systems

by

Joshua Mark Gerner

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2011

© Joshua Mark Gerner 2011

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis relates to the usage of trust modelling in multi-agent systems – environments in which there are interacting software agents representing various users (for example, buyers and sellers exchanging products and services in an electronic marketplace). In such applications, trust modelling may be crucial to allow one group of agents (in the e-commerce scenario, buyers) to make effective decisions about which other agents (i.e., sellers) are the most appropriate partners. A number of existing multi-agent trust models have been proposed in the literature to help buyers accurately select the most trustworthy sellers.

Our contribution is to propose several modifications that can be applied to existing probabilistic multi-agent trust models. First, we examine how the accuracy of the model can be improved by limiting the network to a portion of the population consisting of the most trustworthy agents, such that the less trustworthy contributions of the remaining agents can be ignored. In particular, we explore how this can be accomplished by either setting a maximum size for a buyer’s advisor network or setting a minimum trustworthiness threshold for agents to be accepted into that advisor network, and develop methods for appropriately selecting the values to limit the network size. We demonstrate that for two models, both the Personalized Trust Model (PTM) developed by Zhang as well as TRAVOS, these approaches will yield significant improvements to the accuracy of the trust model, as opposed to using an unrestricted advisor network.

Our final proposed modification is to use an advisor referral system in combination with one of the network-limiting approaches. This would ensure that if a particular agent within the advisor network had not met a specified level of experience with the seller under consideration, it could be replaced by another agent that had greater experience with that seller, which should in turn allow for a more accurate modelling of the seller’s trustworthiness. We present a particular approach for replacing advisors, and show that this will yield additional improvements in trust-modelling accuracy with both PTM and TRAVOS, especially if the limiting step were such that it would yield a very small advisor network.

We believe that these techniques will be very useful for trust researchers seeking to improve the accuracy of their own trust models, and to that end we explain how other researchers could apply these modifications themselves, in order to identify the optimal parameters for their usage. We discuss as well the value of our proposals for identifying an “optimal” size for a social network, and the use of referral systems, for researchers in other areas of artificial intelligence.

Acknowledgements

First of all, I wish to express my deep appreciation for the support and advice provided by my supervisor, Dr. Robin Cohen, throughout my graduate studies at the University of Waterloo, and particularly during the writing of this thesis. Dr. Cohen spent numerous hours providing suggestions for my research, co-authoring several papers with me, and helping to revise this thesis. This support has been invaluable to me and I consider myself very fortunate to have had her as an advisor.

I also wish to thank Dr. Kate Larson and Dr. Peter van Beek for devoting a portion of their valuable time to serve as readers for this thesis.

I am grateful as well to Dr. Jie Zhang, who provided invaluable assistance in the form of the source code from his earlier work, much of which has been modified and reused for the experimental phases of this thesis. Dr. Zhang also provided useful suggestions and input throughout my research.

Finally I wish to express my thanks to the staff, faculty, and fellow graduate students for their assistance, input, and friendship during my time at this university. More specifically I extend my appreciation to, in no particular order, John Champaign, Hyunggu Jung, Simina Brânzei, Lachlan Dufton, Greg Hines, Lisa Nagy, John Finnson, Reid Kerr, Tyrel Russel, Dr. Georgia Kastidou, and likely others to whom I apologize for their omission.

Financial support for this research was provided in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), through the Alexander Graham Bell Canada Graduate Scholarships program. My advisor, Dr. Robin Cohen, is the recipient of additional funding from NSERC that also funded my research.

Dedication

To my parents, for making it all possible.

Table of Contents

List of Tables	xv
List of Figures	xviii
List of Algorithms	xix
1 Introduction	1
1.1 Trust Modelling	2
1.2 Thesis Organization	5
2 Related Work	7
2.1 Trust Models	7
2.1.1 Personalized Trust Model (PTM)	7
2.1.2 Other Trust Models	13
2.2 Techniques for Limiting Network Size	16
2.3 Referral Systems	17
3 Limiting Advisor Network Size	21
3.1 Proposed Techniques	21
3.1.1 Trustworthiness Thresholding	22

3.1.2	Maximum Number of Advisors	22
3.2	Examples	23
3.2.1	Using the PTM	23
3.2.2	Reputation Thresholding	26
3.2.3	Maximum Number of Advisors	27
3.3	Experimental Results	27
3.3.1	Validating Effectiveness	28
3.3.2	Applicability to Alternative Model	36
3.3.3	Effects for Larger Populations	38
3.3.4	Using Random Selection for Very Large Populations	43
3.3.5	Experimental Conclusions	45
4	Advisor Referral Systems	49
4.1	Proposed Technique	49
4.2	Examples	52
4.3	Experimental Results	54
4.3.1	Verifying Effectiveness	54
4.3.2	Using Referrals to Further Reduce Network Size	55
4.3.3	Applicability to Alternative Model	61
4.3.4	Effects for Larger Populations	65
5	Discussion	69
5.1	Applying Techniques to Other Trust Models	69
5.1.1	Optimizing the Modifications	71
5.2	Selecting Appropriate Parameters	74

5.2.1	Comparison with Collaborative Filtering	74
5.2.2	Comparison with Advisor Referrals	78
5.3	Alternative Referral Trust Computation Methods	80
5.3.1	Related Work	80
5.3.2	Proposed Alternatives	86
5.4	Balancing Similarity and Reputation Modelling	95
6	Conclusion and Future Work	97
6.1	Contributions	97
6.2	Future Work	99
6.2.1	Parameter Selection	99
6.2.2	Improvements to Testing Methods	101
6.2.3	Further Verification of Effects on TRAVOS	102
6.2.4	Additional Work with Large Populations	102
6.2.5	Performance and Time Sensitivity	104
6.2.6	Combining Collaborative Filtering and Trust Modelling	105
6.2.7	Information Gain	106
6.2.8	Usefulness of Referrals	106
6.2.9	Domain Specificity	108
	APPENDICES	109
	A Relationship Between max_nbors and Thresholding	111
	References	119

List of Tables

3.1	Ratings of Sellers Provided by Advisors and Buyer b	24
3.2	Trustworthiness of Advisors $a_w, a_x, a_y,$ and a_z for Buyer b	25
3.3	Ratings of s_0 Provided by a_w, a_x, a_z	25
3.4	Advisor Network Size with a Correlation Threshold	26
3.5	Trustworthiness of s_0 Using a Maximum Number of Advisors	27
4.1	Ratings of s_6 Provided by Advisors	53
5.1	Calculations of Normalized Local Trust Values for Simplified EigenTrust Example	87
5.2	Summary of Proposed Referral Trust Weighting Methods	89

List of Figures

3.1	Verification testing for the modifications using 30% lying advisors.	31
3.2	Verification testing for the modifications using 60% lying advisors.	32
3.3	Mean average error of various trust model variants	34
3.4	Mean absolute error when applying optimizations to TRAVOS at 30% and 60% LA	39
3.5	Mean absolute error applying optimizations to PTM with advisor population of 500	40
3.6	Comparison of mean absolute error when varying max_nbors (as proportion of advisor population)	41
3.7	Mean absolute error applying optimizations to TRAVOS with advisor population of 500, compared to results for population of 80	42
3.8	Verification testing for the modifications.	46
4.1	Comparison of approaches with and without referrals.	56
4.2	Mean average error of compared trust models, both with and without referrals.	57
4.3	Comparison of trust models ($max_nbors = 2$, 30% lying advisors (LA)	58
4.4	Evaluation of effects of referrals on small max_nbors advisor networks.	59
4.5	Evaluation of effects of referrals on advisor networks.	62
4.6	Comparison of mean absolute error in TRAVOS using advisor referrals when varying max_nbors and the minimum level of referral experience (N_{RE})	63

4.7	Comparison of mean absolute error in TRAVOS using advisor referrals when varying the trust threshold and the minimum N_{RE} level	64
4.8	Comparison of mean absolute error in PTM using advisor referrals, varying threshold and minimum N_{RE} , when advisor population is 500	67
4.9	Comparison of mean absolute error in PTM using advisor referrals, varying (proportional) max_nbors and minimum N_{RE} , when advisor population is 500	68
5.1	An example of the usage of weighted referral graphs used in [40].	81
5.2	An example to illustrate a simplified EigenTrust “asking friends” mechanism.	86
5.3	The pertinent relationships and trust values held in the examples discussed in Section 5.3.2.	90
A.1	Mean equivalent values for the two proposed network-limiting methods . .	113

List of Algorithms

1	PTM Method for Buyer b 's Trustworthiness Model of a Seller s (From [42])	12
2	Referring Advisors to Buyer b for Trustworthiness of Seller s	51
3	Testing Modifications with a Trust Model: Determine Advisor Trusts . . .	75
4	Testing Modifications with a Trust Model: Determine Seller Trusts	76

Chapter 1

Introduction

Within the broad field of artificial intelligence, much recent work has been devoted to intelligent agents, or software-based entities that are capable of making their own decisions autonomously, often with reference to changes in their environment and interactions with other agents [36]. Environments in which several such agents exist and interact with one another are called multi-agent systems [10].

In these systems, trust often plays an important role. Individual agents may act in their own self-interest to maximize their benefits from the system, and in some cases they attempt to do so through deception or by providing misinformation to other agents. It is thus critical that agents are able to accurately determine the trustworthiness of other agents. While “trust” may often have different meanings from one domain to the next [24], for the purposes of our discussion, our work makes use of the definition of “trust” as an agent’s belief that some other agent will carry out the tasks it says it will perform [42].

While we believe that the proposals and findings presented herein will be applicable to a number of different scenarios, we will be grounding our discussion in one particularly pertinent application, specifically electronic commerce, in which customers may shop for and purchase goods or services from sellers over the Internet, even though the buyer and the seller may be separated geographically by a long distance. Each buyer and each seller would be represented by an intelligent agent, reasoning about which partners to select, to conduct business.

Regardless of the source from which a customer chooses to purchase items, there is always a need for some element of trust that the seller will carry through on the promise to deliver the purchased items. When purchasing directly from a large, well-known retailer like Amazon.com (<http://www.amazon.com/>), this trust is usually more implicit because that retailer has already derived a reputation from selling items to consumers over a number of years. In other cases, however, such as purchases through the electronic-auction marketplace eBay (<http://www.ebay.com/>) – and for certain goods on Amazon that are not stocked by the company itself, e.g. out-of-print books – the website is merely acting as an intermediary for some other person or company. In such cases the customer will typically need to pay much more attention to the seller’s reputation to ensure that he or she is not being taken advantage of.

1.1 Trust Modelling

Trust modelling is a field of artificial intelligence focusing on determining the trustworthiness of agents. This is typically conducted in contexts where a requesting agent is considering interacting with a providing agent, and is based on the information about the providing agent that is available to the requesting agent. It is especially pertinent with regards to multi-agent systems such as electronic marketplaces, in which customers (or buyers) often need to decide whether a seller is sufficiently trustworthy to purchase goods or services, and several recent researchers in the field have focused on this application [39][33][43].

Some of the initial work in the area defined this information solely in terms of the direct past interactions between the requesting agent and the agents under consideration [20][33]. However, more contemporary systems also consider the availability of information from other agents that might be able to better inform the initial agent – we refer to these agents as *advisors*, as they may provide advice to the initial agent about an item (or, indeed, another agent) of interest. This additional information is particularly pertinent on sites such as eBay, which computes reputation scores for each seller based on the ratings assigned by different buyers for each interaction.

In regards to multi-agent trust models, our focus in this thesis will be on the subcategory of *probabilistic* models. By “probabilistic” we mean that the models in question make use of probability density functions (pdfs) in order to estimate the trustworthiness of providing agents. Most commonly the pdf used is the beta probability distribution function, which is used to represent probability distributions associated with binary events, indexed by two parameters α and β . In essence, it is used to compute the relative likelihood of the values for some probability parameter p given fixed parameters α and β . This function may be expressed as follows [13]:

$$\text{beta}(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1} (1 - p)^{\beta-1} \quad (1.1)$$

where Γ is the gamma function, $0 \leq p \leq 1$, and $\alpha, \beta > 0$. The α and β parameters define the shape of the density function when plotted, and are generally defined in terms of the numbers of positive results and negative results, respectively [32][42], as will be discussed in detail in Chapter 2.

Probabilistic models that have been previously documented include the Beta Reputation System [13], the TRAVOS model [32], and the Personalized Trust Model (PTM) offered by Zhang [43], the latter of which is geared primarily towards electronic marketplace, whereas the first two are more general. All three of these models make use of beta pdfs. We will discuss two of these models, PTM and TRAVOS, in greater detail in Chapter 2.

This thesis documents our work, the aims of which are twofold. The first is to improve the accuracy of such multi-agent trust modelling systems. While many of the existing trust models seem to provide excellent results in accurately modelling the trustworthiness of agents, there is some room for improvement, which we address with our proposals.

The other aim is to work towards finding the “ideal” size of a requesting agent’s *social network* of advisors. In this context, an agent’s social network is a collection of the other agents which directly interact with the first agent. This does not imply, however, that the physical individuals whom those agents represent necessarily have any knowledge of one another. Indeed, the techniques we outline in this thesis are not at all affected by any

friendships or other direct relationships, or lack thereof, between individuals; it has little in common with social networking websites like Facebook (<http://www.facebook.com/>). For this reason, we also refer to this concept as the agent’s *neighbourhood* or *advisor network*.

It has been previously noted [9][42] that having a larger number of advisors will increase the computation time required to calculate these advisors’ trustworthiness, and may indeed result in poorer accuracy in predicting the trustworthiness of provider agents, due to the inclusion of additional “outlier” data. At the same time, while using a smaller number of advisors may improve the accuracy somewhat, those advisors may not have a sufficient level of experience. We therefore need to determine an appropriate “sweet spot” – not too large and not too small – that will provide us with the most reliable results.

We first examine two methods that limit the size of a requesting agent’s advisor network. Specifically, we consider selecting a maximum number (or maximum proportion) of advisors (or *max_nbors*) out of the total advisor population, or only selecting advisors that have achieved some trustworthiness *threshold*.

We show that by using these methods – with appropriately-chosen parameters – to limit the size of the advisor network, we will obtain an overall more accurate measure of the trustworthiness of individual agents that the requesting agent is considering. We thus also look at how to best determine the parameters to use for a given scenario, in order to ensure the more accurate trust-modelling results. We further show that these results should not be specific to any single model or scenario.

We then discuss the augmentation of one of these two techniques with a referral system, in which advisors which have had an insufficient amount of experience with a particular agent will be replaced by other agents with a higher level of experience. We show that careful application of our proposed advisor referral system will result in further improvements in trust modelling accuracy, particularly when the advisor network is initially very small, such as when a small maximum number of advisors or a high trustworthiness threshold is used. As with the earlier methods, we also examine how to determine the most appropriate parameters when using advisor referrals.

As we explain further in Chapter 2, some researchers have previously looked at techniques, either in trust modelling or related fields such as collaborative filtering, that to some

extent resemble those that we propose in this thesis [9][40]. We believe the value of our work lies in not merely providing improvements to the use of these techniques within trust modelling, but also in setting out a more comprehensive procedure for carefully applying those techniques. This procedure will ultimately demonstrate how to best determine an appropriate size for a social network of this type.

1.2 Thesis Organization

The remainder of this thesis is organized as follows.

In Chapter 2, we provide an overview of some of the earlier work that directly influences our contributions in this thesis. Specifically, we outline the PTM as well as TRAVOS to provide context to our proposals and to the examples and experiments which follow. We then summarize the literature with regards to limiting network size, as well as referral systems, in other domains.

Chapter 3 discusses our two proposals for limiting the size of advisor networks: setting a maximum size to the advisor network (or *max_nhors*), and trustworthiness thresholding. We outline how these techniques would be applied to the PTM, and later provide a general overview of how they could be applied to other trust models. We also provide demonstrative examples of how these techniques function and how they affect the trust models that are produced. Finally we provide experimental results demonstrating the effectiveness of these techniques with both PTM and TRAVOS using moderately-sized advisor populations, as well as with PTM when the advisor population is larger. Included here is an examination of the use of random selection to overcome performance issues sometimes associated with large populations.

Similarly, in Chapter 4 we outline our proposal for advisor referrals in combination with *max_nhors* or thresholding, and their application to the PTM and, potentially, to other trust systems. We then continue with the examples and experimental results begun in the previous chapter with specific reference to the effectiveness of advisors in both PTM and TRAVOS. This includes an exploration using PTM for large population sizes as well.

This is followed in Chapter 5 by some discussion of the value of our particular approach and its implications for other researchers. Included is a suggested method of applying our techniques to other trust models, an exploration of some of the outstanding questions related to choosing appropriate parameters in our approach, and an investigation of alternative methods for weighting advisor referrals which may be used within our framework.

Finally, in Chapter 6, we outline some potential future work that could follow from our current results, and provide some concluding comments.

Chapter 2

Related Work

2.1 Trust Models

2.1.1 Personalized Trust Model (PTM)

We ground this work, initially, in the multi-stage “personalized” trust modelling approach developed by Zhang for representing reputation in an e-commerce system [43]. This approach, which for brevity we will refer to as the Personalized Trust Model or PTM, is summarized below.

Buyer agents regularly interact with seller agents to purchase desired goods or services. Following each transaction with a seller, a buyer assigns a rating to that transaction, specifying whether its experience was positive (1) or negative (0) – more fine-grained ratings are not provided for at present in the PTM – and submits this rating to a centralized database server of some kind.

A single buyer agent, denoted by b , may wish to model the trustworthiness of all the sellers in the system, in order to determine which sellers to purchase from in the future. To do so, it first constructs a measure of the private reputation of each of the other buyer agents – that is, b 's *advisors* – based on the advisors' ratings for sellers that b has previously dealt with, as retrieved from this central server, and representing an estimation of the probability that an advisor a will give fair ratings to b .

For *each* seller s for which both b and a have submitted ratings, there will be a corresponding pair of ratings, which may be classified as either positive if the ratings are the same (i.e. both ‘0’ or both ‘1’), or negative if the two ratings differ. Each pair of ratings considered is assigned a weight based on the amount of time that separates the submission of the two ratings using a “forgetting factor”, λ ($0 \leq \lambda \leq 1$), such that a pair of ratings will have a greater weight if they are made within close time proximity. The PTM uses the concept of a “time window” to represent this temporal proximity: ratings are partitioned into several elemental time windows, the length of which could either be fixed (e.g. two days) or variable based on how frequently a seller is rated. If the ratings are submitted during the same time window, the weight z for that rating pair will be 1; otherwise, it will be calculated as follows:

$$z = \lambda^{T_a - T_b} \quad (2.1)$$

where T_a and T_b are integer values identifying two time windows, and T_b corresponds to the more recent of these two time windows.¹

The overall evaluation of the shared experiences of buyer b with advisor a is known as a ’s “private” reputation, and is estimated as the probability that a will provide fair ratings to b . However, given that the buyer only has incomplete information about an advisor, the best estimation of this probability is through calculating its expected value. As noted in Section 1.1, the beta probability distribution function is frequently used to represent probability distributions of binary events. Thus the private reputation of advisor a is calculated as shown in Equation 2.2:

$$\alpha = N_p + 1, \quad \beta = N_{all} - N_p + 1, \quad R_{pri}(a) = E(Pr(a)) = \frac{\alpha}{\alpha + \beta} \quad (2.2)$$

In Equation 2.2, N_p represents the sum of the weights (as calculated using Equation 2.1) of all positive rating pairs for all sellers commonly rated by b and a , and N_{all} is the

¹ T_b will be the smaller of the two values, such that 1 indicates the most recent time window, 2 the second-most recent time window, and so on.

total sum of weights of all rating pairs involving b and a . If $\lambda = 0$, then N_p and N_{all} will be simply the counts of the applicable types of rating pairs. Note also that $Pr(a)$ is the probability that advisor a will provide fair ratings to the buyer b ; an advisor's rating is considered to be "fair" if it is the same as the buyer's rating. Thus $E(Pr(a))$ represents the probability expectation value of the beta distribution defined by α and β ; that is, it is the most likely probability value that a will be honest in the future [14].

Next, the public reputation of an advisor, which (in a similar fashion to the determination of private reputation) is estimated as the probability that an advisor will provide "consistent" ratings, is calculated using Equation 2.3:

$$\alpha' = N_c + 1, \quad \beta' = N'_{all} - N_c + 1, \quad R_{pub}(a) = \frac{\alpha'}{\alpha' + \beta'} \quad (2.3)$$

Here, N_c represents the number of ratings, provided by an advisor a , that are consistent with the majority of ratings provided for that seller by all other buyers up to the moment that this additional rating is submitted, while N'_{all} is the total number of ratings provided by a .

At this point, given some maximum acceptable level of error $\epsilon \in (0, 1)$ and level of confidence $\gamma \in (0, 1)$, w , the reliability of the private reputation value, is derived – which is then used in the calculation of the overall trustworthiness of a (as in Equations 2.4, 2.5, and 2.6). As can be seen from Equation 2.6, a more reliable private reputation will have a greater effect on the overall result.

$$N_{min} = -\frac{1}{2\epsilon^2} \ln \frac{1-\gamma}{2} \quad (2.4)$$

$$w = \begin{cases} \frac{N_{all}}{N_{min}} & \text{if } N_{all} < N_{min} \\ 1 & \text{otherwise} \end{cases} \quad (2.5)$$

$$Tr(a) = wR_{pri}(a) + (1-w)R_{pub}(a) \quad (2.6)$$

Once the trustworthiness value has been calculated for each advisor, advisors may be

then classified as “trustworthy” or “untrustworthy”. If the advisor’s trustworthiness is 0.5 or above, it will be considered trustworthy; otherwise it will be considered untrustworthy.

A similar approach can next be taken for the trustworthiness of a given seller s , once again making use of the beta family of probability density functions to estimate appropriate probabilities. First the buyer b calculates her private reputation of s , an estimation of the probability that s will provide good service, based on b ’s past experiences with s . This makes use of the number of positive ratings, $N_{pos,i}^b$, and negative ratings, $N_{neg,i}^b$, she provided for s in each time window T_i , as well as the forgetting factor λ , as in Equation 2.7.

$$R_{pri}(s) = \frac{\sum_{i=1}^n N_{pos,i}^b \lambda^{i-1} + 1}{\sum_{i=1}^n (N_{pos,i}^b + N_{neg,i}^b) \lambda^{i-1} + 2} \quad (2.7)$$

Next the public reputation of the seller is derived, based on an estimation of the probability that the seller will provide good service given the trustworthy advisors’ past experiences with s , taking into account b ’s own model of trustworthiness of each advisor a_j . First, the ratings are discounted based on the trustworthiness of the applicable advisor. Equations 2.8 and 2.9 may then be used to determine b ’s trust of ratings provided by each a_j .

$$D_{pos_i}^{a_j} = \frac{2Tr(a_j)N_{pos,i}^{a_j}}{(1 - Tr(a_j))(N_{pos,i}^{a_j} + N_{neg,i}^{a_j}) + 2} \quad (2.8)$$

$$D_{neg_i}^{a_j} = \frac{2Tr(a_j)N_{neg,i}^{a_j}}{(1 - Tr(a_j))(N_{pos,i}^{a_j} + N_{neg,i}^{a_j}) + 2} \quad (2.9)$$

The public reputation of s is itself calculated using Equation 2.10.

$$R_{pub}(s) = \frac{\left[\sum_{j=1}^k \sum_{i=1}^n D_{pos,i}^{a_j} \lambda^{i-1} \right] + 1}{\left[\sum_{j=1}^k \sum_{i=1}^n (D_{pos,i}^{a_j} + D_{neg,i}^{a_j}) \lambda^{i-1} \right] + 2} \quad (2.10)$$

Finally the overall trustworthiness of the seller s may be calculated with Equation 2.11.

$$w' = \begin{cases} \frac{N_{all}^b}{N_{min}} & \text{if } N_{all}^b < N_{min} \\ 1 & \text{otherwise} \end{cases} \quad (2.11)$$

$$Tr(s) = w' R_{pri}(s) + (1 - w') R_{pub}(s) \quad (2.12)$$

Note that N_{min} , the minimum number of ratings needed for the buyer b to be confident about the private reputation value it has of the seller s , is calculated according to Equation 2.4, but is not necessarily the same value used in Equation 2.5.

For greater clarity, we reproduce an edited version of the pseudo-code summary of the PTM's algorithm for modelling seller trustworthiness from [42] as Algorithm 1.

The model also includes an incentive mechanism, whereby honest advisors are rewarded by better offers from sellers, and in turn these sellers receive better reputations and ultimately more customers. While interesting in its own right, this mechanism does not directly affect our current work, and therefore we do not discuss this part of Zhang's model further.

We will provide an example scenario showing how the PTM is used in the context of our proposals, and their effects on the trust modelling in that scenario, in Section 3.2.1.

Algorithm 1 PTM Method for Buyer b 's Trustworthiness Model of a Seller s (From [42])

- 1: {Buyer estimates *private reputation* of seller based on buyer's own ratings}
 - 2: Set $N_{pos}^b = N_{neg}^b = 0$: amount of discounted positive/negative ratings of b ;
 - 3: $\{T_1, T_2, \dots, T_n\}$: time windows;
 - 4: **for all** T_i *in* $\{T_1, T_2, \dots, T_n\}$ **do**
 - 5: Set $N_{pos,i}^b = N_{neg,i}^b = 0$: number of b 's positive/negative ratings in T_i ;
 - 6: $R_{b,s}$: buyer b 's ratings for seller s ;
 - 7: **for all** $r_{b,s}$ *in* $R_{b,s}$ *and occurring during* T_i **do**
 - 8: **if** $r_{b,s} = 1$ **then**
 - 9: $N_{pos,i}^b = N_{pos,i}^b + 1$;
 - 10: **else**
 - 11: $N_{neg,i}^b = N_{neg,i}^b + 1$;
 - 12: **end if**
 - 13: **end for**
 - 14: $N_{pos}^b = N_{pos}^b + N_{pos,i}^b \lambda^{i-1}$; $N_{neg}^b = N_{neg}^b + N_{neg,i}^b \lambda^{i-1}$;
 - 15: **end for**
 - 16: *Private reputation* is then calculated using Equation 2.7;
 - 17: Calculate weight w' using Equations 2.4 and 2.11;
 - 18: Set *public reputation* = 0;
 - 19: **if** $w' < 1$ **then**
 - 20: {private knowledge is limited, buyer estimates *public reputation* of s } {based on advisors' ratings for the seller}
 - 21: $\{a_1, a_2, \dots, a_k\}$: trustworthy advisors that have provided ratings for seller s ;
 - 22: Set $N_{pos}^a = 0$: amount of all discounted positive ratings of advisors;
 - 23: Set $N_{neg}^a = 0$: amount of all discounted negative ratings of advisors;
 - 24: **for all** a_j *in* $\{a_1, a_2, \dots, a_k\}$ **do**
 - 25: Set $N_{pos}^{a_j} = 0$: amount of discounted positive ratings of a_j ;
 - 26: Set $N_{neg}^{a_j} = 0$: amount of discounted negative ratings of a_j ;
 - 27: **for all** T_i *in* $\{T_1, T_2, \dots, T_n\}$ **do**
 - 28: Count $N_{pos,i}^{a_j}, N_{neg,i}^{a_j}$: number of a_j 's positive/negative ratings in T_i ; {similar to the procedure of counting $N_{pos,i}^b$ and $N_{neg,i}^b$ }
 - 29: Set $D_{pos,i}^{a_j}$ based on $N_{pos,i}^{a_j}$ using Equation 2.8;
 - 30: Set $D_{neg,i}^{a_j}$ based on $N_{neg,i}^{a_j}$ using Equation 2.9;
 - 31: $N_{pos}^{a_j} = N_{pos}^{a_j} + D_{pos,i}^{a_j} \lambda^{i-1}$; $N_{neg}^{a_j} = N_{neg}^{a_j} + D_{neg,i}^{a_j} \lambda^{i-1}$;
 - 32: **end for**
 - 33: $N_{pos}^a = N_{pos}^a + N_{pos}^{a_j}$; $N_{neg}^a = N_{neg}^a + N_{neg}^{a_j}$;
 - 34: **end for**
 - 35: *Public reputation* is then calculated using Equation 2.10;
 - 36: **end if**
 - 37: *Trustworthiness* is then calculated using Equation 2.12
-

2.1.2 Other Trust Models

We also examine how our modifications will improve other trust models, mainly focusing on TRAVOS [32]. TRAVOS has some similarities to PTM, as in [44] – both take a probabilistic approach to the modelling of trust, using beta probability density functions (pdfs) – making it a good comparison to the results we will obtain for PTM.

The TRAVOS model was developed to (a) provide a trust metric, representing the level of trust in an agent, to be used to conclude whether a given agent is more or less trustworthy than another; (b) reflect an individual’s confidence in the level of trust it holds in another agent, in order to gauge the degree of influence which the aforementioned metric has on the decision to interact with another agent (or not); (c) ensure that an agent cannot assume that the opinions of others are necessarily accurate or based on real experiences.

Similar to the PTM, under the TRAVOS approach, it is assumed that a truster agent, a_{tr} , will not generally have complete information about a trustee agent, a_{te} , in order to definitively state the probability, $B_{a_{tr},a_{te}}$, that a_{te} will fulfill its obligations to a_{tr} . At most, we can calculate an expected value of this probability based on the set of interaction outcomes of the past interactions between the agents up to some time t , $O_{a_{tr},a_{te}}^{1:t}$. In other words, the level of trust $\tau_{a_{tr},a_{te}}$ is defined as:

$$\tau_{a_{tr},a_{te}} = E[B_{a_{tr},a_{te}} | O_{a_{tr},a_{te}}^{1:t}] \quad (2.13)$$

Determining this expected value requires a beta pdf to determine the relative probability that $B_{a_{tr},a_{te}}$ will take a certain value. The shape of the plotted pdf is normally given in terms of two parameters α and β , which leads to the following formulae:

$$\alpha = m_{a_{tr},a_{te}}^{1:t} + 1 \quad (2.14)$$

$$\beta = n_{a_{tr},a_{te}}^{1:t} + 1 \quad (2.15)$$

where t is the time of evaluation, $m_{a_{tr},a_{te}}^{1:t}$ is the number of successful interactions for a_{tr}

with a_{te} up to time t , and likewise $n_{a_{tr},a_{te}}^{1:t}$ is the number of unsuccessful interactions under the same scenario. This leads to the determination of the level of trust $\tau_{a_{tr},a_{te}}$ as:

$$\tau_{a_{tr},a_{te}} = E[B_{a_{tr},a_{te}}|\alpha, \beta] = \frac{\alpha}{\alpha + \beta} \quad (2.16)$$

Suppose, for example, that a buyer agent b_0 is attempting to determine how much it should trust some target seller agent s_0 given that, as of time t , b_0 has had four satisfactory purchases and three unsatisfactory purchases from s_0 . Under the TRAVOS model, this trust value would be computed as follows:

$$\begin{aligned} \alpha &= m_{b_0,s_0}^{1:t} + 1 = 4 + 1 = 5 \\ \beta &= n_{b_0,s_0}^{1:t} + 1 = 3 + 1 = 4 \end{aligned}$$

$$\tau_{b_0,s_0} = \frac{\alpha}{\alpha + \beta} = \frac{5}{5 + 4} = \frac{5}{9} = 0.\bar{5}$$

A separate metric is then determined such that a_{tr} may measure its confidence in this trust value, $\tau_{a_{tr},a_{te}}$, after choosing an acceptable margin of error ϵ . This metric, $\gamma_{a_{tr},a_{te}}$, represents the posterior probability that the true value of $B_{a_{tr},a_{te}}$ lies within the range $[\tau_{a_{tr},a_{te}} - \epsilon, \tau_{a_{tr},a_{te}} + \epsilon]$, and is calculated as a function of α , β , ϵ , and $\tau_{a_{tr},a_{te}}$.

If the confidence is not sufficiently high, the advice of third parties may be considered as well. This would be performed by asking other agents to report the number of successful and unsuccessful interactions that each has had with a_{te} , the aggregate of which are computed as $M_{a_{tr},a_{te}}$ and $N_{a_{tr},a_{te}}$ respectively. This in turn leads to a separate probability distribution, D^c , with $\alpha = M_{a_{tr},a_{te}} + 1$ and $\beta = N_{a_{tr},a_{te}} + 1$, which can then be used in Equation 2.16.

However, mindful of the possibility that some other agents may report untruthfully, TRAVOS also incorporates a mechanism to filter out the reports by agents which have low reputations. The first stage is to estimate the probability that an agent a_{op} 's reported

opinion about the trustee a_{te} , denoted $\hat{\mathcal{R}}_{a_{op},a_{te}}$, is accurate. This is performed by (i) constructing additional beta distributions, D^r derived from $\hat{\mathcal{R}}_{a_{op},a_{te}}$, and D^o which is based on the outcomes of the previous interactions for which a_{op} provided to a_{tr} an opinion similar to $\hat{\mathcal{R}}_{a_{op},a_{te}}$ about a_{te} or some other trustee, and (ii) finding their respective expected values, E^r and E^o . The range of possible values for both E^r and E^o , i.e. $[0, 1]$, is then divided into several disjoint intervals (bins) of equal size² to determine (in essence) whether E^r and E^o are located in the same bin, via the calculation of an accuracy value denoted as $\rho_{a_{tr},a_{op}}$.

Finally, TRAVOS attempts to reduce the effect of unreliable opinions on D^c through an approach that discounts high values of parameters, unless the probability of a rater’s opinion being accurate is very high. This is performed through the construction of another beta distribution, \bar{D} , based on (i) the accuracy value $\rho_{a_{tr},a_{op}}$, and (ii) the expected values and standard deviations of the uniform distribution ($\alpha = \beta = 1$) and of the distribution(s) of the unreliable opinion(s) that are sought to be removed, D^r .

We can identify three important distinctions between PTM and TRAVOS:

- PTM uses both private and public knowledge regarding all sellers, whereas TRAVOS uses only the private knowledge regarding some selected sellers.
- The method used by TRAVOS to aggregate ratings provided by certain advisors is more complex, reducing the effect of ratings from less trustworthy advisors using a method of filtering.
- TRAVOS reasons about the specific seller being considered when determining how much to trust an advisor. By contrast, in PTM, advisor reputation is calculated independently of any specific seller.

We will return to these distinctions in Section 3.3.2 in considering the differences in results between the two models when applying our approach.

²Teacy et al. specified that five intervals should be used. However, in our simulations in Section 3.3.2 and thereafter, we followed the selection in [42] of two bins as providing the best results.

2.2 Techniques for Limiting Network Size

In examining potential techniques for finding the most appropriate size of an advisor network in the aforementioned trust models, we drew some inspiration from techniques in other areas of artificial intelligence, more specifically from collaborative filtering. Before outlining these specific techniques, we first briefly explain the relevant background in this area.

A *recommender system* is a system which generates recommendations for its users based on prior inputs from its user base, often in the form of subjective opinions, about the items in question [27][26]. For example, a film recommender system might accept inputs from users about whether a particular film was good or bad, and then uses that information to recommend the best or most relevant selection to users who hadn't previously seen that film.

One of the earliest, if not the first, recommender system described in the literature was the Tapestry system [5], which suggested a means of recommending the most interesting e-mail messages to users, regardless of its source, based on (for example) whether the sender or a replier of a message was indicated on a pre-specified list. That work also introduced the term *collaborative filtering* (abbreviated CF), which refers to a subset of recommender systems in which users collaborate – or their information is used in a collaborative fashion – in order to generate recommendations. GroupLens [26] was a later CF mechanism that used a more implicit filtering mechanism, recommending newsgroup articles to a user if they were rated highly by other users similar to the first user, based on a comparison of the users' past ratings.

The research group that first proposed GroupLens discussed some of the design choices for collaborative filtering algorithms in [9]. It is from this work that we find two possible methods for limiting the number of advisors.

The first method, *correlation thresholding* [30], sets a minimum correlation weight that an advisor must have in order to be considered part of the user's "neighbourhood". However, if the threshold is set too high, then the neighbourhood may be very small, limiting the possibilities for predictions.

The second method discussed, *best-n-neighbors*, which is cited as being used in the GroupLens [26] system among others,³ picks a maximum number of neighbours to use, *max_nbors*. (The same process is also referred to in the literature as *k*-nearest neighbours, or *k*NN [18].) The neighbours chosen would be those with the highest correlation to the instant user.

In Chapter 3, we introduce these two methods into our proposed framework. In Section 5.2, we return to contrast our treatment of these techniques, in comparison with the efforts described in [9].

2.3 Referral Systems

A potential supplement to finding a more appropriate size for the advisor network is derived from Yu and Singh [39][40]. In [39] they discussed, as an idea for future work, reputation management in a social network making use of a *referral* mechanism. In this mechanism, a requesting agent (roughly equivalent to the buyer in our e-commerce scenario) would consult its “neighbour” agents, each of which might either provide advice on the question itself, provide references to other appropriate advisors, or both, depending on the question. As a result, a requesting agent would be able to benefit from the information held by the pool of agents without having a large number of neighbours [42].

A version of this mechanism is implemented in [40]: Each agent has a set of *acquaintances*, which were randomly determined at the outset, and did not change thereafter. For each of those acquaintances, each agent maintains a model of the acquaintance’s expertise (trustworthiness) and sociability (i.e., whether that acquaintance would itself be likely to refer to other trustworthy agents). A subset of the acquaintances representing the most trustworthy and sociable agents would be designated as *neighbours*, the composition of which might change from time to time. Each agent would also be assigned a “branching factor” specifying how many referrals from its set of neighbours it could provide at any one time.

³Notwithstanding this citation in [9], we were unable to find any further details about the use of this technique, or how it was applied, in our review of [26] and other papers from the GroupLens research group.

A requesting agent, A_r , might then desire to evaluate the trustworthiness of some target agent A_g . If A_g was already among A_r 's acquaintances, it would use its existing local trust model for A_g for its evaluation. Otherwise it would query its acquaintances for information about some A_g . If an acquaintance were also acquainted with a target agent, it would be accepted as a “witness”; else it would provide *referrals* to a subset of its neighbours (limited to the value of the branching factor). The requesting agent would then query these referred agents in turn, and so on until either a maximum depth of the trust network was reached or the desired number of witness agents had been found. Each witness would then provide some belief function computed using Dempster-Shafer theory, based on its local ratings of prior interactions with the target agent, indicating its trustworthiness of that agent. Additional steps would then be taken to determine whether each of those witnesses were themselves trustworthy or untrustworthy, and ultimately the belief functions from the remaining witnesses would be combined to compute the reputation of the target agent.

Yolum and Singh [37][38] expanded on the earlier work, in part by introducing the concept of *self-organizing* referral networks – that is, a referral network acting without external control, and adapting to take into account useful entities. Interestingly this work also discusses a “capability” metric that might be used to measure whether an agent would be likely to give a good answer for a given query, or alternatively whether an answer that has been given is a good answer to the given query. This would be calculated as follows:

$$Q \otimes E = \frac{\sum_{t=1}^n (q_t e_t)}{\sqrt{n \sum_{t=1}^n q_t^2}} \quad (2.17)$$

where $Q = \langle q_1 \dots q_n \rangle$ is a query vector derived from the requesting agent's interest vector (representing the agent's interest in receiving different services), $E = \langle e_1 \dots e_n \rangle$ is an expertise vector (which specifies the trustworthiness of the queried agent with respect to each of these services); and n is the number of dimensions that these vectors have.

This work also examines different *referral policies* that an agent might use – either referring all neighbours, referring all “matching” neighbours (those scoring above a given capability threshold), or referring only the best neighbour (that is, the one with the highest capability threshold).

A quite exhaustive examination of various design choices in referral systems is also contained in [38], some of the more interesting results being as follows:

- Requiring a high number of referrals will not guarantee a high-quality referral network; in some cases fewer referrals may be desired, such as for low capability thresholds.
- If agents value high sociability (as defined earlier in this section), then agents with similar interests are more likely to become neighbours; in some (but not all cases) this might improve the likelihood that suitable service providers (sellers) will be found.
- On the other hand, if agents place a higher priority on a high quality of service, then more strong authorities – that is, agents that are highly reliable, as in [25] – will emerge.

We finally note that the referral techniques discussed above have some similarity to the Repage system presented in [28] – a system that is more directly related to trust modelling. Similar to certain of the referral systems discussed above [40], Repage combines the requesting agent’s own evaluation, or *image*, of the target agent with the *reputation* of that agent, i.e. the requesting agent’s belief about the consensus evaluation regarding the target. The latter component is derived in large part from third-party agents, known as informers, which can transmit their own reported images of the target; units known as detectors are then responsible for determining which information will be most useful in evaluating the reputation of the target. Thus, the set of informers need not be static, much as the referral mechanism we will introduce in Chapter 4 will seek to find the advisors that are most experienced with a given target agent.

Based on this earlier work, it stands to reason that a similar *advisor referral* method could be used in combination with the limiting techniques discussed in the previous section in order to yield an overall smaller advisor network size. However, we will also explore the effects of our more principled approach to applying network limiting techniques – specifically, varying the size of the advisor network – alongside advisor referrals, whereas the size of the network and other parameters were kept constant in some of the earlier works, specifically [40][38].

We will introduce our proposal for integrating advisor referrals in Chapter 4. We then return to contrast our approach to those of other authors in Section 5.3.

Chapter 3

Limiting Advisor Network Size

3.1 Proposed Techniques

In the previous chapter we noted some past work examining the usefulness of setting a maximum number of advisors or correlation thresholding in collaborative filtering recommender systems [9]. We now wish to examine whether either method will lead to improvements in the accuracy of trust modelling. We hypothesize that both of these methods should be effective in that, although they will remove some information from the model, this information will originate from advisors that are themselves modelled as being less trustworthy than those that remain. As a result, we should be able to have greater confidence in the results computed with these methods, since they will be generated using information from the more trustworthy advisors, and not the others.

We note that the results from [9] (which we discuss in greater detail in Section 5.4) indicate that setting a maximum size to the advisor network performs better than thresholding in the collaborative filtering scenario, which in turn might suggest that it is likely to be the most effective approach for trust modelling as well. However, we cannot overlook the distinction between correlation for collaborative filtering and reputation. While similarity with a buyer may indirectly impact on that buyer's private reputation of an advisor, the private reputation of a seller only relates to the buyer's ratings for that seller, ignoring similarity, while similarity is not a factor at all in public reputation. Hence we

propose that both options, trustworthiness thresholding and maximum number of advisors, should be thoroughly examined. (We will return to the challenge of balancing similarity and reputation modelling in Section 5.4.)

That said, neither setting a maximum number of advisors nor using thresholding can be directly applied to the PTM’s computation of advisor reputation. For example, the public reputation component of an advisor’s reputation relies on the comparison of its interactions with those of other advisors. If we had some *a priori* information about these other advisors that could be used when computing the public reputation for a single advisor, then *max_nbor*s or thresholding might be useful in this computation. However, in PTM, this is not the case: the advisor trustworthiness values must be calculated for all possible advisors, using all of the available information, before the buyer can proceed to calculate seller reputation.

Our application of these techniques in the seller reputation model is formalized as follows.

3.1.1 Trustworthiness Thresholding

We first choose some threshold L ($0 \leq L \leq 1$) which represents the minimum advisor trustworthiness value $Tr(a)$ required for an agent to be included in the advisor network. We then define the set $A_{L,b} = \{a_1, a_2, \dots, a_k\}$ consisting of all advisors for which $Tr(a) \geq L$ for a particular buyer b . We then use the subset $A_{L,b,s}$, consisting of the advisors in $A_{L,b}$ that have provided ratings for the seller s , in place of the previously-defined set $\{a_1, \dots, a_k\}$, the set of all advisors that have provided ratings for s , in the PTM seller reputation algorithm (reproduced in Section 2.1.1 as Algorithm 1).

3.1.2 Maximum Number of Advisors

For a particular buyer b , after having calculated the personalized trustworthiness of each advisor for b as per the first part of the PTM, we sort the list of all n advisors from greatest trustworthiness value to least, in the set $\{a_1, a_2, \dots, a_n\}$. We choose some maximum

number of advisors for each buyer, $max_nbors \leq n$, and then truncate this set to the set $A_b = \{a_1, a_2, \dots, a_{max_nbors}\}$. We thus obtain the set of max_nbors advisors that have been calculated to be the most trustworthy for b . Again, the subset of A_b that has provided ratings for the seller s is used in place of the larger set $\{a_1, \dots, a_k\}$ in the PTM seller reputation algorithm.

3.2 Examples

3.2.1 Using the PTM

As in [42], we consider the case where a buyer b wishes to assess the trustworthiness of a particular seller s_0 with whom the buyer has had little or no experience. For the purposes of this simplified example, we assume that there are four available advisors from which b may seek advice, namely a_w , a_x , a_y , and a_z .

We assume initially that, among sellers that b has had past dealings with, each of these advisors has provided ratings only for the five sellers $(s_1, s_2, s_3, s_4, s_5)$, and has rated each of the sellers at most once in each time window in the sequence T , where T_1 is the most recent time window. The ratings may be either positive (1) or negative (0); a dash (-) indicates that no rating was provided during the indicated time window. The ratings provided by each advisor for these sellers are listed in Table 3.1. The buyer b has also provided some ratings for the sellers, as indicated in the same table; note here that b does not provide ratings for every seller each time window.

We derive the trustworthiness values for each advisor using Equations 2.2 through 2.6. For simplicity, in these calculations, we follow the method used in the examples provided in [42]. First, we will only consider pairs of ratings provided during the same time window. We thus assume that the forgetting factor as defined previously is $\lambda = 0$. Hence Equation 2.1 will yield $z = 1$ for all pairs of ratings occurring within the same time window, and $z = 0$ otherwise. For the determination of N_c , we assume for simplicity that any rating of 1 provided by the advisor is a “consistent” rating, meaning a rating that matches the majority opinion of other advisors for a particular seller that received a rating. Finally, in

Table 3.1: Ratings of Sellers Provided by Advisors and Buyer b

	a_w					a_x					a_y				
T	T_1	T_2	T_3	T_4	T_5	T_1	T_2	T_3	T_4	T_5	T_1	T_2	T_3	T_4	T_5
s_1	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0
s_2	0	0	1	1	1	1	1	1	1	1	0	1	1	0	1
s_3	0	1	1	1	1	1	1	1	1	1	1	0	0	0	1
s_4	0	0	0	1	1	1	1	1	1	1	0	1	0	1	0
s_5	0	1	1	1	1	1	1	1	1	1	1	0	0	0	1

	a_z					b				
T	T_1	T_2	T_3	T_4	T_5	T_1	T_2	T_3	T_4	T_5
s_1	0	0	0	0	0	1	1	1	1	1
s_2	0	0	0	0	0	1	1	1	1	-
s_3	0	0	0	0	0	1	1	1	-	-
s_4	0	0	0	0	0	1	1	-	-	-
s_5	0	0	0	0	0	1	-	-	-	-

Equation 2.4 we use $\gamma = 0.8$ and $\epsilon = 0.15$, leading to $N_{min} = 51$. The pertinent values are shown in Table 3.2.

We proceed to the calculation of the trustworthiness of a seller s_0 . As a preliminary matter, we remember that the buyer b has not provided any ratings in the past for s_0 , and therefore $R_{pri}(s) = \frac{1}{2}$. Of our four advisors, only a_w , a_x and a_z have provided ratings for the seller s_0 , as indicated in Table 3.3a. The subsequent Table 3.3b indicates how these ratings translate into positive and negative amounts, while Table 3.3c shows how these ratings are discounted based on the advisor trustworthiness values calculated earlier.

Using Equation 2.10, we may then find the public reputation of s_0 . In keeping with the examples provided in [42], we remove our previously-stated simplification that only compared ratings in the same time window, and thus set a forgetting factor of $\lambda = 0.9$:

Table 3.2: Trustworthiness of Advisors a_w , a_x , a_y , and a_z for Buyer b

a_j	N_p	N_{all}	α	β	R_{pri}	N_c	N'_{all}	α'	β'	R_{pub}	w	$Tr(a)$
a_w	5	15	6	11	0.353	14	25	15	12	0.556	0.294	0.497
a_x	15	15	16	1	0.941	25	25	26	1	0.963	0.294	0.957
a_y	8	15	19	8	0.529	11	25	12	15	0.444	0.294	0.469
a_z	0	15	1	16	0.059	0	25	1	26	0.037	0.294	0.0434

Table 3.3: Ratings of s_0 Provided by a_w , a_x , a_z

(a) Ratings						(b) Amounts of Ratings					
T_i	T_1	T_2	T_3	T_4	T_5	T_i	T_1	T_2	T_3	T_4	T_5
a_w	1	0	1	0	1	$N_{pos,i}^{a_w}$	1	0	1	0	1
a_x	0	0	0	1	1	$N_{neg,i}^{a_w}$	0	1	0	1	0
a_z	1	1	1	1	1	$N_{pos,i}^{a_x}$	0	0	0	1	1
						$N_{neg,i}^{a_x}$	1	1	1	0	0
						$N_{pos,i}^{a_z}$	1	1	1	1	1
						$N_{neg,i}^{a_z}$	0	0	0	0	0

(c) Discounted Amounts of Ratings					
T_i	T_1	T_2	T_3	T_4	T_5
$D_{pos,i}^{a_w}$	0.397	0	0.397	0	0.397
$D_{neg,i}^{a_w}$	0	0.397	0	0.397	0
$D_{pos,i}^{a_x}$	0	0	0	0.937	0.937
$D_{neg,i}^{a_x}$	0.937	0.937	0.937	0	0
$D_{pos,i}^{a_z}$	0.0294	0.0294	0.0294	0.0294	0.0294
$D_{neg,i}^{a_z}$	0	0	0	0	0

Table 3.4: Advisor Network Size with a Correlation Threshold

L	0	0.2	0.4	0.6	0.8	1
$A_{L,b}$	$\{a_w, a_x, a_y, a_z\}$	$\{a_w, a_x, a_y\}$	$\{a_w, a_x, a_y\}$	$\{a_x\}$	$\{a_x\}$	$\{\}$
$ A_{L,b} $	4	3	3	1	1	0

$$R_{pub}(s_0) = \frac{\sum_{i=4}^5 0.937 * 0.9^{i-1} + 0.397 * (0.9^0 + 0.9^2 + 0.9^4) + \sum_{i=1}^5 0.0294 * 0.9^{i-1} + 1}{\sum_{i=1}^5 0.937 * 0.9^{i-1} + \sum_{i=1}^5 0.397 * 0.9^{i-1} \sum_{i=1}^5 0.0294 * 0.9^{i-1} + 2} = 0.4480$$

Finally, since the buyer has not dealt with s_0 before, the weight for the private reputation w' is zero, meaning we can immediately conclude that $Tr(s_0) = 0.4480$.

3.2.2 Reputation Thresholding

We now turn to exploring the effects of the modifications proposed in Section 3.1. We first examine how setting a minimum reputation threshold would affect the size of our network and, in turn, the computation of the private reputation component of seller trustworthiness. We choose several potential values for the threshold L and indicate, based on the results in the previous section regarding advisor trustworthiness, how many advisors would be included in the buyer b 's advisor network in this case. The results are shown in Table 3.4.

Trivially, when $L = 0$, all advisors will be included in the network, and $Tr(s_0) = 0.4480$ – the same value as calculated at the end of the previous subsection. For $L = 0.2$ and $L = 0.4$, the advisor network consists of a_w , a_x , and a_y , of which only a_w and a_x contribute ratings for s_0 . We refer to the resulting trustworthiness value as $Tr_{w,x}(s_0)$, which we calculate using Equation 2.10 to be 0.439.

For $L = 0.6$ and $L = 0.8$, the advisor network consists solely of a_x , and therefore the

Table 3.5: Trustworthiness of s_0 Using a Maximum Number of Advisors

max_nbors	0	1	2	3	≥ 4
A_b	$\{\}$	$\{a_x\}$	$\{a_w, a_x\}$	$\{a_w, a_x, a_y\}$	$\{a_w, a_x, a_y, a_z\}$
$min(Tr(a))$	undefined	0.957	0.497	0.469	0.0434
$Tr(s_0)$	0.5	0.697	0.439	0.439	0.4480

seller trustworthiness $Tr_x(s_0)$ (again by Equation 2.10) would be 0.697. Finally, for $L = 1$, the advisor network is the empty set and, trivially, $Tr_{empty}(s_0) = \frac{1}{2}$.

3.2.3 Maximum Number of Advisors

We might elect to use a maximum number of advisors in place of the thresholding procedure examined in the previous subsection, so we now look at how the determination of seller trustworthiness would be affected in this case. Depending on the max_nbors value chosen, we would have the results shown in table 3.5, with the advisor network representing the max_nbors advisors most trusted by the buyer b . For comparison, we indicate the minimum trustworthiness value of the advisors in the network, to show the maximum threshold L that could be used to get the same result using thresholding.

3.3 Experimental Results

To verify that these modifications provide an improvement over the original PTM, we now turn to evaluating the performance of each of these changes. Some of this evaluation is inspired by the methods used in [43] to show the effectiveness of the original model.

We pause briefly to note some background details about the experimental setup for the simulations in this thesis. The necessary software written to simulate the scenarios documented in this thesis was written in Java using standard Java libraries. Much of the code was reused from the similar experiments conducted by Zhang and documented in [42], with appropriate changes in order to apply the techniques described earlier in this chapter,

as well as the referral approach that will be described in Chapter 4. The simulations were primarily run, in sequence, on a dual-core workstation computer.

However, for scenarios involving large populations (which required significantly more memory and computation time), the corresponding simulations were run in parallel on additional research servers using single-core processors. These large-population scenarios also generally required a significant amount of memory in order to allow the buyer to model the trust of every other advisor agent; this required adjustments to the standard size of the Java Virtual Machine’s heap, using up to 1 GB of random access memory per simulation.

Overall we estimate that approximately 250–350 hours of processing time were required to complete the experiments documented in this thesis.

3.3.1 Validating Effectiveness

We first verify that each of our modifications to PTM maintains the effectiveness of the trust model – that is, the new models still accurately reflect the trustworthiness of agents in the system. We first reiterate that these modifications do not in any way affect how individual buyer agents model the trustworthiness of their potential advisors. As noted at the beginning of this chapter, the *max_nbor*s and thresholding approaches are not appropriate to use as part of the PTM’s method for modelling advisor trustworthiness, but only as a means of selecting advisors once this model has been constructed. Thus any application of either approach will only affect *which* advisors are used once the advisor reputations have already been determined, and in turn the computations of the trustworthiness of sellers, but will not retroactively affect advisor trustworthiness.

Hence, to show that the modified trust model remains effective in modelling seller trustworthiness, we do not need to concern ourselves with the computation of advisor trustworthiness, but instead confirm, as in [43], that in the modified models, a decrease in the honesty of a seller corresponds to a decrease in the trustworthiness value calculated for that seller – for example, we expect that a seller that is dishonest 20% of the time should be modelled as being more trustworthy as one that is dishonest 40% of the time, and both should be modelled as having higher reputations as a seller that lies 60% of the time.

Note that at this stage we are not overly concerned with the accuracy of the calculated reputation values, so long as the computed trustworthiness is at least roughly proportional to the actual honesty of the seller.

This property was shown in [43] by simulating an environment consisting of one buyer, 80 advisors, and 100 sellers, where the sellers are evenly divided into ten groups, each having a probability of dishonesty between zero and 0.9. The buyer and the advisors each randomly select 80 of the sellers and rate each of those sellers. Finally, given these ratings, the buyer calculates the trustworthiness values corresponding to each of the sellers using Algorithm 1 (see Section 2.1.1). These tests are performed for two values of the percentage of lying (dishonest) advisors, specifically 30% and 60%, and repeated a total of ten times for each possible combination. The results indicated that PTM (combining public and private reputation) does reflect, relatively accurately, the actual trust of each seller in these scenarios.

We repeat these conditions for our simulation, using a derivative of the modelling and simulation software originally written by Zhang for [42] and related work, and now apply our techniques for limiting the size of the network. The results of these simulations under various combinations of these modifications are shown in Figure 3.1 for simulations where 30% of advisors are lying, and Figure 3.2 for scenarios where the percentage of lying advisors is 60%. In each of these graphs, the x -axis represents the predetermined probability of dishonesty for each category of sellers. The y -axis represents the average (mean) of the trust values, calculated using Algorithm 1 (as modified as discussed in Section 3.1.1 or 3.1.2, as applicable), averaged over all repetitions and all of the ten sellers in each category.

We stated in Chapter 1 that we consider trust to be an agent's belief that some other agent will carry out the tasks it says it will perform. We consider that an ideal trust model under this definition would represent trust as the proportion of interactions in which an agent does carry out the promised tasks. For example, a seller that is modelled as likely to not deliver the sold goods for four out of every ten transactions would ideally be assigned a trust value of $1 - 0.4 = 0.6$. In Figures 3.1 and 3.2, this model is represented graphically by a slope of -1 starting at (0, 1). Ultimately, we seek one or more variants that produce

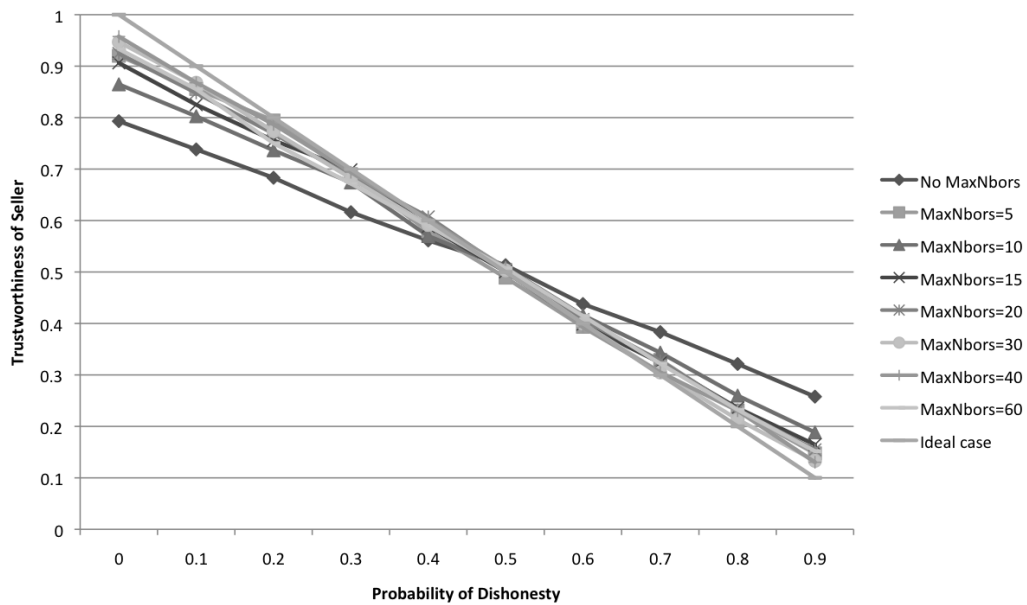
trust values very close to these “ideal” values.

We see that, in general, using thresholding and *max_nbor_s* does yield a trust model that tracks well with the ideal trust values, for both the 30% and 60% lying advisors (LA) cases. However, the error – the difference between the actual trust values and the ideal figures – varies depending on the parameters used, particularly the specific value of the threshold or *max_nbor_s*.

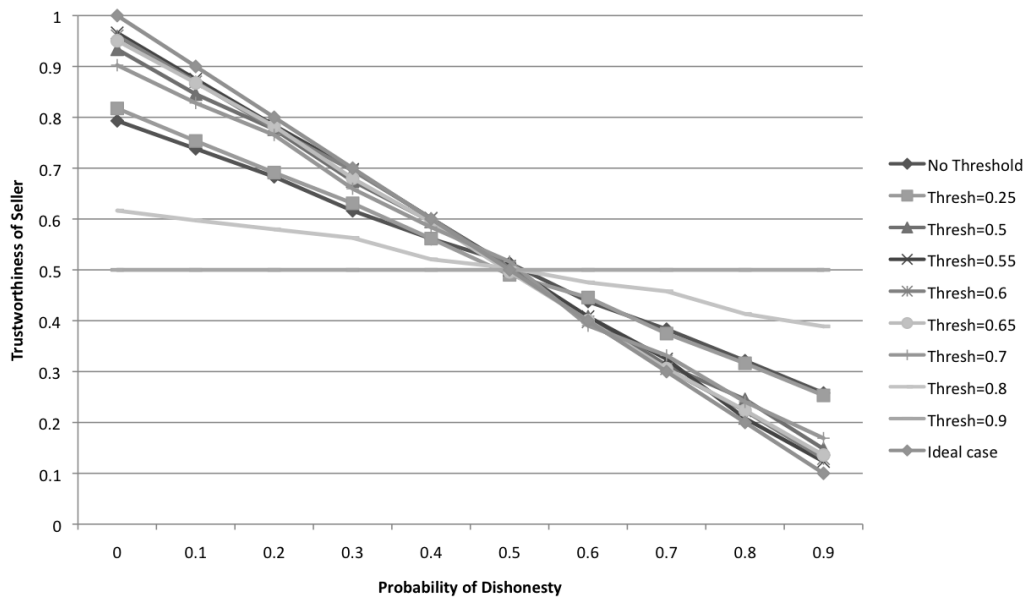
Figure 3.1a compares the performance of an unrestricted advisor network against several possible *max_nbor_s* parameters, for the case where 30% of the advisors are dishonest. All of the variants using *max_nbor_s* provided better results compared to not using any network-limiting method at all; that is, the plotted results all came closer to the “ideal” slope than for the unrestricted network. However, as expected, the best results came not from choosing extreme parameters (i.e. a very high or very low value for *max_nbor_s*), but rather from somewhere in the middle: the results that best matched the ideal trust model came for *max_nbor_s* = 40 (that is, the top 50% of the 80 advisors in the network). However, performance did not suffer significantly for smaller values, up to *max_nbor_s* = 15 (i.e. the top 19%).

A similar comparison for different threshold values is shown in Figure 3.1b. Here the results differed slightly: using a threshold between 0.5 and 0.7 improved the results compared to a network that had not been thresholded, with a threshold of 0.55 showing the greatest improvement. However, for thresholds of 0.8 and 0.9, the accuracy was reduced dramatically, as shown by their near-horizontal and horizontal graphs, respectively, on this figure. This is because very few (if any) advisors, apart from the buyer itself, would have a trust value above such a high threshold. The 0.9 threshold graph demonstrates the worst case where *no* suitable advisors could be found. In this case, in computing the public trustworthiness of sellers using Equation 2.10, the sum components will be zero, therefore the PTM would default to assigning each seller a trustworthiness of $\frac{0+1}{0+2} = 0.5$. The implication is that a threshold of 0.9 is simply an unrealistically high value to use to limit the advisor network in PTM.

Figures 3.2a and 3.2b cover a separate but very similar set of simulations covering the 60% lying advisors case. As can be seen by looking at the “No MaxNbor_s / No Threshold”

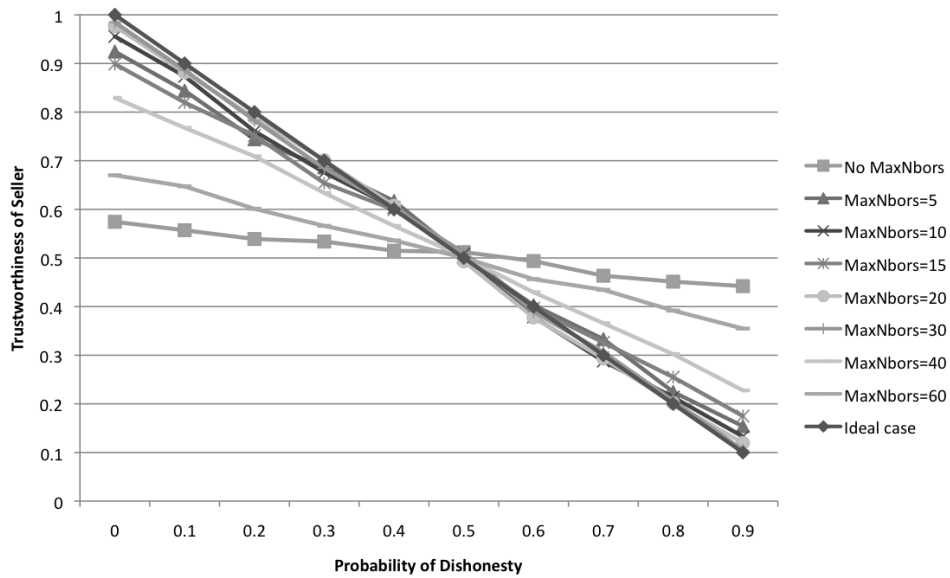


(a) Comparison of *max_nbors* approaches (30% LA)

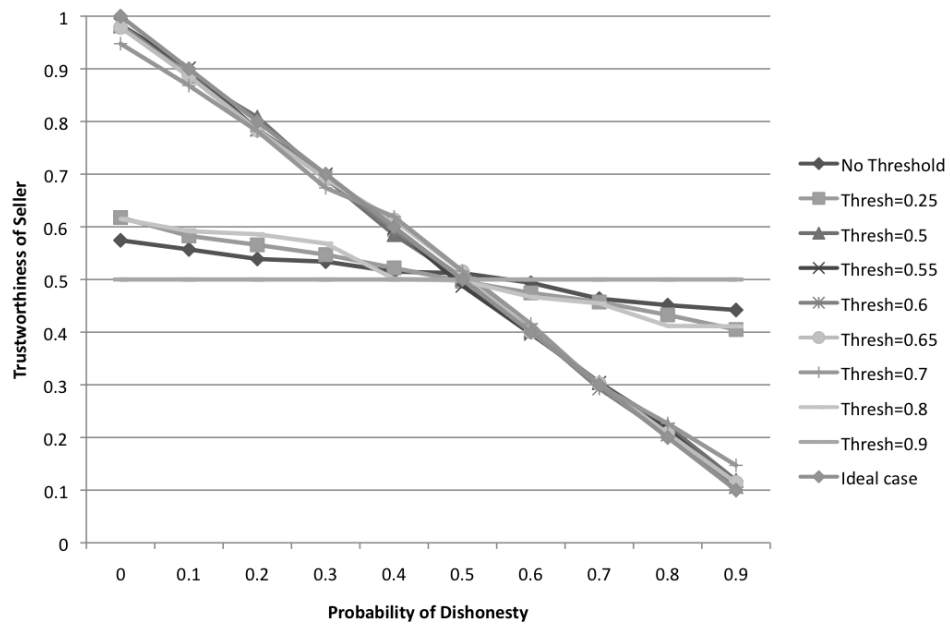


(b) Comparison of thresholding approaches (30% LA)

Figure 3.1: Verification testing for the modifications using 30% lying advisors.



(a) Comparison of *max_nbors* approaches (60% LA)



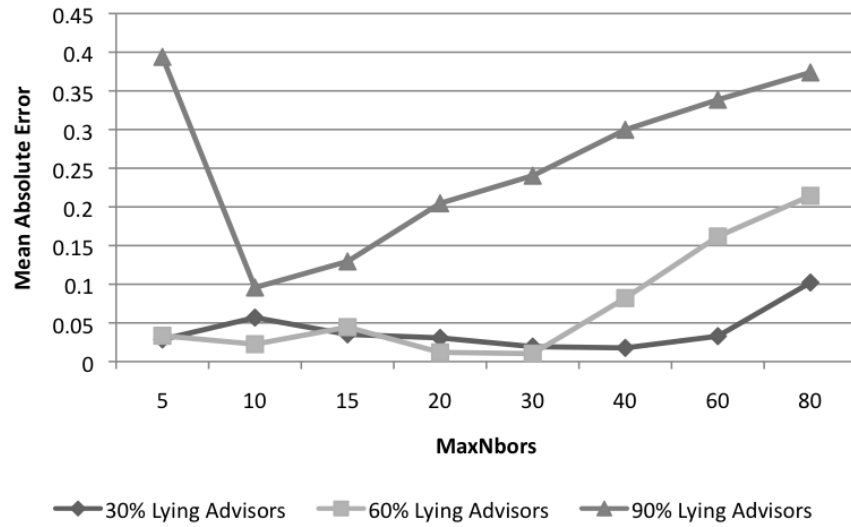
(b) Comparison of thresholding approaches (60% LA)

Figure 3.2: Verification testing for the modifications using 60% lying advisors.

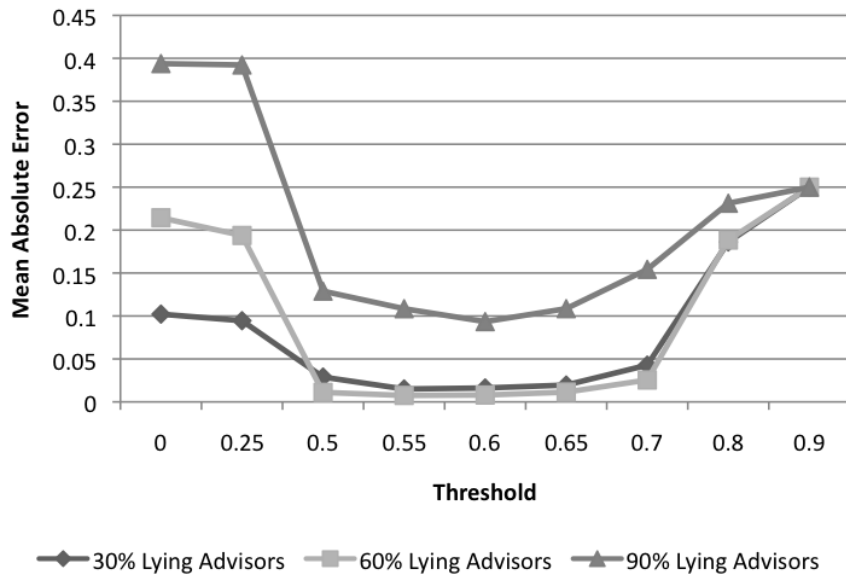
graphs, which are very far from the ideal-case graph shown in both figures, not applying *max_nhors* or thresholding at all will result in poor accuracy for the trust model. However, the accuracy can be significantly improved by applying either *max_nhors* or thresholding, with our experiments indicating 0.55 as the best threshold value, and *max_nhors* = 30 as the best among the tested *max_nhors* options.

A summary of the simulation results for both 30% and 60% lying advisors is provided in Figure 3.3. In these figures we also provide results for a similar set of simulations when the fraction of lying advisors is increased to 90%. This perspective may indeed provide a more intuitive comparison of the results compared to those in Figures 3.1 and 3.2. In both cases, each graph represents one of the tested percentages of lying advisors (either 30%, 60%, or 90%). The *x*-axis indicates the parameter chosen, if any, for *max_nhors* (in Figure 3.3a) or the trustworthiness threshold (in Figure 3.3b). The *y*-axis shows the mean absolute error (MAE) associated with that particular simulation – in other words, the average absolute difference between the “ideal” trust model as discussed above (error = 0), and the actual results for the variant measured. If the predicted trust values are very close to the actual values, the MAE will be low; if these values are far apart, the MAE will be high. The case where no *max_nhors* value is used – that is, all 80 advisors are included regardless – is represented by the far right of the graphs in Figure 3.3a. In Figure 3.3b, the equivalent case where no thresholding is applied is represented by a threshold of zero, at the far left of the graph.

We point out that an MAE of 0.25 – such as seen for a threshold of 0.9 in Figure 3.3b – represents a special value in these graphs, representing the largest MAE that should be expected in our simulations if the model is accurately classifying sellers as either trustworthy or untrustworthy. We noted above that in the “worst case” thresholding scenario where *no* suitable advisors can be found, the trust model will assign each seller a default value of 0.5. Taking account of all 100 sellers, each having been assigned a percentage of dishonest behaviour between 0 (which would correspond to 100% trustworthiness) and 90% (which would indicate 10% trustworthiness), the MAE in this case would be computed as follows:



(a) Error produced in *max_nbars* approaches



(b) Error produced in thresholding approaches

Figure 3.3: Mean average error of various trust model variants

$$\frac{10abs(1 - 0.5) + 10abs(0.9 - 0.5) + 10abs(0.8 - 0.5) + \dots + 10abs(0.1 - 0.5)}{100} = 0.25$$

It is possible to have an MAE above 0.25, as seen in (for example) certain of the 90% lying advisor cases; this would only occur if for some reason the system is modelling some of the sellers as trustworthy when they are in fact untrustworthy, and vice versa.

From Figure 3.3a, we can also see the *max_nbor*s approach may be affected by the percentage of lying advisors. Specifically, setting *max_nbor*s = 40 when 60% of the advisors are dishonest yields significantly worse performance than when 30% of the advisors are lying. On the other hand, using *max_nbor*s = 30 yields similar performance results when either 30% or 60% of advisors are lying. If 90% of advisors are lying, however, *max_nbor*s = 30 yields poor accuracy, whereas the best performance is found by setting *max_nbor*s = 10. This result suggests that when more advisors are lying, it is better to set a smaller value for *max_nbor*s. However, from Figure 3.3b we see that the thresholding approach is not heavily affected by the percentage of lying advisors. Even though there is a significant reduction in accuracy when moving from 60% to 90% lying advisors, the general shape of the graph (and therefore the best choices for the threshold) are largely unchanged. This is somewhat expected since thresholding allows only the trustworthy advisors to be included in buyers' networks.

Note that to simplify the presentation of our remaining results, we will use “summary” graphs similar to those in Figures 3.3a and 3.3b throughout the remainder of this chapter, and again in parts of Chapter 4. We believe it should be clear from our initial results (Figures 3.1 and 3.2) that the MAE is sufficient to determine how well the computed trust values track with the expected values.

We noted in Section 3.2.1, that there is a relationship between *max_nbor*s and threshold parameters, in that there will be values for both methods that will yield the same network composition. We will investigate this relationship further in Appendix A.

3.3.2 Applicability to Alternative Model

Having shown that our optimizations¹ have proven useful to PTM, we now turn to demonstrating that these changes are also effective for other trust models. Specifically we look to show that the TRAVOS model [32] can be improved by applying our modifications.

We discussed TRAVOS in detail in Section 2.1.2 of this thesis. We will nevertheless take this opportunity to reiterate the relationship between the two models. TRAVOS has some similarities to PTM, as in [44] – both take a probabilistic approach to the modelling of trust, using beta probability density functions (pdfs) – making it a good comparison to the results above.

However, there are three important distinctions between the models: One, PTM uses both private and public knowledge regarding all sellers, whereas TRAVOS uses only the private knowledge about certain selected sellers. Two, the method used by TRAVOS to aggregate ratings provided by certain advisors is more complex, which serves to reduce the effect of ratings from less trustworthy advisors.

Third, the TRAVOS model of the trustworthiness of a particular advisor is specific to the seller being considered. This means that the buyer (and each advisor) should construct a separate advisor network for *each* seller in the system. As we will discuss in Section 6.2.4, this may have certain negative impacts on the memory usage of such a system.

To demonstrate the effectiveness of our optimizations, we perform similar sets of experiments to those performed in Section 3.3.1; again, we used a modified version of the simulation and modelling software developed for the earlier work by Zhang. These simulations for the *max_nbor*s and thresholding optimizations use an environment consisting of one buyer, 80 advisors, and 100 sellers with varying probabilities of dishonesty. During the simulation, the buyer and each advisor both randomly select and rate a total of 80 sellers. Finally, the buyer calculates the trustworthiness values corresponding to each of the sellers. These tests are performed for two values of the percentage of lying advisors, 30% and 60%. The results of these experiments are shown in Figures 3.4a and 3.4b. Each

¹In this thesis we occasionally refer to our proposed techniques as “optimizations”. This is meant to indicate that we are using these techniques to improve the accuracy of trust modelling. We do not claim that these results are “optimal” in the sense of no further improvements being possible.

figure shows two graphs, indicating how each model performs for both of the tested levels of lying advisors; as with the graphs shown previously, the data points map the applicable *max_nbor*s or threshold parameter on the *x*-axis to the mean absolute error (MAE) of the trust model under that scenario on the *y*-axis.

These figures indicate mixed results with regards to the effect of applying these modifications to TRAVOS. Consider that an unrestricted network (as represented by the far right of Figure 3.4a, or the far left of Figure 3.4b) will yield a mean absolute error value between 0.15 and 0.25. Recalling our discussion of the meaning of an MAE of 0.25 in the previous section, it should be clear that these MAE values indicate relatively low accuracy. In comparison, most of the models incorporating either *max_nbor*s or a threshold will have a smaller error value, and thus improved accuracy over an unrestricted network.

However, the progression is not entirely consistent. For example, in examining both Figures 3.4a and 3.4b, the graphs representing the TRAVOS model have a zig-zag shape, with the MAE increasing and decreasing at various points in a somewhat haphazard fashion. This is particularly true for the thresholding cases, where the changes in the MAE seem to be fairly random.

For the *max_nbor*s cases (Figure 3.4a), there is less randomness in the MAE, but it is still surprising that the error decreases between *max_nbor*s = 40 and *max_nbor*s = 60, given that error had increased as *max_nbor*s was increased from 20 to 40.

The implication of both results is that under certain circumstances, decreasing the threshold (or increasing the *max_nbor*s parameter) serves to add additional advisors into the network which serve to reduce the trust modelling error – despite the fact that those agents would have been modelled as “less trustworthy” than those included in the more restricted network. In other words, some advisors seem to have been modelled as more trustworthy (or less trustworthy) than they should have been. The most likely apparent cause of this fault would be with the TRAVOS approach itself, but we cannot say conclusively that this is the case based on these results, and further examination of this theory is raised for future work in Section 6.2.3.

Nevertheless, the results reinforce the value of our proposed approach, to set an effective value for *max_nbor*s or thresholding, shown here through experimental methods. To

this end, we are able to say that in general, TRAVOS works best in this scenario when *max_nbars* is set to 20, or when a threshold of 0.5 is set.

3.3.3 Effects for Larger Populations

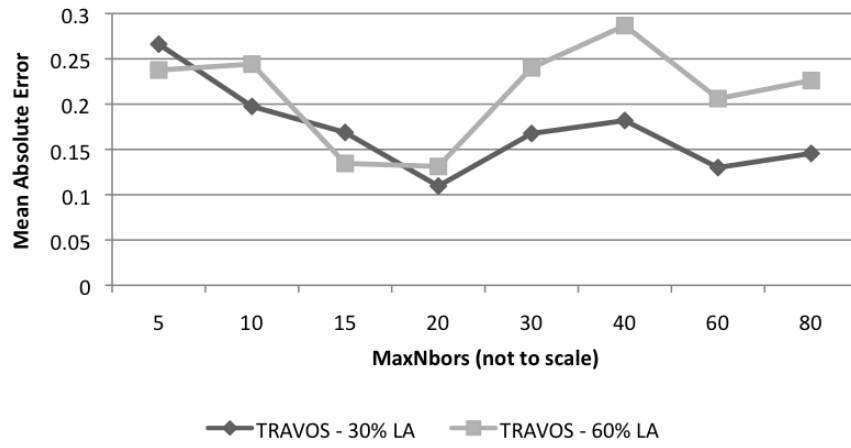
Given that most existing e-commerce systems have very large populations of users – on the order of thousands or even millions – we felt it would also be useful to demonstrate that our techniques also work with larger populations of advisors. Assuming this to be true, we would also observe the results and any distinctions from the earlier results involving smaller advisor populations.

We perform these simulations using PTM for an advisor population size of 500, which we feel is a useful starting point for considering systems with populations of this magnitude.² These tests otherwise maintain the same test conditions used for our earlier tests in Section 3.3.1 in terms of the number of sellers and the duration of the simulation. Again, however, we run the simulations with two values of the percentage of lying advisors (LA) – 30% and 60% – and with several values of *max_nbars* and thresholds.

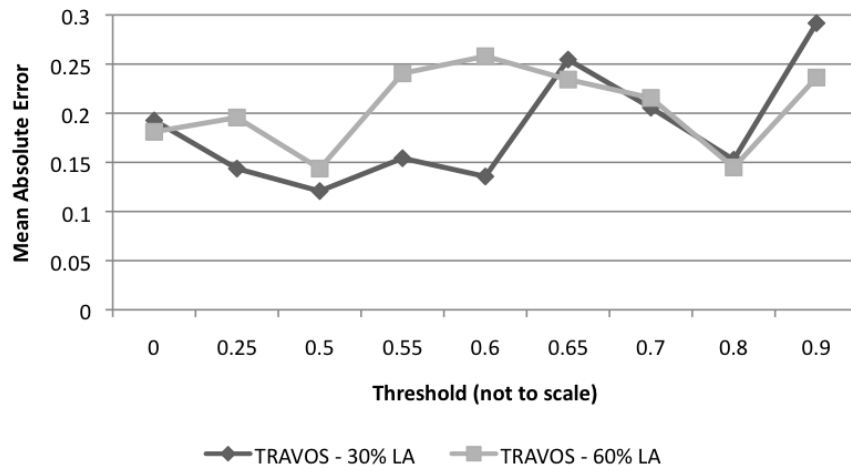
The results of these simulations, in terms of the mean absolute error of the trust model under each simulation as plotted against the *max_nbars* or threshold value used, are indicated in Figures 3.5a and 3.5b respectively.

It seems clear in comparing the results from Figure 3.5b to those in Figure 3.3b that the results of applying different threshold values is reasonably consistent despite the change in advisor population. More precisely, we mean that for both populations, the MAE is quite high for thresholds below 0.5 – approximately 0.1 for the 30% LA case, and approximately 0.2 for the 60% LA scenarios – but then decreases sharply, to below 0.03, as the threshold is increased to 0.5, then remains at a similarly low value as the threshold increased further, up to a threshold of 0.7. The MAE then climbs sharply again for both populations as the threshold is increased to 0.8 and then to 0.9. This similarity notwithstanding, we observe that, when all other parameters are the same, the MAE is still slightly lower for

²We reiterate that this only reflects the total number of users in the system, not the number that will be used in the advisor network, which we will consider next.

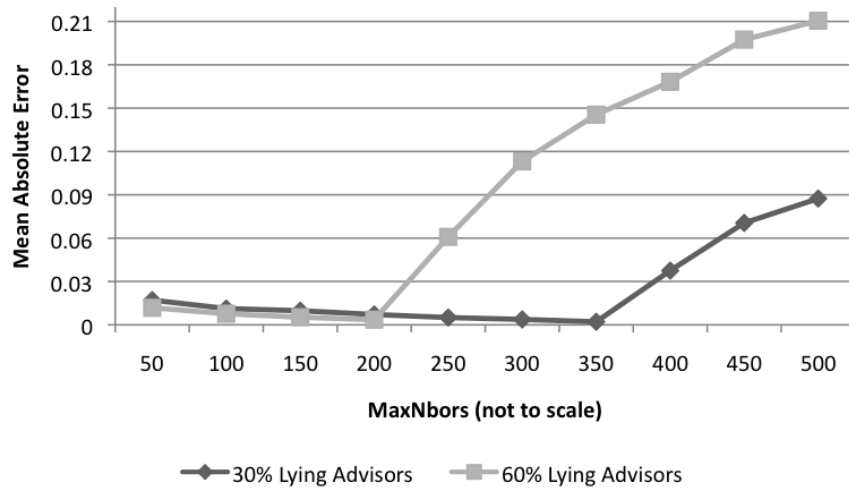


(a) Varying *max_nbars*

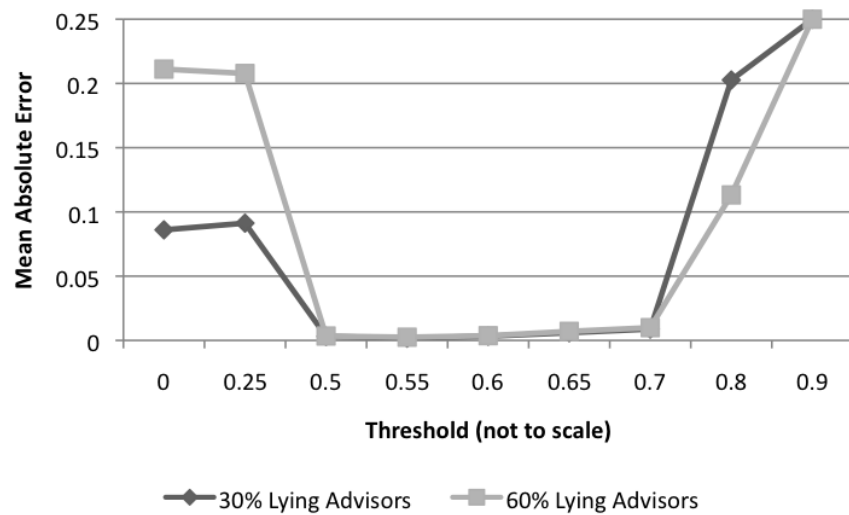


(b) Varying threshold

Figure 3.4: Mean absolute error when applying optimizations to TRAVOS at 30% and 60% LA



(a) Comparison when varying *max_nbars*



(b) Comparison when varying threshold

Figure 3.5: Mean absolute error applying optimizations to PTM with advisor population of 500

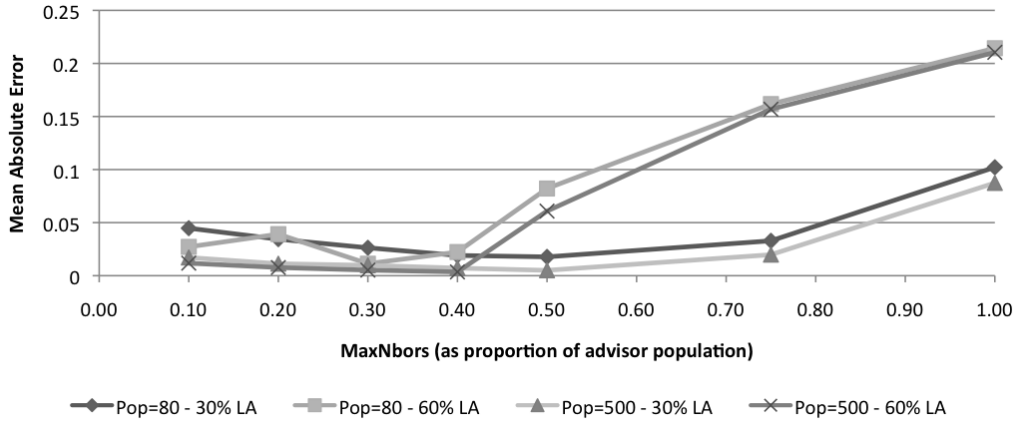


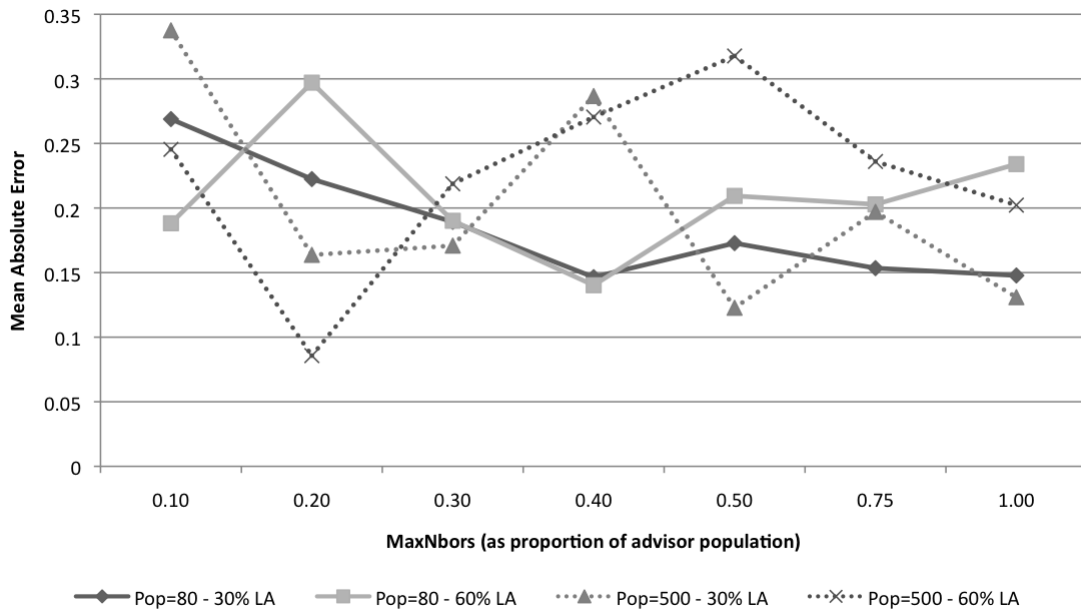
Figure 3.6: Comparison of mean absolute error when varying max_nbors (as proportion of advisor population)

the larger population. We believe this is simply because the larger population means that more highly-trusted advisors are available, and thus the system can make use of more information about each of the sellers, regardless of the threshold applied.

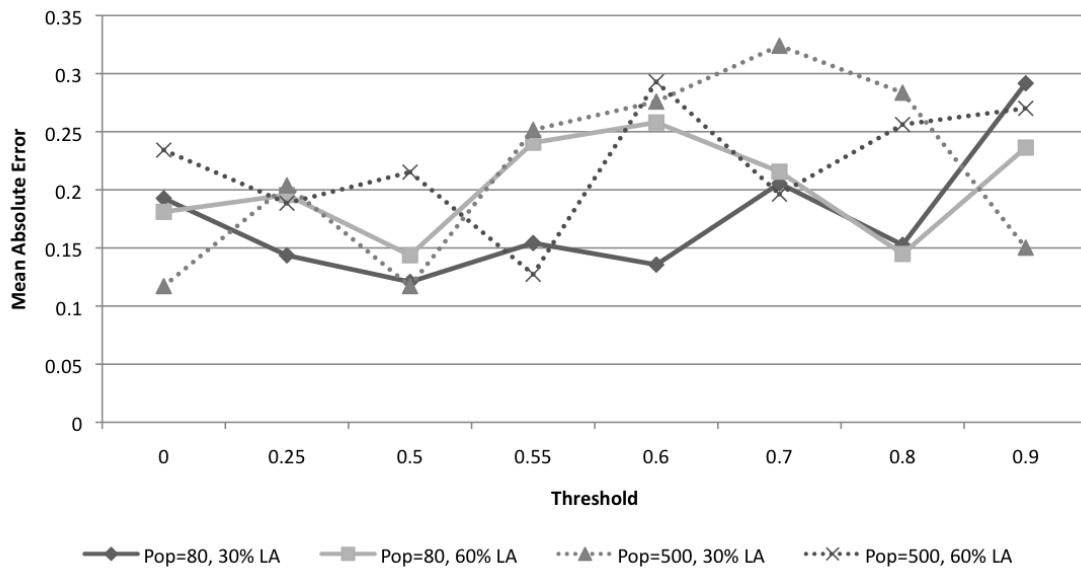
On the other hand, comparing the corresponding max_nbors tests, as shown in Figures 3.5a and 3.3a, is somewhat trickier. It is clear that for both of the tested advisor populations, setting some max_nbors value that is somewhat less than half of the advisor population size will result in a reduction in trust modelling error. However, while a value of max_nbors of 30 is optimal for an advisor population of 80, the optimal value when the total population is 500 is much larger, at about 200 (which suggests that the value should not simply be set in absolute terms).

To find the solution to the max_nbors issue, we remark that the two figures (3.5a and 3.3a) have some visual similarity. This suggests it may be more appropriate to compare the two results in terms of max_nbors as a proportion of the total advisor population. An effort to do this is provided as Figure 3.6.³ This figure confirms that when setting max_nbors as a proportion of the total advisor population, the accuracy of the trust model is relatively consistent from one population size to the next.

³Note that some of the data in this figure was interpolated from the simulations, since comparable proportions were not used in both sets of experiments.



(a) Comparison when varying *max_nbars*



(b) Comparison when varying threshold

Figure 3.7: Mean absolute error applying optimizations to TRAVOS with advisor population of 500, compared to results for population of 80

A similar set of simulations were also run to explore the effect of the larger population with the TRAVOS model. These are shown in Figure 3.7.⁴ Much like the results for the smaller population discussed in Section 3.3.2, the changes in the mean absolute error as the applicable parameter increases or decreases tend to be somewhat haphazard. However, the graphs for the two populations do not seem to track each other very well, in that there are no local maxima or minima that are consistent for both sizes of advisor populations.

As discussed above, we are proposing future research on this matter in Section 6.2.3, which might lead to a more effective method consistent with that shown earlier for PTM. At the moment however, we must conclude that, although thresholding and *max_nbor*s will allow us to improve the accuracy of trust modelling using TRAVOS with the larger population, the specific parameter choices will likely change as the population increases.

3.3.4 Using Random Selection for Very Large Populations

Despite having verified the usefulness of our techniques for larger population sizes, particularly for PTM, in the previous subsection, we also note practical limits on the advisor population sizes for which these methods can be used.

In considering the effects of changing the overall advisor population size, we found that using very large populations posed a more challenging situation in terms of memory consumption and the time of execution required. In particular, the means by which we created and sorted advisor networks for the primary buyer (as well as for each advisor, assuming we wanted to allow referrals) adds the most computational complexity, as described as follows:

- If there is a total of n buyer (advisor) agents in the system, sorting the advisor network for *one* of those agents using an algorithm such as merge sort will take $O(n \log n)$ (other sorting algorithms may of course take longer).

⁴Note that unlike Figure 3.6, in this case we did perform additional simulations for the proportional *max_nbor*s cases using the smaller advisor population. We chose not to interpolate values from our earlier results due to the wide variations in MAE noted previously for TRAVOS.

- In PTM, there is one advisor network for each of the n buyer agents, yielding complexity of $O(n^2 \log n)$.
- In TRAVOS, as we reason about the seller being considered when computing the trustworthiness value of an advisor, there are in effect k advisor networks to be computed for each buyer (where k is the number of sellers), such that the complexity will be $O(kn^2 \log n)$.

We thus considered the possibility of using a variant in which the advisor population considered for the purposes of our limiting mechanisms (both for thresholding and for *max_nbars*) would be restricted to a fraction of the total population size. This would be different than *max_nbars* in that in this case, the “new” population would be a randomly-selected sample of the advisors, and not necessarily the most trustworthy among them.

Our hypothesis was that using a smaller, randomly-selected sample, with a size on the order of that considered in the earlier work, would still allow us to avail of agents with a range of experiences, not to mention a range of honesties. We felt that we should still be able to avail of a sufficient supply of information to improve the accuracy of our trust modelling.

To test this hypothesis, we performed the following modifications to our algorithm, using PTM as the trust model: The buyer and the advisors would each interact as usual with sellers to build each agent’s set of experiences. Following the procedure, the buyer would then proceed to select a pre-determined number of advisors to act as its own advisor “population”, and use the PTM to model the trust it should hold in each of these advisors. This randomly-selected population would then optionally be limited using *max_nbars* or thresholding as applicable. For completeness, each of the advisors (being themselves buyer agents) would likewise randomly select the same number of advisors – but, we emphasize, *not* the same composition of advisors – for its own population; however, we believe this was not consequential since we did not test referrals at this stage.

We then tested this scenario using a “reduced” population of 100 advisors out of a total population of either 100 (our baseline), 500, or 1000. This was tested with both 30% and 60% lying advisors (LA), and for both the *max_nbars* and thresholding techniques, using

a small selection of parameters in each case. The simulations all used a seller population of 100, and were otherwise identical to the other *max_nhors*- or thresholding-only experiments documented in Section 3.3.1.

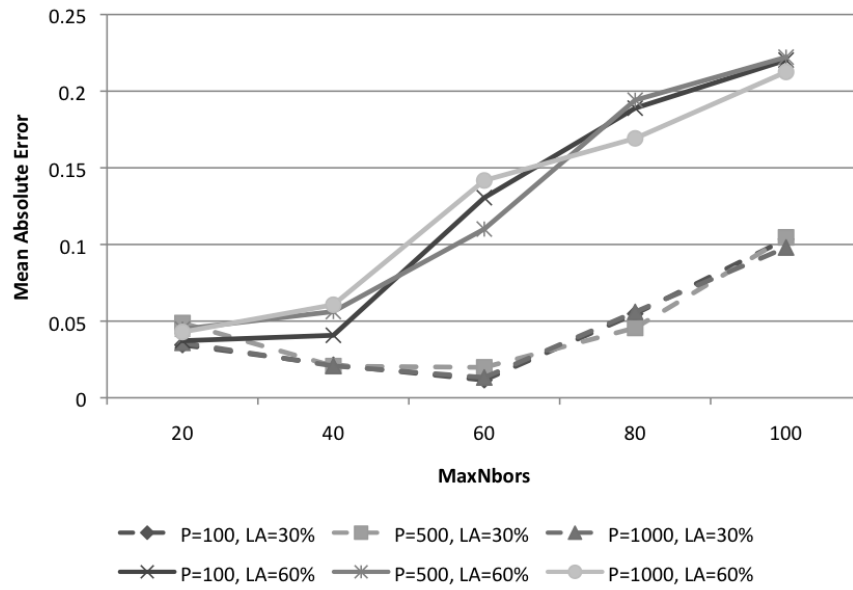
The results of these simulations are shown in Figures 3.8a and 3.8b – each graph represents a selection for the total population size and the percentage of lying advisors, with the *x*-axis representing a *max_nhors* or threshold parameter, and the *y*-axis indicating the mean absolute error (MAE) in the trust model for that simulation. For greater clarity, the graphs representing the 30% LA simulations are drawn with dashed lines, while those showing the 60% LA scenarios are drawn with solid lines.

While the results were not identical for each of these cases, on the whole it seems that trust-modelling accuracy (as represented here by the MAE) was very similar for all of the populations chosen – for each combination of LA percentage and limiting method (*max_nhors* or thresholding), the graphs for the differing populations are generally quite consistent. Any incongruities in the data shown seem to be very small, and not necessarily unexpected given the probabilistic nature of the PTM.

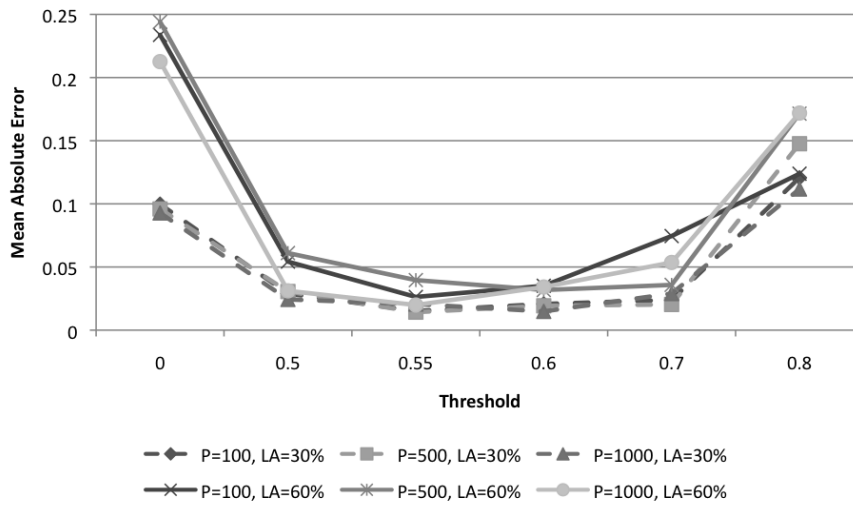
It may be worthwhile to try additional population sizes for advisors and sellers, as well additional sizes for the random-selection population, to more convincingly prove this point. Moreover, it would also be useful to include referrals into future simulations to verify its usefulness under these scenarios – indeed, in these cases referrals may prove to be even more useful, since an advisor’s advisor network might include additional experienced agents that were not included in the buyer’s random selections, and thus not previously considered as advisors. However, from our results to date, it seems very likely that random selection of a subset of the advisor population will indeed be sufficient to achieve trust accuracy roughly equivalent to that achieved by using the entire population.

3.3.5 Experimental Conclusions

These results suggest to us that our proposed optimizations – *max_nhors* and thresholding – can be expected to help model trust more accurately in other trust approaches, at least those similar to the PTM and TRAVOS – for example, the Beta Reputation System [13].



(a) Results when applying *max_nbars*



(b) Results when applying thresholding

Figure 3.8: Verification testing for the modifications.

Moreover, although the exact “optimal” parameters will likely differ from one system to the next, our results suggest that, for PTM, once the applicable threshold or *max_nhors* value has been determined for one population, the same values (in the case of *max_nhors*, the same proportion) can be used for other populations. In turn this could simplify these calculations greatly since it may only take a small population, perhaps 20 or smaller, to accurately determine the optimal threshold or (proportionate) *max_nhors* values. For TRAVOS, however, our results suggest that separate computations must be performed for different advisor population sizes.

In comparing the experimental results from *max_nhors* and thresholding in this chapter – particularly those in Figures 3.3a and 3.3b – we note that, for PTM, the best-performing threshold parameters seem to yield slightly lower trust-modelling error compared to the best-performing *max_nhors* parameters, at least among the specific parameter choices tested here. On the other hand, however, as seen in Section 3.3.2, the best *max_nhors* choice for TRAVOS is slightly better than the best thresholding choice in that case. As a result, we do not feel that we have seen enough evidence to say that one approach or the other is “better”; rather, we conclude that they both provide satisfactory results, and that the best option may depend on the particular trust model and other aspects of the scenario.

We will provide some more general suggestions to the research community about how to apply these techniques in Section 5.1, and then how to determine the appropriate parameters to use in each such technique in Section 5.2.

Chapter 4

Advisor Referral Systems

4.1 Proposed Technique

We wish to consider the possibility of combining some population-limiting mechanism – such as one or both of the methods discussed in the previous chapter – with an advisor-referral technique inspired by the one in [39] and discussed in Section 2.3. We diverge somewhat from the suggestions in [39] insofar as the PTM does not require us to query each advisor for a recommendation. Rather, the buyer has access to each advisor’s ratings for a given seller s via a central server, and uses this data, weighted by the buyer’s trust in each advisor, to determine the public (or network) reputation for the seller.

We thus consider that advisors can “advise” by allowing buyers to make use of each advisor’s own private reputation for a certain seller. In this case, an advisor “referral” system could be implemented using a variant of the measure used to weight private reputation in the original PTM (that is, Equation 2.5). This would work as follows: For each advisor a_j in the advisor network of b , that is, the set $A_b = \{a_1, a_2, \dots, a_k\}$, b checks whether advisor a_j is an acceptable advisor for the seller s . This will be the case if $N_{all}^{a_j} \geq N_{RE}$, where $N_{all}^{a_j}$ is the number of ratings provided by an advisor a_j for s , and N_{RE} (or simply RE) is some minimum number of ratings, representing the amount of experience (i.e., number of interactions, being equivalent to the number of ratings) that an advisor must have had

with the seller in order to be used. (We will explore shortly how to select an appropriate value for N_{RE} .)

If a_j is not an acceptable advisor (that is, if $N_{all}^{a_j} < N_{RE}$), the algorithm will query a_j 's own advisor network, which we would expect should be different from that of b since each buyer / advisor agent creates its own model of how much to trust each other agent.¹ The advisors in this network are sorted from most trustworthy to least trustworthy from the perspective of a_j , and are examined in order to determine, in a similar fashion to the examination of a_j itself, which (if any) of the advisors in a_j 's advisor network meet the criteria to be a suitable advisor for s . The first such advisor encountered that is itself not either (a) already in the set of acceptable advisors; or (b) in A_b — since this would imply that the recommended advisor would be added in any event at a later stage — will be accepted into the set of acceptable advisors, A_s .

This has the effect of “replacing” a_j in the advisor network with respect to the calculations for s : the new advisor will be treated exactly the same way that a_j would have been, with the substitution of its data (and b 's trust in the replacement advisor) for that of a_j . However, this “replacement” only applies with respect to s ; the algorithm will begin anew with the original advisor network A_b for subsequent sellers.

If none of the advisors of a_j meet the criteria stated above, the step would be repeated at each subsequent level of the network — that is, the advisors of each member of the set of advisors just considered — until an acceptable, unduplicated advisor was identified, and then accepted into A_s , again serving as a “replacement” for a_j .

However, this recursion is subject to limitations, since it is not guaranteed that there are at least k buyers that have each had at least N_{RE} interactions with s . To ensure broad coverage of the network while preventing infinite recursion, we limit the number of network “levels” calculated to at most $maxnetlevel = \lceil \log_k(|B|) \rceil$, where B is the set of all buyers (advisors) in the system.² If, after searching $maxnetlevel$ levels, no acceptable advisor has been found to replace a_j , no replacement advisor will be accepted into A_s , and the system

¹We assume for the purposes of this discussion that these advisor networks can be retrieved from a central server, as with the ratings themselves in the PTM.

²We note that practically, in a large scale system, the number of levels may need to be smaller in order for this algorithm to be computationally efficient; we will leave such a decision for later work.

will use a reduced set to determine the network reputation — that is, the size of A_s will be less than k .

Once the set of acceptable advisors has been determined, the “network” reputation would be calculated as in the original model, using the advisor trustworthiness values held by the buyer b .

We summarize this mechanism in pseudo-code format as Algorithm 2.

Algorithm 2 Referring Advisors to Buyer b for Trustworthiness of Seller s

```

1:  $A_b = \{a_1, a_2, \dots, a_k\}$ ; {advisors in  $b$ 's advisor network}
2:  $A_s = \{\}$ ; {set of advisors that are suitable for providing advice regarding seller  $s$ }
3:  $N_{RE}$  = minimum number of ratings for  $a$  to be a suitable advisor regarding  $s$ ;
4:  $maxnetlevel = \lceil \log_k(|B|) \rceil$  {the maximum number of search iterations}
5: for  $j = 1$  to  $k$  do
6:    $N_{all}^{a_j}$  = total number of ratings provided by  $a_j$  for  $s$ ;
7:   if  $N_{all}^{a_j} \geq N_{RE}$  then
8:     append  $a_j$  to  $A_s$ ;
9:   else
10:     $netlevel = 2$ ; {no. of connections between  $b$  and the advisors being searched}
11:     $a_x = \mathbf{null}$ ; {the desired suitable advisor in place of  $a_j$ }
12:     $A_c$  = the set of advisors for  $a_j$  sorted from most to least trustworthy (as per  $a_j$ );
13:    while  $a_x == \mathbf{null}$  and  $netlevel \leq maxnetlevel$  do
14:       $A_n = \{\}$ ; {the set of advisors to be considered in the next round, if necessary}
15:      for all  $a_c$  in  $A_c$  do
16:         $N_{all}^{a_c}$  = total number of ratings provided by  $a_c$  for  $s$ ;
17:        if  $N_{all}^{a_c} \geq N_{RE}$  and  $a_c \notin A_b$  and  $a_c \notin A_s$  then
18:           $a_x = a_c$ ;
19:          break;
20:        else
21:          add the set of advisors for  $a_c$  to  $A_n$ ;
22:        end if
23:      end for
24:       $netlevel ++$ ;
25:       $A_c = A_n$ 
26:    end while
27:    if  $a_x \neq \mathbf{null}$  then
28:      append  $a_x$  to  $A_s$ ;
29:    end if
30:  end if
31: end for

```

4.2 Examples

We now provide examples demonstrating the effects of adding our advisor referral mechanism to the PTM where *max_nbor*s or thresholding is already applied. The examples in this section continue on from those outlined in Section 3.2, in which we showed how *max_nbor*s and thresholding would be applied to a system initially consisting of a buyer b ; four advisors, a_w , a_x , a_y , and a_z ; and six sellers, s_0 through s_5 . The buyer and each advisor each had a separate set of interactions (and thus ratings) with each seller, as indicated in Tables 3.1 and 3.3.

We now introduce a new advisor into the system, a_v , as well as an additional seller, s_6 . To this point a_v has only provided ratings for s_6 , while b has not provided any ratings for that seller; therefore there are no commonly-rated sellers for a_v and b , and thus $Tr(a_v) = 0.5$ from the perspective of b .

We also assume, as in the *max_nbor*s = 3 or $L = 0.4$ cases described in Section 3.2, that the advisor network for the buyer b consists of the set of advisors $\{a_w, a_x, a_y\}$ — a_v is too new to have been considered as a potential advisor in that case, although for purposes of demonstration we assume that a_v has somehow been included in the advisor networks of some of the other advisors. Finally we set N_{RE} , the minimum amount of experience (in terms of number of ratings) for an advisor to be considered acceptable for a given seller, as 3. The ratings that have been given by each advisor, and the resulting discounted amounts, are as shown in Table 4.1.

Given this information, the buyer b will examine its advisor network and find that a_w and a_x are indeed acceptable advisors for s_6 , since both have achieved at least N_{RE} interactions with s_6 . However, a_y has only had one interaction with s_6 , and would therefore not be considered an acceptable advisor. The buyer will then look to a_y 's advisor network to identify an appropriate substitute.

Suppose then that a_y also has a three-agent advisor network consisting of a_v , a_x , and a_z , with trustworthiness values 0.5, 0.6, and 0.7 respectively.³ This information will be

³For clarity, only a_y 's own regular advisor network is considered, even if a_y has itself used referrals in modelling its own trust in s_6 .

Table 4.1: Ratings of s_6 Provided by Advisors

(a) Ratings						(b) Amounts of Ratings					
T_i	T_1	T_2	T_3	T_4	T_5	T_i	T_1	T_2	T_3	T_4	T_5
a_v	1	1	0	1	1	$N_{pos,i}^{a_v}$	1	1	0	1	1
a_w	0	1	1	0	-	$N_{neg,i}^{a_v}$	0	0	1	0	0
a_x	1	0	1	-	-	$N_{pos,i}^{a_w}$	0	1	1	0	-
a_y	0	-	-	-	-	$N_{neg,i}^{a_w}$	1	0	0	1	-
a_z	1	1	-	-	-	$N_{pos,i}^{a_x}$	1	0	1	-	-
						$N_{neg,i}^{a_x}$	0	1	0	-	-
						$N_{pos,i}^{a_y}$	0	-	-	-	-
						$N_{neg,i}^{a_y}$	1	-	-	-	-
						$N_{pos,i}^{a_z}$	1	1	-	-	-
						$N_{neg,i}^{a_z}$	0	0	-	-	-

(c) Discounted Amounts of Ratings

T_i	T_1	T_2	T_3	T_4	T_5
$D_{pos,i}^{a_v}$	0.4	0.4	0	0.4	0.4
$D_{neg,i}^{a_v}$	0	0	0.4	0	0
$D_{pos,i}^{a_w}$	0	0.397	0.397	0	-
$D_{neg,i}^{a_w}$	0.397	0	0	0.397	-
$D_{pos,i}^{a_x}$	0.937	0	0.937	-	-
$D_{neg,i}^{a_x}$	0	0.937	0	-	-
$D_{pos,i}^{a_y}$	0	-	-	-	-
$D_{neg,i}^{a_y}$	0.375	-	-	-	-
$D_{pos,i}^{a_z}$	0.0294	0.0294	-	-	-
$D_{neg,i}^{a_z}$	0	0	-	-	-

gathered by b as the ordered list $\{a_z, a_x, a_v\}$. The buyer will then iterate through the set, discarding a_z as an unacceptable advisor (having provided only two ratings for s_6), and also a_x as it is already in b 's advisor network. Finally, b would then accept a_v as the third advisor, as it has an acceptable level of experience with s_6 but is not part of b 's own advisor network.

As in the previous examples, b does not itself have enough experience with s_6 to generate a private reputation. Therefore, using the above information for the set of advisors $\{a_v, a_w, a_x\}$, the forgetting factor (as defined in Section 2.1.1) $\lambda = 0.9$, and Equation 2.10, we find that $Tr(s_6) = 0.6655$.

If b had not used advisor referrals but instead relied solely on its existing advisor network, namely $\{a_w, a_x, a_y\}$, it would have obtained a significantly different result — $Tr(s_6) = 0.5549$. However, the latter result makes much less use of the experience within the network for s_6 than did the one incorporating advisor referrals.

4.3 Experimental Results

We refer the reader to the beginning of Section 3.3 for a summary of pertinent information regarding the experimental setup of the simulations for this thesis.

4.3.1 Verifying Effectiveness

In Section 3.3.1, we conducted several simulations of the PTM using one buyer, 80 advisors, and 100 sellers, with each buyer and advisor interacting with 80 randomly-chosen sellers, to test the effectiveness of *max_nbors* and thresholding using several possible parameters. We now repeat these simulations using a small subset of the best-performing cases using advisor referrals. Specifically we chose *max_nbors* = 40 and *threshold* = 0.55 based on the results in Section 3.3.1, as well as *max_nbors* = 15, which as noted in that section did not perform significantly worse compared to setting a maximum size of 40, in order to test whether the smaller size might (by itself) cause any differences in later simulations. Referrals were applied with the required minimum number of experiences (N_{RE}) set to

1, the most that would have been available in this scenario since we had prohibited each advisor from interacting with each seller more than once.

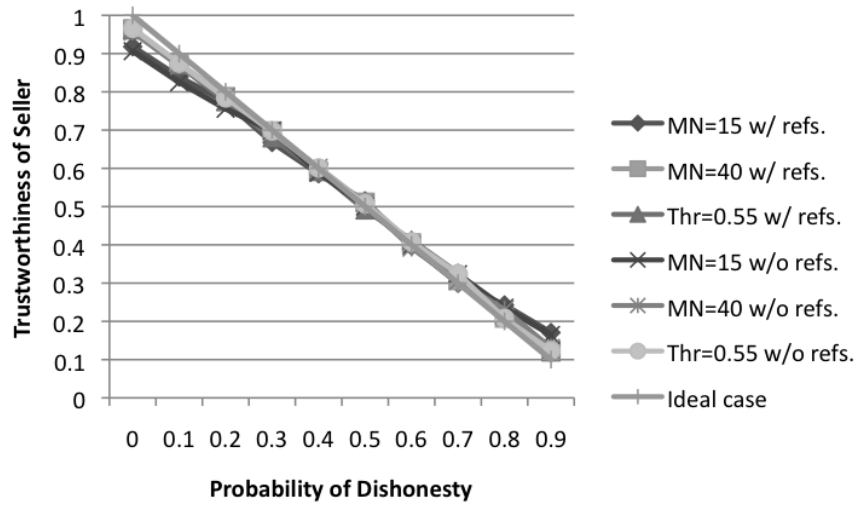
The results of these simulations, as compared to the equivalent versions without advisor referrals, are shown in Figures 4.1a and 4.1b. These graphs show that using the referral mechanism results in accuracy about as good as, if not better than, that of the corresponding variant without referrals, in almost all cases, indicating that applying referrals should *not* significantly reduce the accuracy of the trust model. This is not entirely surprising for these scenarios, since in each case the size of the advisor network has already been optimized to provide the best results, with a sufficient number of users having expertise with most, if not all, sellers.

A summary view is provided as Figure 4.2, consisting of a scatterplot showing how each of the variants tested performed, for both the 30% and 60% lying advisor cases, with or without referrals (when possible).⁴ As in the summary figures used in Section 3.3, the *y*-axis represents the mean absolute error (MAE) of each of the variants in calculating the trustworthiness of sellers as compared to the “ideal” trust model discussed previously. Note that in some cases, two or more data points (icons) overlap, indicating that the MAE for those simulations were approximately equal (the order in which they overlap is not meaningful). Again, as the data points corresponding to the referral-based variants are almost always at about the same level as those of their non-referral equivalents, we can conclude that adding referrals will generally yield about the same accuracy as if referrals are not used. Further evaluation of the effect of referrals will be discussed in the next subsection.

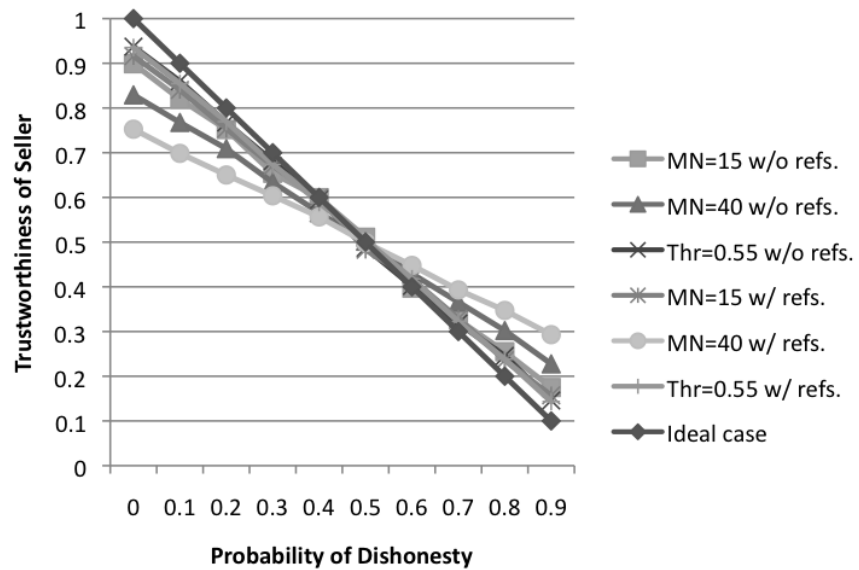
4.3.2 Using Referrals to Further Reduce Network Size

We have shown above that using *max_nbor*s or thresholding to limit the size of the advisor network will significantly improve the accuracy of the trust values calculated for each seller. We have further shown that, given certain “optimal” choices for the *max_nbor*s or

⁴Specifically, simulations were not conducted for referrals where neither *max_nbor*s or thresholding is used, since referrals are of no effect unless the advisor network has already been limited.



(a) 30% lying advisors



(b) 60% lying advisors

Figure 4.1: Comparison of approaches with and without referrals.

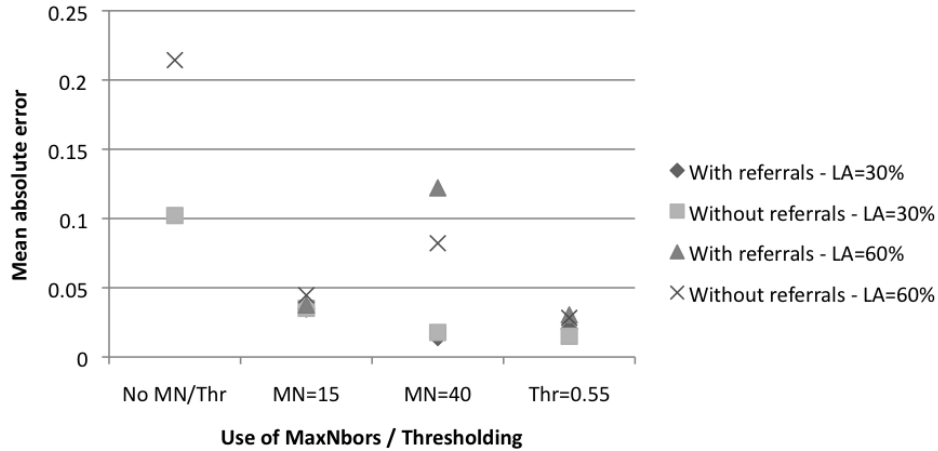


Figure 4.2: Mean average error of compared trust models, both with and without referrals.

threshold values, using advisor referrals will achieve at least approximately the same level of accuracy compared to when referrals are not used.

However, we believe that referrals could be useful when using a smaller advisor network, i.e. a smaller value of *max_nbars* or a higher trustworthiness threshold. Such a scenario might arise due to a need for limiting memory and processing; this will be discussed in greater detail in Chapter 6. In this scenario, it is less likely that the advisors within the network will have a sufficient level of experience – if any experience – in dealing with each and every seller in the system. Here, then, we would expect referred advisors to be more useful in regards to filling in the gaps in experience.

With this in mind, we proceeded to a modified version of the above evaluation that would allow for a greater role for referrals. The parameters and test conditions were the same, except that we reduced the number of sellers to 40, and increased the number of simulation days to 120. We also adopted pure random selection for the sellers, such that buyers would rate each seller a variable number of times (on average three), whereas previously they could rate each seller at most once.

Simulations were then performed in this environment using several variations incorporating *max_nbars* or thresholding, as well as referrals. A subset of these simulations, using *max_nbars* = 2 where 30% of the advisors are dishonest, are shown in Figure 4.3.

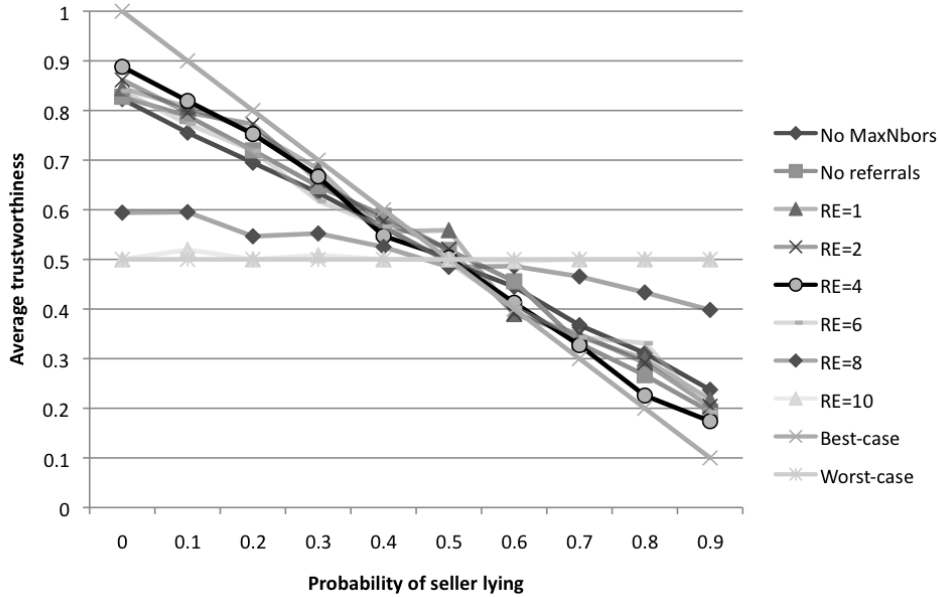
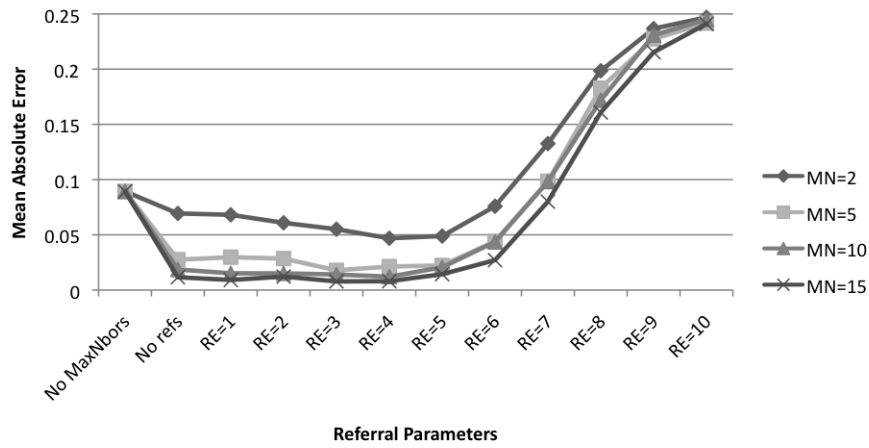


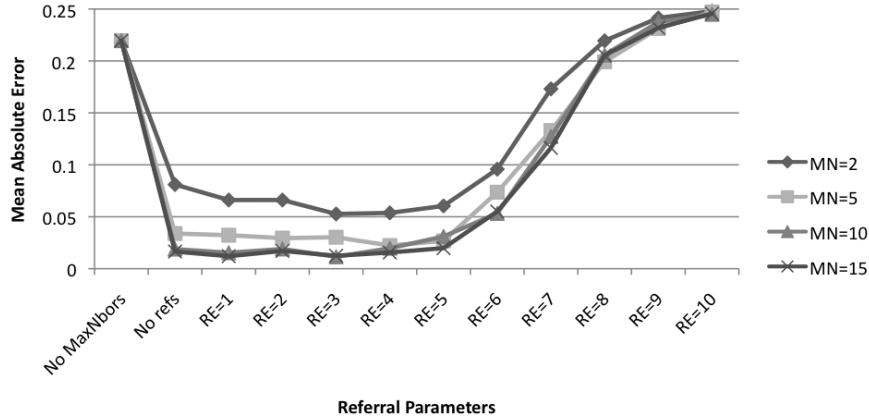
Figure 4.3: Comparison of trust models ($max_nbors = 2$, 30% lying advisors (LA))

The independent variable in this case is the use of referrals and, if referrals are used, the amount of experience required from the referred advisors (N_{RE}), which is varied from one to ten. As in the previous subsection, the graph displays each variation’s average trust value for the sellers with the indicated probability of lying. Also shown for comparison is the “ideal” trust model where the calculated average trust value exactly corresponds to the sellers’ probability of dishonesty – the straight diagonal line; as well as the “worst-case” scenario where all of the advisors are excluded, yielding an average trustworthiness of 0.5 regardless – the horizontal line.

It is clear, based on comparing the closeness of each of the variations’ graphs to the “ideal” graph, that while setting a maximum size for the advisor network yields a more accurate trust model, adding referrals in this case results in a further non-trivial improvement. Furthermore, increasing the N_{RE} value – the experience with the seller that each referred advisor must have – serves to improve the model further; an experience level of four – just above the expected average noted above – comes closest to matching the best-case scenario.



(a) Error comparison for various *max_nbrs* + referrals approaches with 30% LA



(b) Error comparison for various *max_nbrs* + referrals approaches with 60% LA

Figure 4.4: Evaluation of effects of referrals on small *max_nbrs* advisor networks.

However, demanding a much higher level of experience reduces the number of available advisors significantly, approaching the worst-case scenario if an N_{RE} value of ten is required. As we noted above, the average number of experiences that an advisor should have with each seller is three, making a much higher value like ten a very rare occurrence in this simulation. If none of the advisors have met the minimum N_{RE} value for any of the sellers, no advisors will be included in the advisor network, and the level of seller trust will revert to 0.5, the default value if no ratings are available. This is not unlike what was encountered in Section 3.3.1 with regards to thresholding; in both cases, an unrealistically high parameter will result in no useful information being obtained. These results will, in turn, will serve to increase the overall trust-modelling error associated with the scenario being tested – not because the more highly experienced advisors are incorrect, but rather due to very few (if any) sufficiently-experienced advisors being available.

Additional simulation results are shown in summary view as figures 4.4a and 4.4b, which are for 30% and 60% lying advisors, respectively. Each graph represents a single possible value for max_nbors . The positions on the x -axis represent the N_{RE} value used for referrals, if any, except for the position at the far left which indicates the results if neither max_nbors nor thresholding is used. The y -axis indicates the mean absolute error for each of the variants measured, which is the same as outlined for the summary views in the previous subsection.

For $max_nbors = 15$ – identified in Section 3.3.1 as one of the “optimal” network sizes for an advisor population of 80 – using referrals seems to give an improvement, albeit extremely slight. However, for a smaller network such as $max_nbors = 2$, the improvement is much more pronounced, as indicated by the reduced error for the $max_nbors = 2$ graph if the experience level is set to $N_{RE} = 3$ or $N_{RE} = 4$. For example, while using $max_nbors = 2$ without referrals in the 60% lying advisors had an MAE of 0.081, allowing for referrals with an N_{RE} value of 4 led to a significantly smaller MAE of 0.054. Although this does not overcome the benefits of using a larger max_nbors value – for example, setting $max_nbors = 5$ without using referrals resulted in an MAE of 0.034 – these results are still much closer in terms of accuracy.

Figure 4.5 shows a similar set of simulations for various possible threshold levels, again

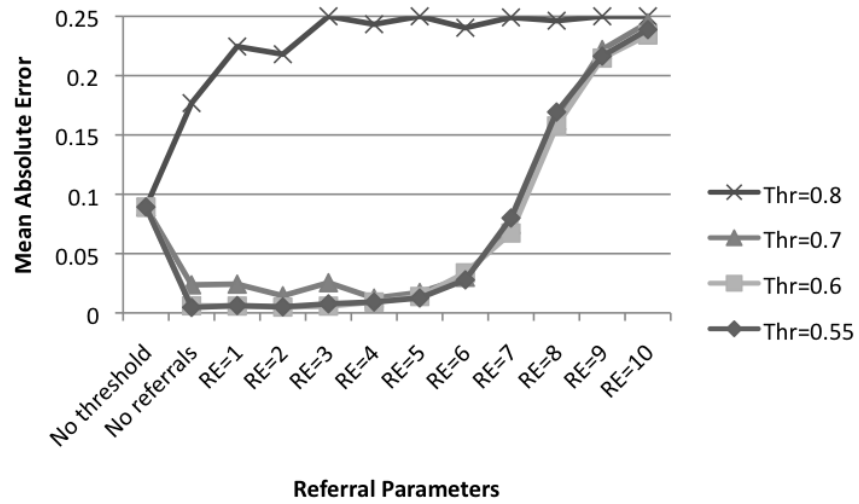
showing how different combinations of thresholds (graphs) and referral experience levels (x -axis) affect the mean error of each trust model variant (y -axis). These figures show that, as with *max_nbors*, adding referrals with an appropriately-chosen experience level in combination with thresholding can reduce the error in the trust model for higher thresholds – but only up to a point. If the threshold is set so high that some buyers end up having advisor networks of size 1 or 0 (as appears to be the case for using a threshold of 0.8), then referrals could end up being quite ineffective.

We conclude that our evaluation indicates that referrals can serve to improve the accuracy of the trust model if the size of the advisor network is very limited, such as if there is a very low maximum number of advisors, or a very high trustworthiness threshold.

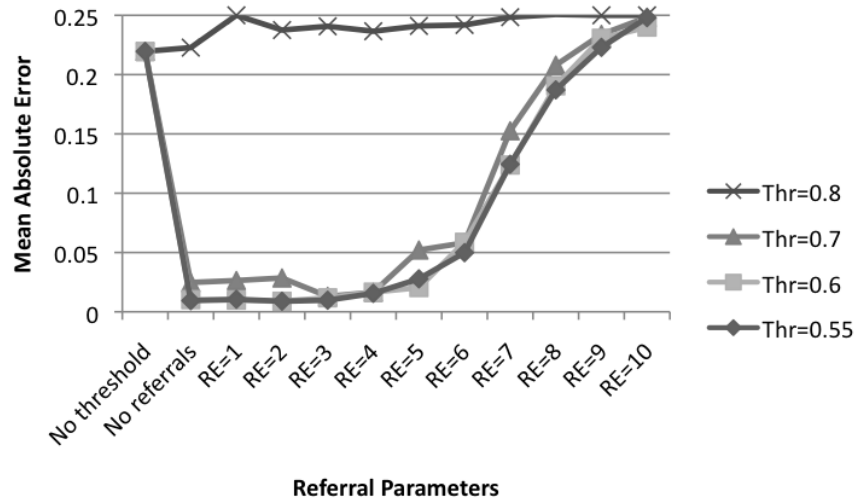
4.3.3 Applicability to Alternative Model

Continuing from the results in Section 3.3.2, we now look at examining the effect of advisor referrals using TRAVOS. Again, as with our work in the previous subsection, this is performed using a modified version of the scenario in Section 3.3.2, with the number of sellers reduced to 40, and each buyer or advisor submitting 120 seller ratings, with no limit on the number of times each seller could be chosen.

The results for these tests are shown in Figures 4.6 and 4.7. As with the earlier figures, these are summary graphs which indicate the mean absolute error obtained for various combinations of minimum referral experience (N_{RE}) and *max_nbors* / threshold parameters; each series represents a different *max_nbors* or threshold value, while the x -axis indicates the corresponding N_{RE} value. Like the results for PTM (see Figures 4.4 and 4.5), these graphs show that for low values of *max_nbors*, where the advisor network size is very small, using referrals will provide a reduction in error (that is, a trust model with improved accuracy). When using thresholding, similar reductions in error were observed by adding advisor referrals to networks using high threshold values (and hence having a small size). However, reductions in error were also occasionally seen for larger networks (those produced by using smaller thresholds); such improvements were rarely seen when applying referrals to large networks using PTM.

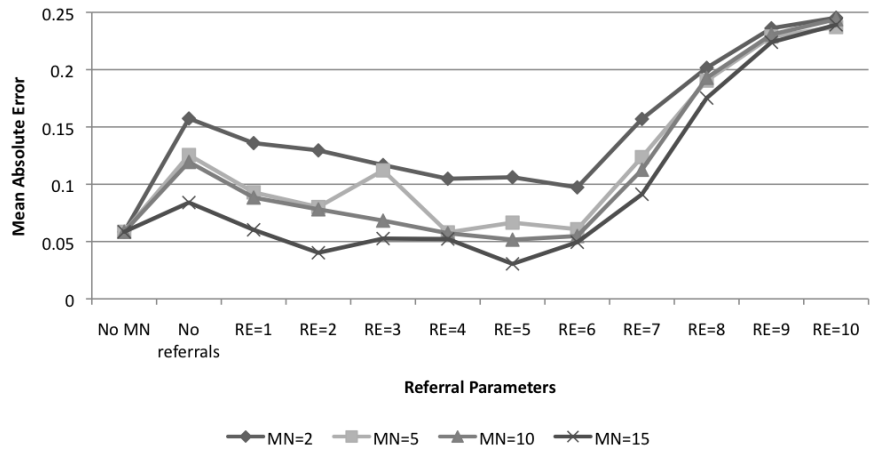


(a) Error comparison for various thresholding + referrals approaches with 30% LA

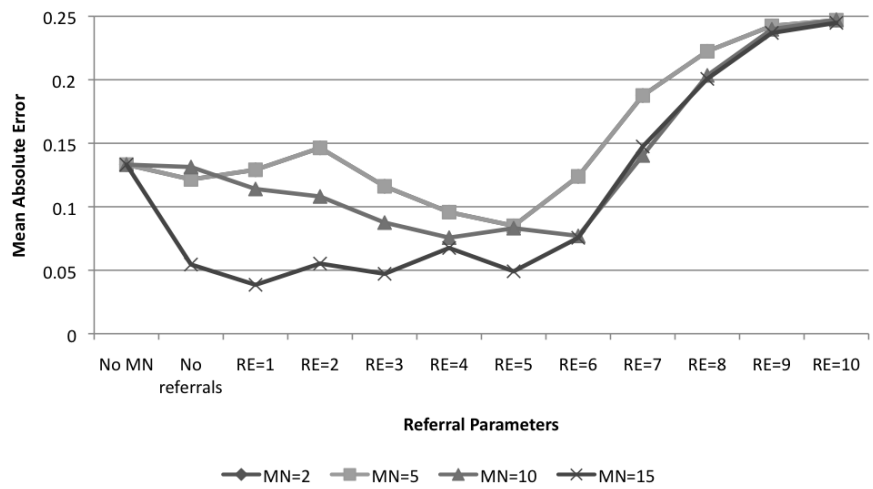


(b) Error comparison for various thresholding + referrals approaches with 60% LA

Figure 4.5: Evaluation of effects of referrals on advisor networks.

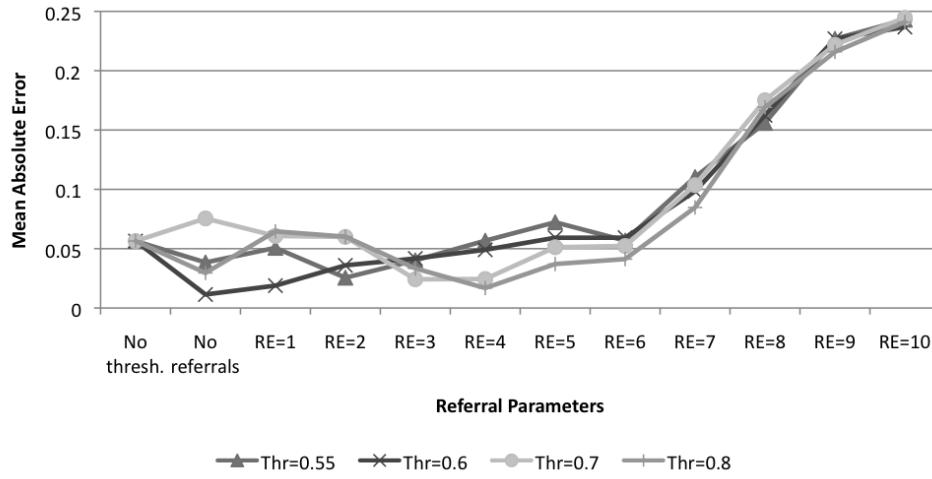


(a) Comparison for 30% LA

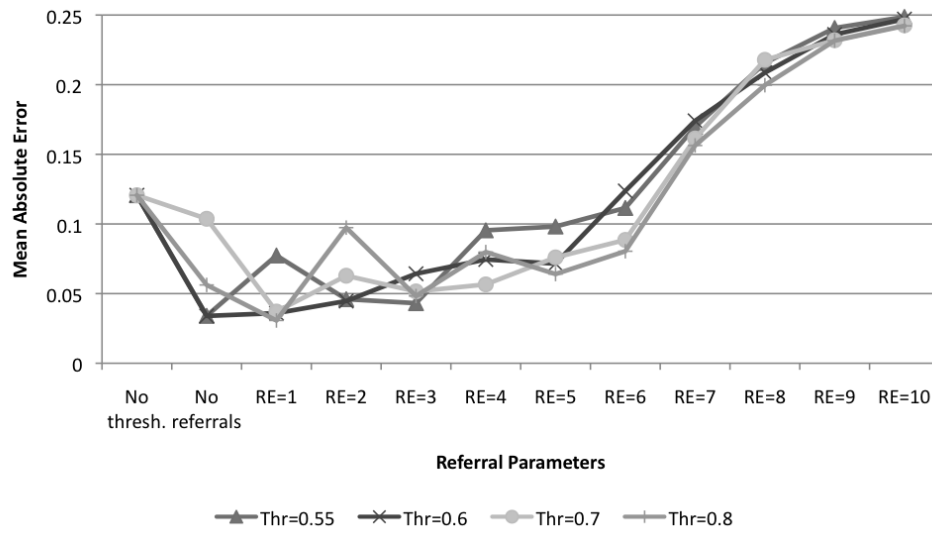


(b) Comparison for 60% LA

Figure 4.6: Comparison of mean absolute error in TRAVOS using advisor referrals when varying max_nbors and the minimum level of referral experience (N_{RE})



(a) Comparison for 30% LA



(b) Comparison for 60% LA

Figure 4.7: Comparison of mean absolute error in TRAVOS using advisor referrals when varying the trust threshold and the minimum N_{RE} level

In comparing the results for using thresholding with TRAVOS with those achieved using thresholding with PTM, we do note one anomaly. As shown in Figures 4.5a and 4.5b, the trust model error increases significantly for the PTM approach for all N_{RE} values when the threshold reaches 0.8. However, such increases did not occur for TRAVOS; as shown in Figures 4.7a and 4.7b, the error in that trust model for high thresholds (0.8 and 0.9) did not increase as significantly as they did for PTM; indeed, simulations using TRAVOS had overall lower error than those using PTM when applying a threshold of 0.8 to both cases.

The high error that is seen in PTM when applying larger threshold values is generally due to the buyer having modelled very few, if any, of the advisors with such a high threshold, which leads in turn to insufficient information to model the trust of sellers (and assigning the default trust value of 0.5). In comparison, during our simulations, TRAVOS would assign high trust values, on the order of 0.8 or 0.9, to advisors more frequently, potentially because that model uses a more fine-grained model of advisor trust based on the advisor, the buyer, and the seller under consideration (whereas PTM calculates an overall value based only on the buyer and advisor). Accordingly, setting a high threshold would not affect the amount of information available to TRAVOS in the same way that it would PTM, leading to the more accurate results in this case.

4.3.4 Effects for Larger Populations

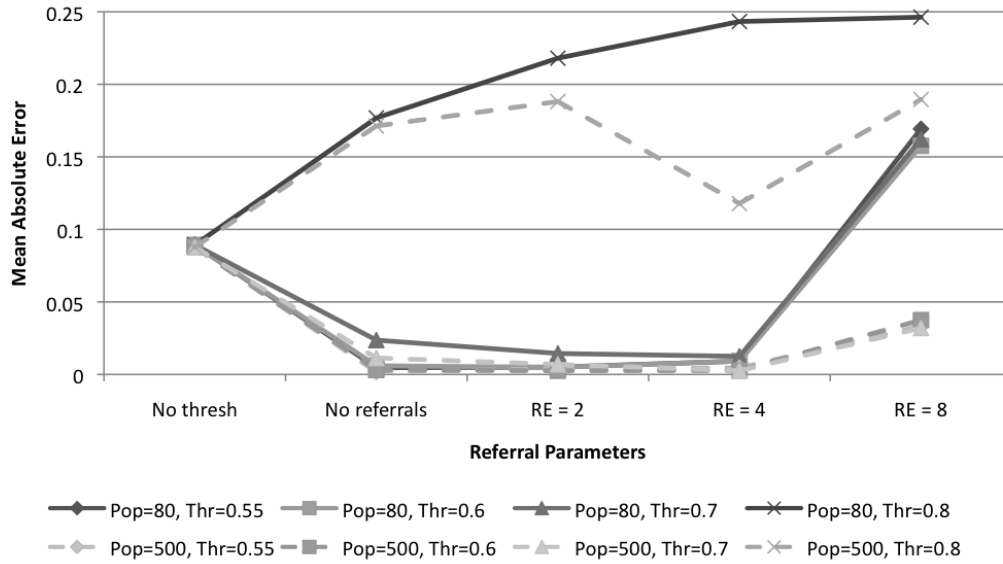
We now turn to testing advisor referrals with PTM for the large advisor population case previously examined in Section 3.3.3. Again, we use the modified scenario used for the advisor referral simulations performed in the earlier sections of this chapter, except with an advisor population of 500. However, we felt it was unnecessary to consider the same number of different minimum referral experience (N_{RE}) parameters as in the smaller-population case in order to demonstrate how the trust prediction accuracy changes as the N_{RE} value increases. We therefore restricted our simulations in this regard to a handful of N_{RE} values which, we considered, would nevertheless show any significant trends in the results.

We first consider how the larger population performs when thresholding and referrals are used in combination, as shown in Figures 4.8a and 4.8b. The results are not identical

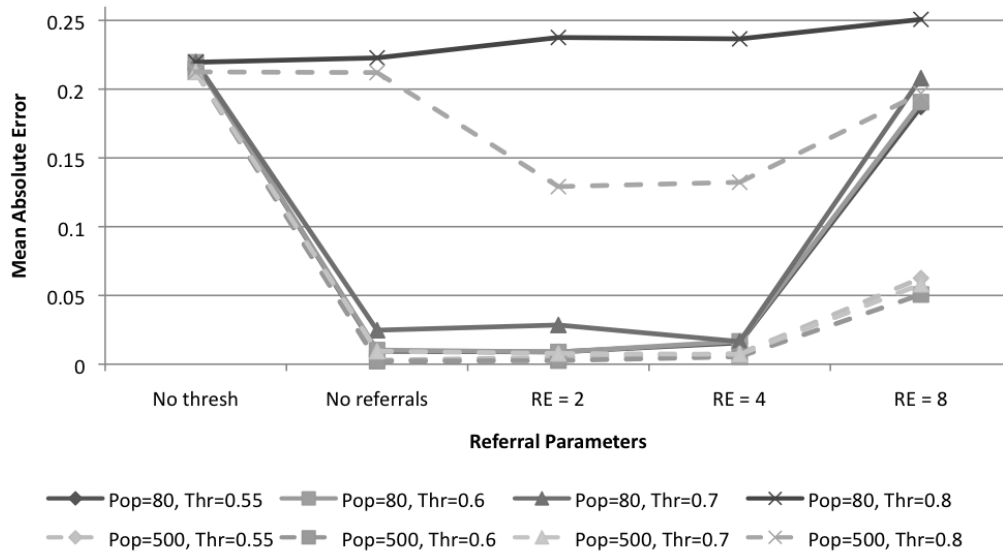
for the two population sizes – there is a generally smaller trust-modelling error for the larger population when high N_{RE} values are used – but for the cases of greatest interest, specifically small advisor networks resulting from high thresholds, using modestly-chosen N_{RE} values, there are still improvements when adding referrals to these networks. Indeed, for a threshold of 0.8, the positive effects of adding referrals are much more pronounced in the larger population than in the 80-advisor scenario, particularly for $N_{RE} = 4$. We attribute this to the fact that more highly-trusted advisors will be available in the larger population, which would make a significant difference considering that perhaps only one or two advisors would survive the thresholding process using the smaller population.

Next we look at using *max_nbors* and referrals in combination, using *max_nbors* values of similar proportions relative to the population size, as shown in Figures 4.9a and 4.9b. In this case, applying both techniques to a large network results in very similar, and in some cases (particularly for $N_{RE} = 8$) much lower accuracy error compared to the smaller network. It seems safe to conclude that using referrals with a large advisor population will not only be effective in general, but that it will yield trust modelling accuracy at least as good as that obtained with a smaller population.

We can thus expect that advisor referrals can help to model trust more accurately in other trust approaches such as BRS [13]. Indeed, our findings indicate that as the size of the advisor population increases, the benefits of using advisor referrals with regards to the accuracy of the trust model will also increase.

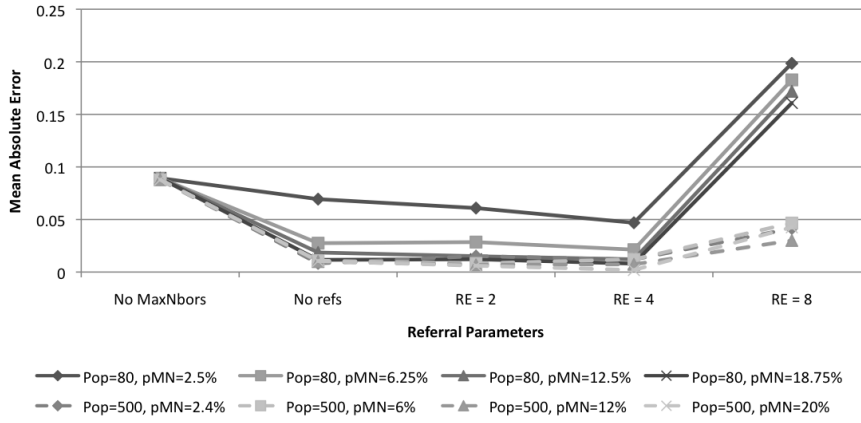


(a) Comparison when varying threshold (30% LA)

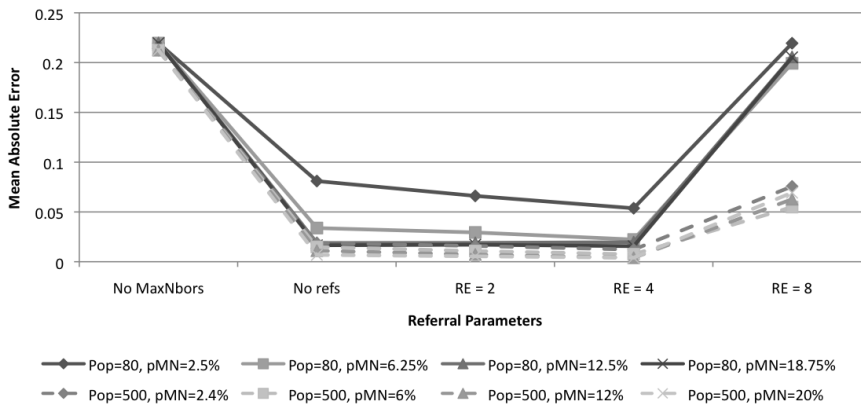


(b) Comparison when varying threshold (60% LA)

Figure 4.8: Comparison of mean absolute error in PTM using advisor referrals, varying threshold and minimum N_{RE} , when advisor population is 500



(a) Comparison when varying max_nbors (30% LA)



(b) Comparison when varying max_nbors (60% LA)

Figure 4.9: Comparison of mean absolute error in PTM using advisor referrals, varying (proportional) max_nbors and minimum N_{RE} , when advisor population is 500

Chapter 5

Discussion

5.1 Applying Techniques to Other Trust Models

In Chapters 3 and 4, we showed that our proposed modifications – *max_nbars*, thresholding, and advisor referrals – can lead to improvements in trust modelling accuracy with both the PTM [43] and TRAVOS [32]. We now move on to describe how other trust researchers can apply our proposals to their own models in order to improve the accuracy of these models.

In order for these improvements to be effective, the model must be such that the primary goal is for some individual agent (for example, in the case of PTM, a “buyer”) to model the trust of each agent or item in some pre-defined group (in PTM, “sellers”), and that in doing so, it makes use of information about some separate group of agents (in PTM, “advisors”). For simplicity we will use the terminology from PTM for the balance of this section, but we emphasize that we intend for these techniques to be applicable in any domain where such a model could be used. We assume that some existing test code implementing the trust model has already been written, verified, and optimized using any existing parameters in the trust model.

The code should then be modified to allow for the following:

- If the thresholding technique is being applied, some data structure should be used to hold information about the advisors that have trust values exceeding the pre-

determined threshold, as well as the trust values themselves. As the trust values for each of the potential advisors are determined, the code should compare each value with the threshold and, if and only if this value meets or exceeds the threshold, add the advisor (and its corresponding trust value) to the data structure. Subsequently, the trust value for each seller should be determined by only making use of the information provided by the advisors included in this data structure.

- If *max_nbor*s is used instead, a similar data structure should be used to store information about advisors and their trustworthiness values, although in this case the advisors will need to be sorted from most trustworthy to least trustworthy. In general it will suffice to set a maximum of *max_nbor*s items for this structure, with less trustworthy advisors being removed as more trustworthy advisors are found and added. In any event, once the advisors have been sorted, only the first *max_nbor*s items in the sorted data structure should be used to calculate the trustworthiness of sellers.
- Turning to the advisor referral technique:
 - Advisor referrals are used in conjunction with thresholding and *max_nbor*s, and thus the same processes should be used as if either of these two techniques were being used on its own – with one exception. The referral process we have proposed assumes that even if trust thresholding is used, the advisors will be sorted from most trustworthy to least, such that when a buyer is attempting to get an referral from an advisor, the latter will consider its own advisors from most trustworthy to least. As a result, sorting must be employed regardless of whether *max_nbor*s or thresholding is used initially.
 - To perform the referral mechanism itself, the code must include a mechanism for checking whether an existing advisor in the advisor network has met or surpassed a set referral experience (N_{RE}) level, with regards to the number of experiences that advisor has had with the seller under consideration. The advisor itself will be used if the minimum N_{RE} level has been achieved by that advisor. Otherwise, the code must examine all of that advisor’s own advisors,

from most trustworthy to least trustworthy, and if necessary the advisors of the advisors just considered, and so on, until either an acceptable advisor (i.e. one that has met the minimum N_{RE} value) has been found, or the maximum number of recursion levels has been exceeded. Further details about the algorithm is found in Section 4.1.

- In certain trust models, such as TRAVOS, the model is such that a separate trust value is calculated for each combination of buyer, advisor, and seller – that is, how much the buyer trusts the advisor to provide accurate information *specifically* about a single seller. In this case, regardless of the technique(s) used, the model will require an enlarged data structure (or multiple structures) to contain all this information. The threshold will then need to be applied separately for each seller under consideration.
- If there is any existing functionality in the original model that resembles these techniques, such features should be disabled. For example, the original PTM only used “trustworthy” advisors – advisors for which the buyer’s trust value was greater than 0.5 – equivalent to setting a threshold of 0.5.

5.1.1 Optimizing the Modifications

Each of the proposed modifications introduces at least one new parameter into the trust model, specifically:

- *max_nhors*, when setting a maximum size for the advisor network
- *threshold*, when using trustworthiness thresholding
- minimum N_{RE} , when using advisor referrals

Our work in Chapters 3 and 4 has indicated that the parameters in each of these techniques should not be applied blindly: there may be significant changes in the accuracy of the trust model as the applicable parameters are changed upwards or downwards. Accordingly, we suggest that trust researchers seeking to use these techniques should be careful in selecting the optimal parameters for their particular trust model.

It is not our present belief that such parameters could be found in a deterministic fashion – the properties of, or distinctions between, different trust models cannot be easily reduced to numerical values. We instead advise researchers to use the empirical route: test several reasonable options, then use the combinations that provide the best performance.

Our recommended test procedure, based on the work in [44] and earlier in this thesis, is as follows:

Run a simulation of the model with a single “buyer” agent, and a sufficient number of both advisors and sellers such that the number of interactions between either the buyer or a single advisor on the one hand, and a single seller on the other hand, will be insignificant compared to the total number of interactions. In other words, the numbers must be sufficiently large to ensure that any outliers that may exist in the data have an insignificant effect on the overall results. That said, we suspect that the exact choices in terms of the numbers of agents used will not be particularly meaningful, insofar as we saw no noticeable difference in the results upon increasing or decreasing these numbers slightly. We will suggest, arbitrarily, that using at least 40 sellers, and at least 80 advisors, seems to be sufficient to obtain reliable results.

Run this simulation such that the buyer and each advisor has at least one experience with a sufficient number of sellers that any outliers will have little effect – for example, following our procedure, if there are 80 sellers, each agent might have one or more interactions with 60 of those sellers.¹ This will suffice by itself if testing *max_nhors* or thresholding alone. However, when testing advisor referrals with $N_{RE} > 1$, the buyer and each advisor should normally have multiple interactions with each seller. In this case, the number of experiences (and/or the number of sellers) should be adjusted to ensure that each advisor has achieved the required number of experiences (i.e., the N_{RE} value) for some, but not all sellers, in order to adequately test the referral mechanism. For example, if testing an N_{RE}

¹The exact number or proportion to use will depend on the domain being considered. For example, on an online auction marketplace with a large user population, such as eBay, it is highly unlikely that any given buyer will come in contact with even a small fraction of the entire population of sellers. As such, it might be more appropriate to choose 10 or 20 sellers, or use a larger seller population, in such cases. At the same time, a buyer with a specific interest, e.g. stamp collecting, *might* come in contact with a significant proportion of the sellers concerned with that particular interest. Likewise, when purchasing an “information good”, having multiple experiences with a seller may not be unrealistic.

value of 5 with a seller population of 80, the test might be such that each advisor makes 400 interactions, each time randomly selecting one of the 80 sellers. In such a case, while, on average, an advisor will have had five interactions with each seller, the specific value associated with each advisor-seller pair may be different.

Regardless of the number of sellers, they should be divided into multiple disjoint trustworthiness categories, each containing the same number of sellers. Each category is assigned a different probability that the seller will act dishonestly ($P(d)$), which is then assigned to the sellers. For example, the evaluation in Section 3.3.1 divided the sellers into ten equal groups, one with $P(d) = 0$, another with $P(d) = 0.1$, and so on up to the tenth group with $P(d) = 0.9$. Then, each time a seller interacts with a buyer or advisor, it will choose whether to act honestly or dishonestly via random selection – for example, it could randomly choose an integer between 0 and 9, and act dishonestly if the chosen integer is less than $10P(d)$, and honestly otherwise. The expected values for each of those sellers in an ideal trust model would then be $1 - P(d)$, such that a seller for which $P(d) = 0.3$ would be expected to be assigned a trust value of 0.7.

In regards to the honesty of advisors, the simulations should be run in two sets: one where advisors are mostly reporting honestly when rating sellers, and one where advisors are mostly lying about their experiences. For example, in the first set, advisors might lie in their seller ratings 30% of the time; in the second set, this probability might increase to 60%.

At this point, depending on which techniques are desired to be used, several options for the appropriate parameters – either *max_nbors* or the trust threshold, and optionally the minimum N_{RE} – should be identified for the simulations. As a guideline, our earlier results testing PTM and TRAVOS in e-commerce seem to suggest that the optimal value for *max_nbors* is likely to be in the range of 20% to 40% of the total size of the advisor population, while if thresholding is used, the threshold value should be set between 0.5 and 0.6. If referrals are used, our results suggest that the optimum value for the minimum N_{RE} is likely to be between 1 and 5. These values might potentially vary for other models and other domains, although we would suggest that for thresholding, 0.5 should be an intuitive minimum threshold – if the trustworthiness of the advisor is below 50%, it should not be

worth considering.

The simulation for each option being considered should be repeated several times (at least five to ten) and then averaged, to help negate the effect of any outliers that appear. Once the simulations are complete, find the mean absolute error – the average of the absolute differences between the expected and experimental trust values for all of the sellers – for each simulation. The option yielding the lowest mean absolute error will then indicate the optimal parameter(s).

We now provide an overview of the application of our modifications in order to test their effectiveness, in pseudo-code format, as Algorithms 3 and 4. This is one example of how to apply the procedure; refer to the information provided above for more details as to how to set the specific parameters outlined in these algorithms.

5.2 Selecting Appropriate Parameters

In our work, we have noted that either using trustworthiness thresholding or setting a maximum number of advisors will provide a modest improvement to the accuracy of the trust model. Moreover, in cases where the size of the advisor network is very small, using referrals may help to further improve the accuracy of this model.

Our results indicate that the parameters to be used should be modestly sized – allowing a reasonable number of advisors to be used, without including a large number of advisors that contribute little to the calculations of the trust model. Additionally, our results suggest how to set the actual value of the parameters. In particular, they indicate that the range of 0.5 to 0.6 is optimal for threshold parameters, while a *max_nbars* parameter should be set as roughly 20% to 40% of the total size of the population.

5.2.1 Comparison with Collaborative Filtering

Interestingly these results seem to diverge from those found for collaborative filtering (CF), as in [9]. We first describe their experimental setup: For both the thresholding

Algorithm 3 Testing Modifications with a Trust Model: Determine Advisor Trusts

```
1: advisorLying = 0.3; {fraction of advisors that lie}
2: numAdvisors = 80; {number of advisors}
3: numSellers = 40; {number of sellers}
4: numDays = 120; {number of interactions by each buyer or advisor with a seller}
5: b = {b0, b1, b2, ..., bnumAdvisors}; {b0 is the true “buyer”, others are advisors}
6: s = {s1, s2, ..., snumSellers}; {sellers}
7: {Select dishonest advisors}
8: for i = 1 to numAdvisors do
9:   {if advisor i is among the fraction of advisors that should lie, set as lying}
10:  if (i - 1) mod 10 < 10 × advisorLying then
11:    bi.setLying(True);
12:  end if
13: end for
14: {Select categories for sellers – each category represents percentage of dishonest behaviour}
15: dishonesties = {0, 0.1, 0.2, 0.3, ..., 0.9}; {for example}
16: for i = 1 to numSellers do
17:   {get and set dishonesty based on seller’s category}
18:   dishonesty = dishonesties[i mod dishonesties.length];
19:   si.setDishonesty(dishonesty);
20: end for
21: for i = 1 to numDays do
22:   {Run interaction simulations}
23:   for j = 0 to numAdvisors do
24:     {Repeat for buyer and all sellers}
25:     sellerId = -1; {Select a seller}
26:     while sellerId == -1 or (referralsUsed and bj.hasUsed(ssellerId) do
27:       sellerId = random integer in range [1, numSellers];
28:     end while
29:     bj.addUsed(ssellerId);
30:     {Is the seller lying this time?}
31:     sellerLieState = random integer in range [0, 9];
32:     sellerLying = (sellerLieState < 10ssellerId.getDishonesty());
33:     {Determine rating buyer will assign to this interaction}
34:     if bj.isLying() xor sellerLying then
35:       rating = 0;
36:     else
37:       rating = 1;
38:     end if
39:     bj.addRating(ssellerId, rating);
40:   end for
41: end for
42: {Determine how much the buyer trusts each advisor, then sort}
43: for i = 1 to numAdvisors do
44:   determine advisorTrust for bi as applicable using the selected trust model;
45:   b0.setAdvisorTrust(bi, advisorTrust);
46: end for
47: b0.sortAdvisorTrusts(); {in descending order}
```

Algorithm 4 Testing Modifications with a Trust Model: Determine Seller Trusts

```
1: {Variables carried over from Algorithm 3}
2:  $b = \{b_0, b_1, b_2, \dots, b_{numAdvisors}\}$ ;  $\{b_0$  is the true “buyer”, others are advisors}
3:  $s = \{s_1, s_2, \dots, s_{numSellers}\}$ ; {sellers}
4:  $numSellers = 40$ ;  $numDays = 120$ ;
5: {Get advisor network}
6:  $advisorNetwork = \{\}$ ;
7:  $advisorTrusts = b_0.getAdvisorTrusts()$ ; {for now, we will assume it’s an array}
8: if usingMaxNbors then
9:   for  $i = 1$  to  $maxNbors$  do
10:     $advisorNetwork.add(advisorTrusts[i].buyer())$ ;
11:   end for
12: else if usingThreshold then
13:    $i = 1$ ;
14:   while  $advisorTrusts[i].trust() \geq threshold$  do
15:     $advisorNetwork.add(advisorTrusts[i].buyer())$ ;
16:     $i++$ ;
17:   end while
18: end if
   {Model the trust for each seller}
19: for  $i = 1$  to  $numSellers$  do
20:   for  $j = 1$  to  $numAdvisors$  do
21:     $advisor = b_j$ ;
22:    if usingReferrals and  $b_j.getNumRatings(s_i) < minExperience$  then
23:      set  $advisor$  to be a referred advisor using Algorithm 2
24:    end if
25:    if  $advisor \neq null$  then
26:      determine contribution of  $advisor$  to  $b_0$ ’s trust model for  $s_i$ ;
27:    end if
28:   end for
29:    $sellerTrust_i = b_0$ ’s overall trust in  $s_i$  given contributions of advisors;
30: end for
   {Determine average seller trust for each category}
31:  $totalAbsError = 0$ ;
32: for  $i = 1$  to  $dishonesties.length$  do
33:   {category}
34:    $categoryTotal = 0$ ;
35:    $numPerCategory = numSellers/dishonesties.length$ ;
36:   for  $j = 1$  to  $numPerCategory$  do
37:     {seller within category}
38:      $categoryTotal = categoryTotal + sellerTrust_j$ ;
39:   end for
40:    $categoryAvg_i = categoryTotal/numPerCategory$ ;
41:    $categoryAbsErr_i = abs(categoryAvg_i - dishonesties[i])$ ;
42:    $totalAbsError = totalAbsError + categoryAbsErr_i$ ;
43: end for
44:  $meanAbsError = totalAbsError/dishonesties.length$ ; {“overall” error for this simulation}
```

and *max_nbor*s methods, different values were tested experimentally, in combination with various other factors considered in that paper, in order to determine the best value, if any. In essence, given an existing data set (specifically, a subset of the MovieLens film-rating database), a small fraction of those ratings were first removed. Those ratings would then be regenerated using a CF prediction algorithm using some combination of the factors being tested, based on the remaining data. These generated ratings were then compared with the original results. The main criterion observed was the mean absolute error (MAE) of the predicted ratings – that is, the average of the absolute values of the differences between the actual and expected ratings. No formulae were offered to suggest an analytical approach to determining the appropriate threshold. We also note the authors’ caveat that although they say they had reason to believe their results were generally applicable to other domains besides film, they did not, at that point, have empirical evidence to prove this was the case.

The results in [9] suggested that thresholding would have little usefulness in a CF system. In fact, for the MovieLens data set examined, correlation thresholding yielded declines in both coverage and accuracy (that is, an increase in the MAE) compared to a non-thresholded algorithm. Specifically, not using thresholding in this CF system would yield an MAE of about 0.7528, while applying any threshold would result in an MAE that was higher – sometimes only marginally, but going up to approximately 0.78 for a threshold of 0.5. We note for completeness that [30] used a similar experimental method to [9] for determining the best threshold for their music recommender system, and had no further insight with regards to an analytical approach.

On the other hand, the results for *max_nbor*s indicated the benefits of a careful application of this approach. Recall that using an unrestricted network in this CF system would yield an MAE of about 0.7528. The results indicated that applying a *max_nbor*s value between 20 and 80 (out of a population of 943 agents) would result in lower MAE, with *max_nbor*s = 60 yielding the lowest average MAE of 0.7508 (although slightly smaller *max_nbor*s values, between 20 and 40, would generally be about the same in terms of performance). On the other hand, using a *max_nbor*s value of 5 or 10 would show increased MAE (of 0.7836 and 0.7605 respectively), while *max_nbor*s = 100 (the highest value tested) would show no improvement compared to the unrestricted network.

It was concluded that a neighbourhood of 20 to 50 users would be a “reasonable” size to provide an acceptable level of accuracy, providing an appropriate balance between sufficient coverage and eliminating inaccuracies. Again, this result was specifically tested with a population of 943, although the authors claim, based on experiments (not fully documented in the paper) using all 80,000 users in the MovieLens database, that a similarly-sized neighbourhood should be sufficient for most “real-world” scenarios, without regard to population size.

This is an interesting contrast in our findings in Chapter 3, in that we encountered very good results when applying trustworthiness thresholding, whereas for the *max_nbor*s approach, setting the *max_nbor*s parameter as a proportion of the total advisor population size (and not as a static value, or one within a small range) seemed to work best. Thus it seems clear that there are strong distinctions between collaborative filtering and trust modelling, at least in regards to how to select the best size for a social network in each application.

5.2.2 Comparison with Advisor Referrals

Yu and Singh also explored, in a sense, the effects of varying of the number of agents used in trust modelling. In [40], they discuss how to find “witness” agents that have interacted with some goal agent A_g , and using information about the witnesses’ experiences to determine the trustworthiness of that goal agent. To do so, they construct a trust network representing the relationships between agents. Ultimately, agents in the trust network provide referrals to other agents that are acquainted with A_g , which then comprise the set of witnesses. (We discussed this scenario in greater detail in Section 2.3.)

For the experimental validation of this research, each agent was assigned a set of 16 acquaintances, out of a total of 100 agents in the system. Of these acquaintances, four highly-trusted and highly-sociable would be designated as neighbours. Each agent was also assigned a “branching factor”, between one and four, specifying how many referrals from its set of neighbours it could provide at any one time.

Because only a low proportion of agents would be acquainted with any other agent,

and due to limits on the depth of the trust network, the number of witnesses that would be found would generally be quite low, typically within the range of 1 to 6 – much smaller than the values considered in [9] (and that we examined in Chapter 3). However, the number of witnesses found would increase as the branching factor increased, and (to a lesser extent) as the depth of the trust network increased. When comparing the number of witnesses generated to the average rating error, it was found that the prediction accuracy would improve slightly, but not significantly, when more witnesses had been found.

However, due to the differences in their model in terms of finding witnesses, the fact that the number of witnesses would be found after the fact (as opposed to being pre-determined), and the use of a large number of simulation cycles using the scenario described above to generate this trust – a luxury that does not apply to the trust models we have examined – these results are not easily comparable to those presented in this thesis.

Nevertheless, the experimental results from this research seem to match our findings in that, given a relatively small proportion of the advisor population that is regularly consulted (i.e. the acquaintances in [40], or the advisor network in our approach), using referrals is helpful in finding specific advisors that will be useful in evaluating a target agent.

With regards to our referral mechanism, we noted in Section 4.1 that there is a limit on the number of levels of advisors through which this algorithm will search when looking for an acceptable replacement advisor. Presently this is set as $\lceil \log_k(|B|) \rceil$, where B is the set of all buyers (advisors) in the system, and k is the number of advisors in the buyer’s own advisor network. This is intended to be a prediction of the approximate number of advisor-network levels that one would need to examine in order to search all nodes. In future work, we might examine whether using a different value might produce improved results for referrals. A smaller value might help to reduce the amount of computation required to perform a referrals search, particularly if some of the same advisors are being redundantly examined multiple times. On the other hand, searching a larger number of levels could ensure a more complete search for suitable advisors, thus ensuring the buyer has more information in modelling the reputation of a seller.

5.3 Alternative Referral Trust Computation Methods

In our discussion of referrals in Chapter 4, we made the simplifying assumption that buyers would continue to use their own previously-calculated trust models to represent the trustworthiness they should hold in each referred advisor.

This method was chosen as it seemed to be the most efficient means of setting the trust of each referred advisor, given that these trust values had already been calculated, whereas using another method might require additional time-consuming calculations. As noted above, this method will provide some positive results. However, it has some weaknesses: first, it will not take into account the referring agents' opinions of the referred agents. Moreover, in the event that multiple agents make referrals to a single agent, the existing method will not take this into account, even though the multiple referrals may indeed suggest that this single agent should command more trust.

However, trust propagation, or defining an agent's trust in another agent as a function of all of the connections between them, has been studied extensively of late [15][8], and might provide an even more accurate trust model, especially in a larger network.

In this section we will discuss some of the most relevant past work on this topic. We will then propose several possible alternative representations of the trustworthiness of referred advisors, and then outline potential future work on the matter.

5.3.1 Related Work

At least two works have attempted to implement methods for similar trust systems which take into account the two weaknesses identified above. In [41], the authors discussed the use of *weighted referral graphs* to decide which referred agent to use. In such a graph, agents are linked based on the agents that each agent refers to for a given query Q . The requesting agent and the agent or agents to which it is ultimately referred will be linked, indirectly, by one or more *referral chains*, each containing one or more intermediary referring agents.

Each referral (edge) and agent (vertex) is assigned a weight, with the requesting agent initially being assigned a weight of 1, and referrals being assigned weights representing the

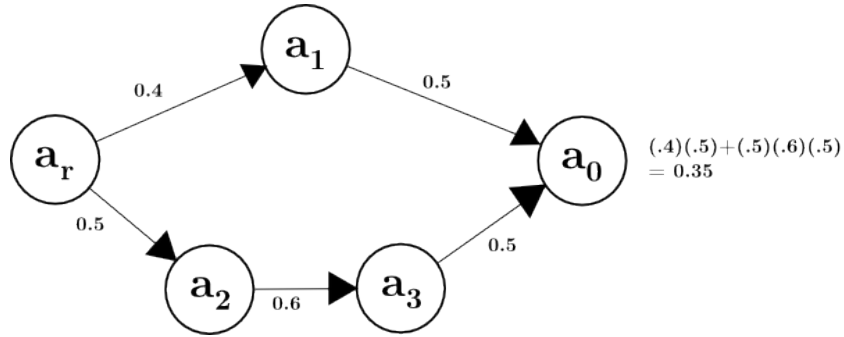


Figure 5.1: An example of the usage of weighted referral graphs used in [40].

relative trust the the corresponding referring agent has in a referred agent. However, as we understand the model, there are no restrictions on the value of any of these weights.

The weight of every other agent along a referral chain is the product of the edge weights between the requesting agent and the instant agent. The implicit assumption (i.e., not explicitly stated in [41]) seems to be that if we assign a weight of x to an agent a_1 , and a_1 likewise assigns a weight of y to a_2 , then the weight that we should apply to a_2 should be the product of these two values, or xy – and so on with additional agents if required. If there are multiple referral chains leading to a single agent, the weight will then be the sum of the individual weights of these referral chains, as the fact that multiple referral chains exist should indicate that the referred agent is more likely to be trustworthy.

For example, assume that an agent a_r receives two referrals to a particular agent a_0 , one through agent a_1 , and another through agents a_2 and a_3 . The weight of the a_r to a_1 referral is 0.4, and that of the a_1 to a_0 referral is 0.5, leading to an overall weight of $0.5 \times 0.4 = 0.2$ for that chain. If the weights for the chain going through a_r , a_2 , a_3 , and a_0 are 0.5, 0.6, and 0.5, the chain will likewise have an overall weight of 0.15. The weight of a_0 will then be the sum of these two chains, or 0.35. (A diagram of this example is provided in Figure 5.1.)

However, we note that a direct application of this approach to trust models such as PTM or TRAVOS would be problematic. In this aggregation approach, if there are multiple chains all referring to the same advisor, these weights will be summed together without any normalization to yield the overall weight for that agent. Although this ensures that

agents receiving multiple referrals will be weighed more heavily, it also means that the agent weights w will not necessarily be restricted to the range $0 \leq w \leq 1$. This makes sense in the context of *weights*, which need not necessarily be bounded to such a range. However, given that such a range is assumed for trust values such as those in PTM, direct application is not appropriate.

Later work from the same research group [35], based in part on the work of Jøsang [12], provided the basis of a more rigorous approach to combining trust values by defining two operators, concatenation and aggregation. To do so, trust is first defined as a triple $\langle b, d, u \rangle$, with the three values respectively representing the belief (positive trust), disbelief (negative trust), and uncertainty associated with a particular entity; this in turn is derived using a transform $Z(\langle r, s \rangle)$, where the pair of numbers $\langle r, s \rangle$ represents the positive and negative evidence for that entity.

We first look at their method for concatenating trust along a path. Suppose that agent a_r holds trust $M_1 = \langle b_1, d_1, u_1 \rangle$ in another agent a_1 . Meanwhile a_1 holds trust $M_2 = \langle b_2, d_2, u_2 \rangle$ in a_2 . Then, using the concatenation operator \otimes , a_r 's trust in a_2 may be represented as $M = M_1 \otimes M_2 = \langle b, d, u \rangle$ where:

$$\begin{aligned} b &= b_1 b_2 \\ d &= b_1 d_2 \\ u &= 1 - b_1 b_2 - b_1 d_2 \end{aligned}$$

We explain each of these values in plainer terms as follows:

- As in [41], a_r 's belief in a_2 is equivalent to the intersection of a_r 's belief in a_1 , and a_1 's belief in a_2 , and thus the value of that belief is the product of these two belief values.
- Similarly, a_r 's disbelief in a_2 is equivalent to the intersection of a_r 's belief that a_1 is trustworthy, and a_1 's disbelief in a_2 's trustworthiness.

- The concatenated uncertainty is calculated as being anything other than the two joint beliefs just noted. The key insight in this regard is that if a_r does not believe a_1 , then a_r believes that a_1 is uncertain regarding whether a_2 is trustworthy; thus this component (d_1) is not included in the calculation of d . [12].

To combine trust models from multiple agents (or paths), an aggregation operator, \oplus , is suggested. Essentially this is performed by finding $\langle r, s \rangle = Z^{-1}(\langle b, d, u \rangle)$ for each of the trusts, summing the respective r and s values, and finding the overall triple $\langle b, d, u \rangle$ given these sums.

Even more recent work from that group [8] suggested an apparently improved method for concatenation. In essence it discounts the evidence in the later “link” by the belief held in the earlier one – that is:

$$\begin{aligned}
 M &= M_1 \otimes M_2 \\
 &= \langle b, d, u \rangle \\
 &= Z(\langle b_1 r_2, b_1 s_2 \rangle)
 \end{aligned}$$

where $\langle r_2, s_2 \rangle = Z^{-1}(M_2)$. The aggregation operator from the earlier paper was carried over as-is in the more recent work.

Certainly, the concatenation in [35] of the “positive” trust value b of a chain as the product of the individual values of b seems to be consistent with the other works studied in this section. At first glance, both the aggregation operator in that work and the revised concatenation formula in [8] also appear potentially useful for our purposes.

However, we note that both PTM and TRAVOS already have their own methods of translating the positive and negative evidence obtained through the interactions between agents into a trust value. These methods are significantly more complex than the transformations used in these papers, and are not easily invertible, meaning we cannot readily insert them into the operators described therein. Although we could still use these methods by applying the transformations referred to above, the resulting $\langle r, s \rangle$ pairs would likely

bear no resemblance to the actual data obtained, and as a result we would be reluctant to apply this method directly.

The same group has also suggested a means of trust-based recommendations based on graph similarity [7]. In this method, a trust network is modelled as a directed weighted graph of agents, and may be represented by an *adjacency matrix with trust*, in which values are zero if no trust relationship exists, or the trust value if such a relationship does exist (as opposed to the standard adjacency matrix which contains only zeroes and ones). From this we can calculate a similarity matrix between this trust network and a standard structure graph representing the desired relationship – that is, if a_1 trusts a_2 and a_2 trusts a_3 , then a_1 also trusts a_3 . Essentially, higher similarity scores between agents should indicate more useful advisors. That said, the paper gives no additional insight on how much the recommended agent should be trusted, and so it is not directly comparable to the other metrics examined in this section.

The EigenTrust algorithm [16] also uses aggregation of local trust values in a slightly different manner to that explored in [41]. It first defines a *local trust value*, s_{ij} , held by a peer i with respect to another peer j , as the number of satisfactory transactions that i has had with j (or $sat(i, j)$), subtracted by the number of unsatisfactory transactions between the two peers ($unsat(i, j)$):

$$s_{ij} = sat(i, j) - unsat(i, j) \quad (5.1)$$

This value may then be converted to a normalized value, c_{ij} , as follows:

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)} \quad (5.2)$$

Note that the special case of $\sum_j \max(s_{ij}, 0) = 0$ – that is, peer i has not previously had any interactions – is handled in EigenTrust by setting $c_{ij} = 1/|P|$, where P is the set of peers known to be pre-trusted by peer i , if $j \in P$ and $c_{ij} = 0$ otherwise.

The overall trust that peer i holds in another peer k , based on i asking its friends, can thus be represented as follows:

$$t_{ik} = \sum_j c_{ij}c_{jk} \quad (5.3)$$

This method can thus effectively aggregate trust values along multiple paths – although as it uses a weighted average, unlike the proposal in [41], it will not provide any real “bonus” for multiple referrals. On the other hand, it does not directly handle “referral” chains using more than one intermediate agent – the intent is to get information about a single target agent, a_g , based on the reported direct experiences of all agents (other than the requesting agent and a_g itself) with a_g .

To handle asking the friends of friends (and so on), EigenTrust uses a slightly more complex form. We first rewrite Equation 5.3 in matrix notation, by defining a matrix $C = [c_{ij}]$, a vector \vec{t}_i containing the values t_{ik} , and \vec{c}_i as the normalized local trust vector of peer i . Thus:

$$\vec{t}_i = C^T \vec{c}_i \quad (5.4)$$

If we wish to consider friends of friends, we could compute $\vec{t}_i = (C^T)^2 \vec{c}_i$, and so on; ultimately, if we wish to compute the trust values considering the entire network, we would compute $\vec{t}_i = (C^T)^n \vec{c}_i$ for $n = \text{large}$. The eventual determination of this vector in EigenTrust requires additional computations that are not within the scope of this current discussion. Nevertheless, we will provide a simple example that demonstrates the functionality of the simplified “asking friends” version discussed above.

Consider the network shown in Figure 5.2. Each edge points from one agent that has made requests to another agent which received the particular set of requests. Adjacent to the edge, the numbers of satisfactory (*sat*) and unsatisfactory (*unsat*) requests are indicated. We assume, for purposes of simplification, that the interactions shown represent the only relevant interactions that have occurred among this set of agents. As in the earlier example, one agent a_r wishes to determine how much it should trust a target agent a_0 . Given that a_r has not previously interacted with a_0 , using Equation 5.2 it would calculate its normalized local trust value for a_0 to be $c_{r0} = 0$.

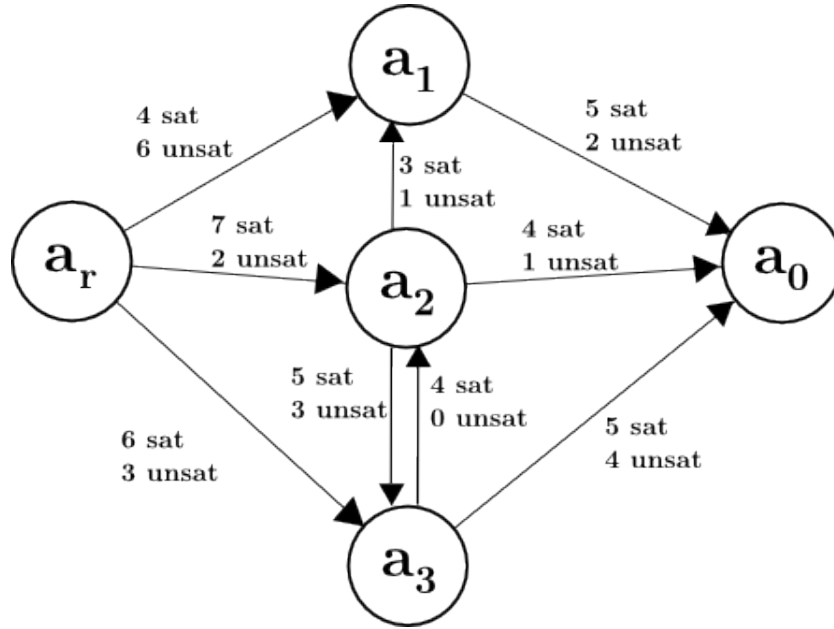


Figure 5.2: An example to illustrate a simplified EigenTrust “asking friends” mechanism.

If a_r wishes to calculate its trust in a_0 based on asking its friends, it will first calculate the appropriate “raw” and normalized local trust values, using Equations 5.1 and 5.2. The pertinent values are shown in Table 5.1.

Then a_r ’s model of the trustworthiness of a_0 would be:

$$\begin{aligned}
 t_{r0} &= c_{r1}c_{10} + c_{r2}c_{20} + c_{r3}c_{30} \\
 &= (0)(1) + (0.625)(0.429) + (0.375)(0.2) \\
 &= 0.343
 \end{aligned}$$

5.3.2 Proposed Alternatives

Based on this research, we now suggest five alternative methods of modelling the trustworthiness of referred advisors, for future consideration and comparison. We reiterate that these alternatives only affect the specific trustworthiness *values* assigned to each referred

Table 5.1: Calculations of Normalized Local Trust Values for Simplified EigenTrust Example

a_i	a_j	sat	$unsat$	s_{ij}	$\max(s_{ij}, 0)$	$\sum_j \max(s_{ij}, 0)$	c_{ij}
a_r	a_1	4	6	-2	0	8	0
	a_2	7	2	5	5		0.625
	a_3	6	3	3	3		0.375
a_1	a_0	5	2	3	3	3	1
a_2	a_0	4	1	3	3	7	0.429
	a_1	3	1	2	2		0.286
	a_3	5	3	2	2		0.286
a_3	a_0	5	4	1	1	5	0.2
	a_2	4	0	4	4		0.8

advisor; alternative means of actually *choosing* these advisors, beyond the method noted in Section 4.1, are beyond the scope of our current work.

The first method is using the buyer’s original trust value for the referred advisor in question – the method we used throughout Chapter 4. This is true to the “personalized” approach that motivates PTM but, as noted, does not take into account the trust that other advisors have in the referred advisor, perhaps reducing the effectiveness of such referrals.

The second method is simply using the product of all of the trust values in the first referral chain found leading to the referred advisor. For example, if b is referred to a_0 via a_1 and a_2 , and we define t_{b1} to be the existing trust that b has in a_1 (and similarly for the remaining links), then $t_{b0}^* = t_{b1}t_{12}t_{20}$.

The third method is a simple average of the two values generated from the first and second methods, i.e. $\frac{t_{b0} + t_{b0}^*}{2}$.

For the fourth method, we return to our discussion of EigenTrust. While the trust measure used in that mechanism differs somewhat from that used in PTM or TRAVOS – in the latter methods, the trust values used are already normalized to a value between zero and one – we feel that this nevertheless provides some inspiration as to how we could normalize the trust values held when aggregating multiple trust values for the same advisor. Specifically, we will seek to determine a weighted average of the trust values for the referred

advisor, in which these trust values are weighted by how much the buyer trusts the chain that referred this advisor.

To do so, we first define the concept of *chain trust* as the overall trust that the buyer holds in the chain of advisors, not including the final “link” (referral) to the referred advisor, a_k . Thus it is equal to the product of the trust values in the chain except for that of the ultimate referral to a_k . For example, in the example above in which b is referred to a_0 via a_1 and a_2 , then the value of the chain trust for that referral chain is equal to $t_{b1}t_{12}$.

Consider that an advisor a_i has received one or more referrals to some other advisor a_k and is seeking to update the trust it should have in a_k , given a_i 's existing trust in a_k , which we will define as t_{ik} , as well as all of the chains of advisors referring to a_k and the trust values in each of those chains. We define each j as representing one such chain of advisors, such that t_{ij} is the chain trust that a_i holds in j . We could consider a_i 's own prior trust value t_{ik} to be the end of such a chain for which $t_{ij} = 1$, as we presume that a_i has complete confidence in itself.

We can then define one method of determining an updated trust value, t'_{ik} , as follows:

$$t'_{ik} = \frac{t_{ik} + \sum_j t_{ij}t_{jk}}{1 + \sum_j t_{ij}} \quad (5.5)$$

Such a mechanism would accept additional referral chains for a particular advisor that had already been accepted as a referral, if the said advisor was referred multiple times. This could be performed by keeping track of a numerator and denominator for each referred advisor, and calculating the final value once the referral process is complete. At this point we contemplate that this would only apply to referred advisors; that is, if an advisor a_m was part of the buyer's original advisor network, only the previously-calculated value t_{bm} will be used, regardless of how many times it is re-considered as part of the referral algorithm.

The final method we would offer is a simplified version of the fourth method, considering only the first referral chain found. In other words, this would be a weighted average of t_{b0} and t_{b0}^* :

$$\frac{t_{b0} + t_{b0}^*}{1 + t_{b0}^*}$$

Table 5.2: Summary of Proposed Referral Trust Weighting Methods

Method	Trust Metric(s) Used	Averaging Method
1	Buyer's direct trust	None
2	First referral chain	None
3	Buyer's direct trust First referral chain	Simple average
4	Buyer's direct trust Multiple referral chains	Weighted using chain trust
5	Buyer's direct trust First referral chain	Weighted using chain trust

We summarize the methods outlined above, and their respective features, in Table 5.2.

Since the proposals in this thesis have been discussed to this point in the context of probabilistic trust models, we also make clear that in adopting any of these methods beyond the first one, which simply uses the exact values previously calculated under the applicable trust model, we do not assume that the values represent probabilities that an agent is trustworthy. The second method does define trust in a referred agent as the product of several other probabilities, but this would only be a probability itself if we could prove the component probabilities to be independent of each other, which is beyond the scope of the present work. The remaining methods use averages (simple or weighted) of trust values derived using the first two methods, and therefore would not necessarily represent probabilities.

Examples

We now turn to demonstrating the use of the five methods identified above using a demonstrative example. A buyer agent b has an advisor network consisting of three advisor agents, a_1 , a_2 , and a_3 . Four additional agents, a_4 through a_7 , are included in successive advisor networks. The pertinent relationships between the agents are shown in Figure 5.3. Although some of the agents may appear to have small advisor networks, we assume that any additional relationships may exist in addition to those indicated here, but that these are not pertinent to our calculations (for example, a_2 having b as an advisor).

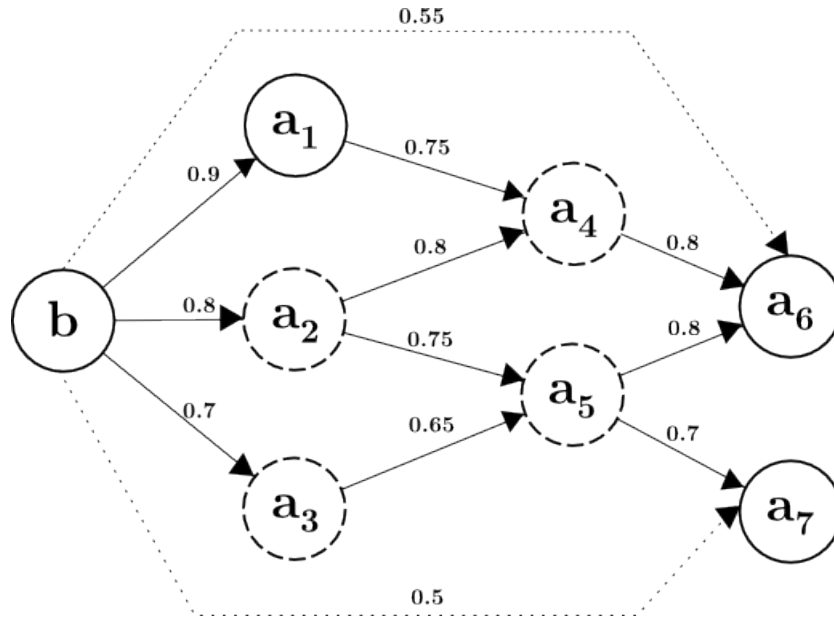


Figure 5.3: The pertinent relationships and trust values held in the examples discussed in Section 5.3.2.

Arrows are used to indicate that the agent at the start of the arrow holds the indicated trust value in the agent being pointed at. Solid-line arrows indicate that the agent being pointed to is included in the advisor network of the agent at which the arrow originates; dashed-line arrows used to indicate pertinent values that were previously calculated, where the agent being pointed to is not included in the other agent's advisor network. For example, it is shown that b calculated a trustworthiness value of 0.5 for agent a_7 , but has not included it in its advisor network, whereas it has selected a_1 , a_2 , and a_3 . Finally, for the advisors, a solid-line border indicates that the agent has had sufficient experience with the seller under consideration to be included in determining the trustworthiness for that seller; a dashed-line border specifies that the experience is insufficient to be included in these calculations.

Regardless of the referral weighting method used, the process of selecting the advisors is the same – that is, the method outlined in Section 4.1. The buyer will examine each of the agents in its advisor network, from most trustworthy to least, to determine whether each is suitable to be included in calculating the trustworthiness of the specified seller. If not,

the advisor's own advisor network will be searched for an acceptable replacement advisor. If this fails to yield a replacement – because all of those advisors were either unsuitable themselves, or already counted as part of the advisor network – then the advisors of the advisors just considered will be examined, and so on until either a suitable replacement has been found, or some pre-determined (currently arbitrary) maximum number of levels has been reached.²

In this case, advisor a_1 , with the highest trustworthiness (0.9), will be the first to be examined, and will be accepted due to having the required amount of experience with the seller. It will be assigned 0.9 as its trustworthiness value regardless of the referral weighting method chosen later in this example.

However, the next advisor to be considered, a_2 (with trustworthiness 0.8), will be rejected as having insufficient experience. In searching for a replacement advisor, the agents in a_2 's own advisor network, a_4 and a_5 , will be similarly rejected. Thus the buyer will continue with the next level, beginning with the advisor network of a_4 , which a_2 had determined to be most trustworthy (0.8) among the agents in its advisor network. The first agent in a_4 's network will be a_6 , which it trusts with value 0.8. Since a_6 has had sufficient experience with the seller, the buyer will accept a_6 as the replacement agent for a_2 when dealing with this particular seller.

Finally we turn to a_3 , the last agent in b 's advisor network, which is trusted by b with value 0.7. Here, once again, we recognize that a_3 will not have sufficient experience to be included in the current calculations. Looking at a_3 's advisor network, we see that it too includes a_5 , which (as discussed in the previous step) will have insufficient experience to be used as a replacement advisor. This brings us to a_5 's advisor network. The buyer first looks at a_5 's most trusted advisor, a_6 – although the buyer cannot accept it again as a replacement advisor, it may be able to obtain additional useful information (depending on the weighting method used). Finally we see that a_5 also has a_7 in its advisor network. As a_7 also has had sufficient experience with the seller under consideration, it can now be accepted as the replacement advisor for a_3 , completing the referral process.

²If and when an acceptable advisor is found during these searches, that advisor will be selected, and the search will cease at this point – that is, only the first acceptable advisor found will be selected.

Using **Method 1**, the buyer relies on the values it had previously calculated when it considered – but rejected – using a_6 and a_7 in its own advisor network – which, as indicated in Figure 5.3, are 0.55 and 0.5, respectively. It can therefore immediately use these values in its calculations of seller trustworthiness.

With **Method 2**, the buyer disregards these previously-calculated values, and instead relies on the chain of advisors used to find each of these new advisors – using the product of the trust values of the individual links. (This could be accomplished, for example, by calculating the value recursively on the chain once a suitable replacement is found.) For the former, the chain is $b \rightarrow a_2 \rightarrow a_4 \rightarrow a_6$, so the value used will be $0.8 \times 0.8 \times 0.8 = 0.512$. For the latter advisor, the corresponding chain is $b \rightarrow a_3 \rightarrow a_5 \rightarrow a_7$, leading to the value used being $0.8 \times 0.75 \times 0.7 = 0.42$.

Method 3 will be a simple average of the values used in the previous two methods. Thus a_6 would be assigned a trustworthiness of $(0.55 + 0.512)/2 = 0.531$. Likewise, a_7 would be assigned a value of $(0.5 + 0.42)/2 = 0.46$.

In **Method 4**, we use a weighted average of the agent’s direct trust in the advisor and all advisor chains leading to that agent found during the referral process. The calculations are therefore more complex: The implementation must keep track of the overall trust corresponding to that chain (as in Method 2), which will be used in the numerator, as well as the “chain trust” leading up to the final link, which will go into the denominator. (It could thus, for example, keep a running total of both the numerator and denominator for each referred agent as new referral chains are found.)

With respect to a_6 , there are, in addition to b ’s direct trust value, two referral chains found during this process: $b \rightarrow a_2 \rightarrow a_4 \rightarrow a_6$, and $b \rightarrow a_3 \rightarrow a_5 \rightarrow a_6$.

- For b ’s direct trust value in a_6 , the numerator component is simply that value: 0.55. The denominator component will trivially be 1 – since b will have no reason to doubt itself, it will have complete confidence in its prior calculation of the direct trust value.
- For the referral chain passing through a_2 and a_4 , the numerator component will be 0.512 (as calculated in Method 2). The denominator component is calculated based on all of the links save the final one, and thus it will be $0.8 \times 0.8 = 0.64$.

- Finally for the referral chain that goes through a_3 and a_5 , the numerator component will be calculated following the procedure in Method 2: $0.7 \times 0.65 \times 0.8 = 0.364$. The denominator component will be calculated similarly as for the previous referral chain: $0.7 \times 0.65 = 0.455$.

The final calculation of the trustworthiness value for a_6 would be as follows:

$$\frac{0.55 + 0.512 + 0.364}{1 + 0.64 + 0.455} = 0.6807$$

In a similar fashion we can show that the corresponding value for a_7 – combining the direct trust held by b and the single referral chain found via a_3 and a_5 – will be 0.575.

A casual reader observing these results may justifiably wonder why the values calculated for a_6 and a_7 under this method are larger than any of the individual trust values being combined. The reason for this is that Method 4 is intended to calculate a weighted average of the trust values directly held in these agents – and not of the overall chained trust values as calculated in Method 2. More specifically, a_4 and a_5 have each assigned a trustworthiness value of 0.8 to a_6 (which are then weighted by the “chain trust” values leading up to each final link). It therefore stands to reason that under this method, the updated trustworthiness of a_6 will be between 0.55 (b ’s direct trust) and 0.8 – which it is, at 0.6807.

Finally we look at **Method 5**, which is essentially a simplified version of Method 4 making use of only a single referral chain. For a_6 , this means we ignore the components corresponding to the referral chain through a_3 and a_5 , leading to the following calculation:

$$\frac{0.55 + 0.512}{1 + 0.64} = 0.6476$$

Since our referral mechanism only found one referral chain to begin with for a_7 , the trustworthiness value for that advisor under Method 5 will be the same as for Method 4: 0.575. We reiterate at this point our comments from Method 4 regarding the calculated values being above those of either of the individual trust values being averaged, which also apply to Method 5.

Discussion

It seems clear that the choice of weighting method may well have a significant impact on how trust is modelled when using referrals – and hence upon the accuracy of trust modelling when referrals are applied. At this time we opt to leave an experimental comparison among these methods to future work. However, we believe that testing these five methods should be fairly straightforward: Given one or more scenarios where the other parameters (i.e. size of the advisor population, percentage of lying advisors, *max_nbor*s or thresholding parameter, and N_{RE} value if applicable) are held constant, test all five of these methods using the testing procedure documented in Section 5.1.1, and determine which one yields the lowest mean absolute error in modelling the trust of sellers. A small handful of these scenarios – perhaps two values for each of the parameters just mentioned – should be sufficient to indicate the best method(s).

We believe there may also be some merit in further examination of recommendations based on graph similarity [7], in regards to selecting which agents to be referred. While we reiterate that this discussion does not really focus on trust propagation so much as using trust and similarity to select appropriate agents, this does seem to be a promising alternative means of *choosing* referred advisors, and could likely be implemented quite easily with the models discussed in this thesis, as an alternative means of obtaining referrals. We could compare this to our existing referral-selection mechanism – and potentially others – to determine whether modifying the mechanism has any effect on the accuracy of trust modelling. That having been said, we will also leave this to future work.

Finally, particularly for referral chains, we may also wish to consider the appropriateness of applying a decay factor to the trust value of a referred advisor based on the length of the chain that provides that referral. Similar to the forgetting factor used in the PTM and discussed in Section 2.1.1, the calculated trust value would be multiplied by λ^n , where λ is the decay factor ($0 \leq \lambda \leq 1$) and n is the length of the referral chain. This would reflect the belief that an agent that is referred by one of the requesting agent’s existing advisors should be more trustworthy than one referred by one of the advisor’s advisors (and so on). Although some decay does occur along a referral chain, it is possible, for instance, that an advisor referred by a chain of two agents, each trusting each other with a value of 0.8

($0.8^3 = 0.512$), would end up being assigned greater weight than an existing advisor with a trust value of 0.5. If such an occurrence is not desired, a decay factor will be required in order to prevent it.

5.4 Balancing Similarity and Reputation Modelling

In light of the fact that techniques used in CF recommender systems helped inspire the proposals documented in this thesis, we wish to touch briefly on some of the other research incorporating both trust modelling and recommendations.

A common theme in recent research has been recommender systems where trust is the primary criterion — although, in many cases, this has essentially taken the form of a CF-style recommender with the correlation-based weightings replaced with trustworthiness-derived values. In fact, some researchers have referred to this as “trust-based” or “trust-aware” CF [18][21], although more commonly, CF is solely used in regards to correlation-based recommenders.

Perhaps the best known version of these is the work by Massa et al. [1][21][22]. Their goal was to use trust-based recommendations to help resolve the data sparsity issue; that is, if a user (particularly a new user) has few items commonly-rated with other users, then it may be difficult to determine similarities with these users and hence make recommendations. Trust in this case was based primarily on explicit statements from agents, which could then be propagated to a pre-defined depth using a selected trust metric.

In their study [22], the more classical similarity-based CF recommender was tested against two trust metrics, the graph-based local trust metric MoleTrust [1] and the global metric PageRank [25], using the *epinions.com* dataset. As well, MoleTrust was tested using different levels of trust propagation. Their results indicated that even before trust propagation was applied, the mean average user error (MAUE, as defined in [22]) using trust as the criterion was significantly lower than with traditional CF (0.790 vs. 0.938). Adding propagation using MoleTrust served to improve the coverage, although the error increased slightly as well; meanwhile PageRank generally served only to increase the error.

The authors then proceeded to examine a system combining trust and user similarity, finding that although the coverage was better than for the two metrics used individually, the error was between that of the two “pure” metrics, indicating little benefit of such a combination.

A similar but more implicit approach to trust was taken in [18]. As with [22], trust is used in place of similarity to weight the ratings of other agents. Here, however, an agent’s measure of its trust in another agent is based primarily on the amount of information the second agent provides, but is also varied based on the quality of that information (that is, how closely their ratings matched). In choosing the best set of advisors for predictions, this study takes note of the thresholding approach but instead dismisses it on the basis that such a selection is a “difficult decision”. Instead they use k -nearest-recommenders (k NR), a variant of *max_nhors* (which is also known as k -nearest-neighbours, or k NN) that dynamically selects the best k neighbours that are able to provide information about a particular desired item.

The experimental results (run on the MovieLens dataset) indicate better coverage for the trust-based model as opposed to (similarity-based) correlation filtering, as well as lower prediction error for small values of k . However, as k increases, while accuracy improves for both methods, eventually the trust-based approach falls behind traditional CF. The authors hypothesize that since the neighbourhood in the trust-based method will grow to include more agents that the current user has had little experience with, greater error will result.³

We will consider future possibilities for linking collaborative filtering techniques and trust modelling, in light of this past work, in Section 6.2.6.

³Work by other researchers incorporating very similar methods has also been documented elsewhere [11][4].

Chapter 6

Conclusion and Future Work

6.1 Contributions

In this thesis, we have outlined three potential improvements to trust modelling – trustworthiness thresholding, maximum number of advisors, and advisor referrals – all of which aim to improve the accuracy of the recommendations for trustworthy agents derived from a buyer’s advisors. These three improvements can be used with different trust modelling methods, specifically the Personalized Trust Model and the TRAVOS model, as demonstrated in our study. We have also demonstrated that our proposed approach is sufficiently robust that it can be applied to offer improvements, even to large-sized populations of agents.

We have seen that either using trustworthiness thresholding or setting a maximum number of advisors will provide an improvement to the accuracy of the trust model. We have also seen that, in cases where the size of the advisor network is very small, using referrals may help to further improve the accuracy of this model.

In Sections 3.3.3 and 4.3.4, we evaluated our three optimization methods with a larger population. Our experimental results for PTM show that trustworthiness thresholding is not affected by the population size, while the proper parameter for the *max_nbor*s method can be selected as a proportion of the population size. For TRAVOS, however, it appears

that these parameters will need to be set separately for each population. Meanwhile, in larger populations, allowing for advisor referrals will have a greater benefit when used with more restricted advisor networks, as compared to the fairly modest improvements seen with smaller populations.

We note that other trust researchers have also explored ideas related to thresholding and *max_nhors*. As we discussed in Section 5.4, some trust researchers have examined the use of *kNR*, a variant of *max_nhors* (or *kNN*), in connection with a trust-based collaborative filtering recommender [18]. Meanwhile, the original version of PTM uses an implicit threshold in its computation of seller trustworthiness: it only makes use of “trustworthy” advisors, which are defined as those having trustworthiness values of 0.5 or above [42].

However, we should distinguish between the arbitrary (if intuitive) selection of parameters in [42], and to a lesser extent in [18],¹ and the more careful methodologies for selecting parameters outlined in [9] (for collaborative filtering) and in this work (for trust modelling). In both works, simulations were conducted using several different parameters for all the methods examined – thresholding, *max_nhors*, and (in our case) referrals – and the authors attempted to draw conclusions based on those results. Indeed, in considering the results in Section 3.3.1, it seems clear that the “intuitive” threshold used for the PTM in [42] was not the best one.

This seems to point towards the overall benefits of using a more principled methodology for selecting parameters in trust systems: while an arbitrary choice may work fine, there may be something better, given other parameters such as agent population size. Such a methodology may also help to demonstrate more robustly the benefits of the model and the specific parameter choices used therewith.

We have also observed other work relating to finding the “best” size of a social network. For example, Seth [29] examined the creation of *clusters* of users with close ties, such as interests in similar topics, in participatory-media social networks such as Orkut. He argued that the amount of contextualization that occurs in a community is proportional to the size of the cluster, and observed that a larger cluster size will generally result in a greater

¹The experimental verification for *kNR* included simulations using several values for *k*, but did not test the full range of possible *k* values exhaustively. Many of the subsequent experiments simply assumed *k* = 1, even though it was not the best performer in terms of minimizing error.

amount of information being generated in that cluster. At the same time, the author notes past observations suggesting that a group with more than 150 members might be unsustainable [3], and suggests that adjustments might be necessary above this point.

Our results seem to concur with this work insofar as increases to the size of the advisor network do not always result in improved results in terms of trust modelling accuracy: a very small group is not advisable, but after a certain point, further increases will only serve to reduce accuracy. However, that comparison should be taken with a grain of salt, in that the clusters discussed in [29] are of users that have consciously decided to join a group based on similar interests. By contrast, even though the agents in PTM or TRAVOS might represent actual human actors, it is these models, in combination with our techniques, that will determine the advisor network of each buyer agent, not the actors themselves.

The positive results outlined in this thesis do suggest that other researchers should be able to adopt these optimizations when seeking to improve their own trust models. Towards this end, we have also clarified, in Section 5.1, the experimental framework which can be used to derive appropriate parameter values.

6.2 Future Work

We believe this thesis provides a solid foundation for the usage of network limiting and advisor referral techniques for improving trust modelling in multi-agent systems. We identify below some of the open questions which remain with respect to the application of our work.

6.2.1 Parameter Selection

In Chapters 3 and 4 we conducted simulations aimed at not only verifying the usefulness of our techniques, but also at finding, in a methodological fashion, the best parameters to use for our techniques for the given situation. We expounded on this methodology in Section 5.1.

There are of course many ways we could improve on this methodology. For example, in our simulations, we normally chose to vary the *max_nbors* parameter in increments of 10, and the threshold parameter in increments of 0.05 or 0.1. It could be worthwhile to implement an iterative approach to finding the “best” value. For example, if we initially use increments of 10 and determine that *max_nbors* = 40 works best, we might repeat the simulations for increments of 2 between 30 and 50, and so on until the best value is found.² Identifying a precise “best” value is not particularly relevant to the main conclusions of this thesis, but future researchers seeking to benefit from our results may wish to bear this in mind.

For referrals, the main parameter examined in this thesis was N_{RE} , the minimum number of experiences that an advisor would need to have with a seller in order to be used in determining the trustworthiness of that seller. However, there may be superior ways of determining the amount of experience between a seller and an advisor. For example, at present the referral procedure does not take into account the length of time that has elapsed since a particular experience – it stands to reason that experiences that happened in the distant past may not count for very much, especially if the seller’s behaviour has changed over time. It may also be useful to consider how factors such as the dynamicity of the community or the amount of knowledge held by the buyer and/or other advisors might affect these selections.

Future research should also certainly consider the maximum number of levels searched when a referral is needed. Presently, this value is set as $maxnetlevel = \lceil \log_k(|B|) \rceil$, as an estimation of the number of levels required in order to search the entire population of advisors. Reducing this number may serve to improve the performance of referrals, albeit potentially at the expense of some useful information from some referred advisors that might no longer be identified. It might be useful therefore to examine how many referred advisors are found at each level, and how much their presence affects the modelling of seller trustworthiness.

²Alternatively we could use a binary search algorithm to identify the best parameter, assuming we could show that the graph would have a global minimum and no local minima (based on our results in Chapter 3, this seems to be the case for PTM, but not necessarily for TRAVOS).

6.2.2 Improvements to Testing Methods

It would be useful to test our techniques on “real” data sets – that is, data originating from real users – and not arbitrary data as performed in the simulations in Chapters 3 and 4. For example, the work on collaborating filtering design choices [9], mentioned earlier as an inspiration for some of our techniques, used the MovieLens database to help determine how useful various methods would be. Potentially more useful for our purposes would be the *epinions.com* dataset, in which users not only provided reviews of items, but also defined a “Web of Trust” consisting of users whose reviews they found to be consistently useful, and a “block list” of reviewers who were consistently not useful [21]. Testing against a dataset of this type could help us to show the robustness of our techniques in a real-world scenario.

An additional beneficial extension to our testing would be to make the advisor networks dynamic, changing over time as agents become more or less trustworthy. Our verification in this thesis simply created a “static” advisor network after a training period, and did not consider the effects of potential subsequent interactions. However, in a real-world use case, we would expect the advisor network, and trust values, to be regenerated or updated from time to time, perhaps every few days, as agents engage in additional transactions, and as new agents enter the environment. We might, for instance, regularly update the trust values associated with existing advisors, and replace the least trustworthy existing advisors with new, more trustworthy agents.

As noted previously, our work in this thesis has examined the effect of our techniques on two trust models: the PTM and TRAVOS. We felt that although there are many similarities between these models, such as the usage of beta distributions, they were sufficiently distinct to show that our techniques, and the results of their use, should not be specific to any specific model. However, it is by no means a comprehensive study, even among probabilistic trust models. In particular we note the recent contribution by Vogiatzis et al. [34] which is claimed to be, unlike the others, a “fully probabilistic” trust model, modelling agent interactions using a Hidden Markov Model, and apparently making no use of beta distributions. We leave consideration of such matters to future research.

6.2.3 Further Verification of Effects on TRAVOS

In Sections 3.3.2, 3.3.3, and 4.3.3, we looked at the effects of the application of our techniques on the TRAVOS trust model. However, our results did not seem to be as clear-cut as those for the PTM. In particular, we noted that the figures showing the initial results for TRAVOS for *max_nbors* and thresholding, Figures 3.4a and 3.4b respectively, displayed graphs with a zig-zag shape, whereas the graphs for PTM (Figures 3.3a and 3.3b) had a relatively smooth shape.

Because of the way advisor trustworthiness is modelled with TRAVOS – taking into account the buyer, the advisor, *and* the seller under consideration, whereas the corresponding computations for the PTM are independent of the seller – it would be valuable to control additional criteria in our simulations that have not been held constant in our research to date, and to then examine any effects on the simulations. These criteria might include the total number of ratings received by each seller, and the overall percentage of unfair ratings from the advisors that interact with each seller. Again, we leave this for future research.

In addition, we could determine more carefully the circumstances under which TRAVOS is challenged in properly determining the trustworthiness of advisors. A useful starting point for this investigation would be the comparison of probabilistic approaches performed in [44].

Finally, we might examine if the proposal in the following subsection of a unified advisor network for TRAVOS might affect our results going forward.

6.2.4 Additional Work with Large Populations

In Section 3.3.3 we examined the effect of increasing the advisor population size when using the PTM with either *max_nbors* or thresholding. We showed that these techniques will have approximately the same effect on the larger population, if the threshold or the proportionate *max_nbors* value is kept constant. However, our results with TRAVOS in that section diverged from this, appearing to show that a given threshold or proportionate *max_nbors* value will yield different results for different advisor populations. Of course,

this may be less a matter of the large populations themselves than one of issues with the TRAVOS model itself, as just discussed in Section 6.2.3, but this is not entirely clear at this stage.

Although we intended to test as well the effects of referrals on the large population scenario in TRAVOS, we found this to be impractical, since increasing the population using TRAVOS seems to have a much more significant effect on the amount of required memory compared to the PTM. This is in part because the modelling of advisor trust in TRAVOS takes into account not only the buyer and advisor, but also the specific seller being considered. This in turn means that, to allow each buyer / advisor to determine its own advisor network, they must in fact create separate networks for each potential seller. This means that in an environment having 100 sellers and making use of advisor networks, TRAVOS may require up to 100 times as much memory as would an otherwise identical scenario in PTM.

This was not a significant issue when thresholding and *max_nbor*s alone are tested, since the advisor networks of the advisors themselves have no effect on the simulations in these cases and thus could be ignored. However, since this component is crucial to the functioning of our referral mechanism, we may need to examine ways to reduce the memory usage in these cases, or alternatively determine a means to collapse the separate advisor networks used for each seller into a single advisor network as used with PTM.

Returning to the results in Section 3.3.3, we noted that notwithstanding the similarity in the general effects of our techniques for both of the advisor population sizes we tested, the larger population tended to perform slightly better – that is, it almost always had lower MAE than the comparable small-population results. We posited that this was simply because more highly-trusted advisors were available in the larger population, yielding more trustworthy information about each of the sellers.

As a suggestion for future work, we would verify this by re-running the simulations, this time keeping track of the trustworthiness values of each advisor, and then sorting the advisors into several trustworthiness bins (e.g. five bins of $[0, 0.2)$, $[0.2, 0.4)$, \dots $[0.8, 1]$). We hypothesize that the higher-population simulations would yield more advisors in the high-trustworthiness bins than those for the smaller advisor populations.

6.2.5 Performance and Time Sensitivity

One topic which would be promising to explore is improving the performance of the trust models when using our proposed methods. Given that these methods will in many cases substantially reduce the size of the advisor network used to produce the trust model of sellers, some performance optimization of these methods could help to improve the overall performance of the trust model.

Zhang also suggested [42] that it may be useful to apply the PTM (and presumably other trust models) to time-sensitive tasks which may require a buyer to make a very quick decision – for example, vehicular ad-hoc networks (VANETs), which might need to consider information from various sources regarding weather, traffic, and road conditions, all of which is changing constantly. Perhaps a more pertinent example in the e-commerce scenario would be a time-limited online auction marketplace, in which bidders may place or increase their bids at any point up until the specified end time. An agent might wish to decide, for instance, how worthwhile placing or increasing a bid would be, given not only the agent’s own worth of the item, but also the updated trustworthiness of the seller and of other bidders (i.e. detecting collusion).

In such a scenario, Zhang argued, an agent might only have time to consult a limited number of advisors. We note, however, with specific regard to the models and modifications studied in this thesis, that the time required of querying advisors is quite small compared to the amount of time required to model the trustworthiness of, and select, advisors.

Our instinct is that a “good enough” model of the trustworthiness of an agent might be sufficient – perhaps one only considering a random subset of the advisor population, as in Section 3.3.4. However, under these circumstances, we would want to have high confidence in our selection, especially if that selection was a particularly risky one (such as, using the VANET example, making recommendations about when to change lanes).

Our review of the literature has not revealed any significant past work on time-sensitive decisions in the context of trust modelling. Although a number of researchers have discussed “time sensitivity” in the context of trust modelling in peer-to-peer networks, including [19][6][2], the references to time sensitivity relate to the use of models incorporating

time-slots to keep track of when events occur, and not time-sensitive decisions.

6.2.6 Combining Collaborative Filtering and Trust Modelling

In Section 5.4, we mentioned that several other researchers have proposed methods of incorporating CF techniques into trust modelling or vice versa. In many cases these involved simply substituting trust as the primary metric in place of similarity [21]. We noted one novel method in the literature used for “trust-based collaborative filtering”, known as k -nearest recommenders (k NR), which dynamically selects the best k neighbours that are able to provide information about a particular desired item [18].

It appears that k NR may be a useful alternative to the thresholding and *max_nbor*s methods we outlined in Chapter 3, and in that regard, future research could include examination of k NR’s performance, in terms of both trust modelling accuracy and computation, compared to the other two methods when applied to the same environment. This would necessitate a more complex experimental environment, with different items (with different corresponding trustworthiness values) sold by each seller.

There is perhaps, however, a larger question relating to potentially considering both similarity and trust when it comes to deciding which agents to interact with. Conceivably we could attempt to combine the measures in some fashion, but this might not be particularly helpful in all cases. Consider for instance an agent that has low trustworthiness but very high similarity to the buyer, or vice versa – would this agent be used as an advisor in this scenario, and if so, will it serve to help or harm the buyer?

A potentially more useful approach might be to use a dual-stage determination. For instance, we could set two thresholds, one representing the minimum trustworthiness for an advisor to have, and another stating the minimum similarity the advisor must have with the buyer; an advisor would then have to pass both thresholds to be accepted into the advisor network. This too would require an experimental environment considering not only generic interactions between buyers and sellers, but also the specific types of items being transferred (and the characteristics thereof). This we leave to future work as well.

6.2.7 Information Gain

Work has also been done on the measure of information gain obtained as more agents are introduced into a trust or reputation system, which may be an additional factor to consider when determining how large the size of the advisor network should be.

For example, Sierra and Debenham [31] suggest the following formula to predict the amount of information $\mathbb{I}_{X_i}^t(\alpha, \beta, \mu)$, with respect to some probability distribution X_i , obtained when a message μ is obtained at time t :

$$\mathbb{I}_{X_i}^t(\alpha, \beta, \mu) = \mathbb{H}^t(X_i) - \mathbb{H}^t(X_{i(\mu)}) \quad (6.1)$$

where $\mathbb{H}(\bullet)$ is Shannon entropy, α is a (requesting) agent (i.e., a buyer), and β is one of the negotiating agents that interacts with α (i.e., a potential seller).

Many of our results in this thesis indicated that, when varying the *max_nbor*s or thresholding parameter slightly in close proximity to the “optimal” values, there was little change to the computed trust modelling error – the error had “bottomed out”. This would presumably indicate that the information gain by adding or removing advisors around these optimal points would be minimal. Thus work along these lines – specifically, determination of parameters for which the instantaneous information gain is minimal – might allow us to compute, in a more deterministic fashion, the appropriate parameters for *max_nbor*s or thresholding.

6.2.8 Usefulness of Referrals

In Chapter 4, we outlined a proposed technique for incorporating an advisor referral system into the trust models being studied; however, our experimental results thus far have had mixed results. In Section 4.3.2, we noted that the use of our referral mechanism yields improvements in trust-modelling accuracy when the advisor network had been limited to a very small number of advisors – as would occur if a small *max_nbor*s value (e.g. 10 or below) was set, or thresholding was applied using a very high threshold (e.g. 0.8 or above).

On the other hand, if the best *max_nbor*s or threshold parameter – as found through experimental determination – had been used instead, the improvements would be significantly lessened. Indeed, using a more carefully-set *max_nbor*s or threshold parameter, even without referrals, will still yield significantly better trust-modelling accuracy compared to a very small advisor network with referrals.

That does not, however, mean that referrals are necessarily without merit, since there may well be practical reasons to reduce the amount of memory that such a program requires.

For example, if a system contains an extremely high agent population – i.e., in the millions – and a limiting method such as random selection (see Section 3.3.4) proves ineffective (for example, if there is high data sparsity), it may well be impractical to maintain a model of all of the agents unless the network is limited to a very small fraction of the overall size, either by using a small *max_nbor*s value (or proportion) or a very high threshold. It is also conceivable that some version of this procedure might eventually be incorporated into embedded systems – i.e., certain types of mobile devices, or devices developed using nanotechnology – in which it tends to be much more difficult to supply a large capacity for either storage or random-access memory compared to larger-scale modern workstations.

It would therefore, we believe, be useful to conduct additional research, as future work, into the potential benefits of using referrals when the available memory is much more limited. This would depend on whether (a) the use of referrals does (or can be made to) reduce memory usage – given that although we assume that a smaller advisor network would be used, the referral process itself requires some memory allocation; and (b) whether the amount of the resulting reductions is in fact significant compared to the overall memory consumption of the trust model.

In a similar vein, we could also consider optimizations to the referral mechanism, as described in pseudo-code in Algorithm 2, to maximize the efficiency of the referrals process and reduce the information load. For example, if certain advisors are highly trusted by several agents, one agent could end up being considered multiple times during the referrals process and rejected each time, due to either the agent having insufficient experience, or the restrictions on duplicate advisors in a referred network. It might be better, for instance,

to (a) have each advisor pre-determine the most useful advisors it already has for each seller, such that when referrals are sought, only the most useful referees for the particular scenario are provided; and (b) ensure that once the advisor network of a particular advisor has been examined, it is not re-examined on some deeper level of the network.

6.2.9 Domain Specificity

Finally, it would be useful to consider how the domain under consideration might affect the choice of both the trust model and the specific methods or parameters used to optimize it. While we have focused on electronic marketplaces in this paper, other models are used in different domains – as in modelling the trust between agents collaborating on a health-related challenge [17] – and the usefulness of our proposed methods may vary from one domain to the next.

Although it would be impractical to model every possible trust-model domain, we can still identify certain combinations of characteristics that could serve to represent many of these domains. For example, motor vehicles tend to have, relative to over consumer goods, high values, but also exist in relatively low quantities, since a typical consumer would only need to buy one every few years. Houses and other real estate would have similar characteristics. In contrast, most food items are only useful for a small number of servings, and thus must be purchased more regularly – but also exist in higher quantities and have lower prices. It therefore stands to reason that the trust model used for buying a car might have very different characteristics from one used for buying household staples.

We expect by identifying the most pertinent characteristics of the relevant domains and adjusting the simulations accordingly, one could make additional conclusions about the role of domain specificity in applying our techniques.

APPENDICES

Appendix A

Relationship Between *max_nbors* and Thresholding

In providing an example scenario applying *max_nbors* and thresholding in Section 3.2.1, we acknowledged that there is a relationship between these two methods, in that applying a particular *max_nbors* value to a given scenario will be equivalent to applying *some* threshold for the same scenario, and vice versa. We now examine whether or not we could use such a relationship on a more general basis.

In conducting the simulations outlined in Section 3.3.1, we also kept track of (a) the size of the generated advisor network, representing the *max_nbors* value that would have generated the same results; and (b) the minimum trust value of the advisors in the network, representing the (maximum) threshold that would have generated the given results. We are primarily concerned with the former with respect to cases where thresholding is applied, and the latter for instances where *max_nbors* was used.

In Figure A.1a, we indicate, on the *y*-axis, the average network size that is produced (i.e. the equivalent *max_nbors* parameter) when different threshold parameters are used (*x*-axis), for three different percentages of lying advisors. As expected, we see that as the applied threshold increases, the equivalent *max_nbors* value decreases. However, the slopes of the graphs (each representing a particular percentage of lying advisors) are not consistent, and in each case the bulk of the decrease happens approximately between the

threshold values of 0.25 and 0.8 (that is, the graphs are fairly flat outside of this range).

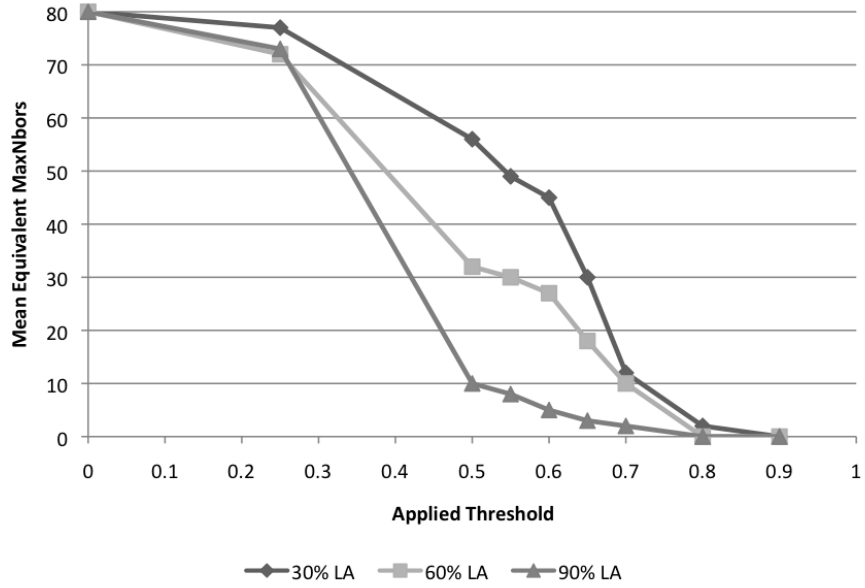
We also see, in comparing the three percentages of lying advisors, that the size of the network will “shrink” more quickly for the 90% lying advisor scenario – that is, the network will drop to a size of 10 for a threshold of 0.5, whereas a threshold of 0.7 is required to get a similar size for the 30% and 60% lying advisor cases.

Finally, that when applying a threshold of 0.55 – which we indicated earlier was the best-performing of the tested thresholds for the 30% and 60% cases – the equivalent *max_nbor_s* values are approximately 49 and 30 respectively. While the latter is consistent with the *max_nbor_s* value that performed best for those two cases (30), the former is significantly different, indicating that we cannot blindly “convert” between *max_nbor_s* and thresholding, and then expect the results to be equivalent.

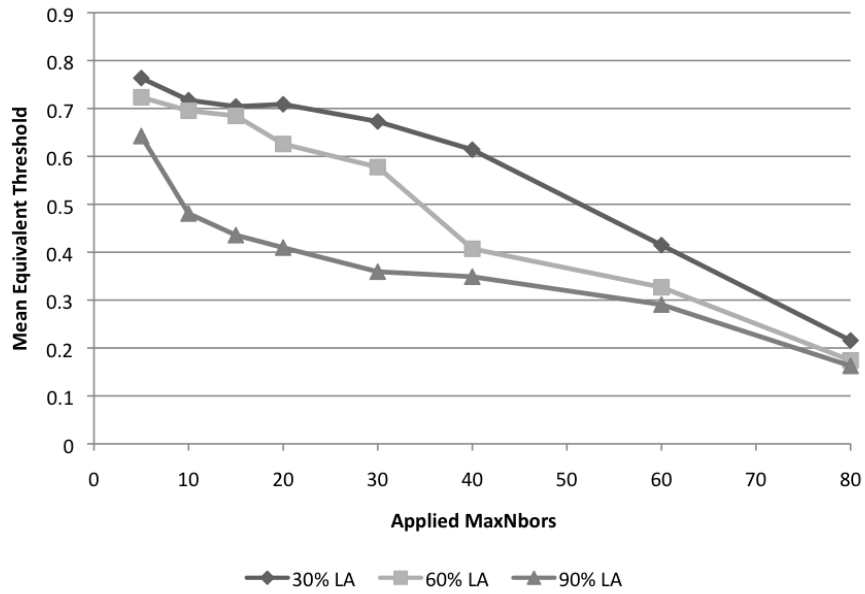
In Figure A.1b, we use the *y*-axis to indicate the average minimum trustworthiness value, i.e. the equivalent threshold, of the advisors accepted into a network determined using *max_nbor_s* (the *max_nbor_s* parameter being indicated by the *x*-axis). Once again, as expected, as the *max_nbor_s* value is increased, the equivalent threshold of the resulting network decrease. But we also see once again that the slope is not consistent for any single graph (i.e. a single percentage of lying advisors), and that the slopes of these graphs differ from each other as well.

Moreover, in observing the results for a *max_nbor_s* value of 30, the “best case” for the 30% and 60% lying advisors cases, we once again see that the resulting equivalent thresholds are not consistent with the “best case” threshold found for those scenarios, 0.55.

These results suggest to us that *max_nbor_s* and thresholding should not be considered interchangeable, particularly if the percentage of dishonest advisors is subject to change.



(a) Equivalent *max_nbars* values for various thresholding scenarios



(b) Equivalent threshold values for various cases where *max_nbars* was used

Figure A.1: Mean equivalent values for the two proposed network-limiting methods

References

- [1] Paolo Avesani, Paolo Massa, and Roberto Tiella. A trust-enhanced recommender system application: Moleskiing. In *2005 ACM Symposium on Applied Computing*, 2005. 95
- [2] Junsheng Chang, Huaimin Wang, and Gang Yi. A time-frame based trust model for P2P systems. In *ICISC 2006*, 2006. 104
- [3] Malcolm Gladwell. *The Tipping Point: How Little Things Can Make a Big Difference*. Little, Brown and Company, New York, 2000. 99
- [4] Jennifer Golbeck and James Hendler. Filmtrust: Movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE Consumer Communications and Networking Conference*, 2006. 96
- [5] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, December 1992. 16
- [6] Leitao Guo, Shoubao Yang, Jing Wang, and Jinyang Zho. Trust model based on similarity measure of vectors in P2P network. In *Fourth International Conference on Grid and Cooperative Computing (GCC 2005)*, 2005. 104
- [7] Chung-Wei Hang and Munindar P. Singh. Trust-based recommendation based on graph similarity. In *Proceedings of the Thirteenth International Workshop on Trust in Agent Societies*, 2010. 84, 94

- [8] Chung-Wei Hang, Yonghong Wang, and Munindar P. Singh. Operators for propagating trust and their evaluation in social networks. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1025–1032, Columbia, SC, 2009. IFAAMAS. 80, 83
- [9] Jon Herlocker, Joseph A. Konstan, and John Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4):1386–4564, October 2002. 4, 5, 16, 17, 21, 74, 77, 79, 98, 101
- [10] Michael N. Huhns and Larry M. Stephens. Multiagent systems and societies of agents. In Gerhard Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 2000. 1
- [11] Chein-Shung Hwang and Yu-Pin Chen. Using trust in collaborative filtering recommendation. In *IEA/AIE 2007*, 2007. 96
- [12] Audun Jøsang. A subjective metric of authentication. In *5th European Symposium on Research in Computer Security (ESORICS)*, 1998. 82, 83
- [13] Audun Jøsang and Roslan Ismail. The beta reputation system. In *15th Bled Electronic Commerce Conference*, 2002. 3, 45, 66
- [14] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007. 9
- [15] Audun Jøsang and Simon Pope. Semantic constraints for trust transitivity. In *Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling*, 2005. 80
- [16] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *WWW2003*, 2003. 84
- [17] Andrew Koster, Jordi Sabater-Mir, and Marco Schorlemmer. Engineering trust alignment: a first approach. In *Proceedings of the Thirteenth International Workshop on Trust in Agent Societies*, 2010. 108

- [18] Neal Lathia, Stephen Hailes, and Licia Capra. Trust-based collaborative filtering. In *Joint iTrust and PST Conferences on Privacy, Trust Management and Security*, 2008. 17, 95, 96, 98, 105
- [19] Lintao Liu, Shu Zhang, Kyung Dong Ryu, and Partha Dasgupt. R-Chain: A self-maintained reputation management system in P2P networks. In *17th International Conference on Parallel and Distributed Computing Systems (PDCS-2004)*, 2004. 104
- [20] Stephen Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Mathematics and Computer Science, University of Stirling, 1994. 2
- [21] Paolo Massa and Paolo Avesani. Trust-aware collaborative filtering for recommender systems. In *Proc. of Federated Int. Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE*, 2004. 95, 101, 105
- [22] Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *RecSys'07*, 2007. 95, 96
- [23] B. W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta*, 405(2):442–451, October 1975.
- [24] D. Harrison McKnight and Norman L. Chervany. Trust and distrust definitions: One bite at a time. In Rino Falcone, Munindar Paul Singh, and Yao-Hua Tan, editors, *Trust in cyber-societies: integrating the human and artificial perspectives*. Springer, 2001. 1
- [25] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999. 19, 95
- [26] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 Conference on Computer Supported Collaborative Work*, 1994. 16, 17
- [27] Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, March 1997. 16

- [28] Jordi Sabater, Mario Paolucci, and Rosaria Conte. Repute: Reputation and image among limited autonomous partners. *Journal of Artificial Societies and Social Simulation*, 9(2):3, March 2006. 19
- [29] Aaditeshwar Seth. Understanding participatory media using social networks. Technical Report CS-2007-47, University of Waterloo, 2008. 98, 99
- [30] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of ACM CHI’95 Conference on Human Factors in Computing Systems*, pages 210–217, 1995. 16, 77
- [31] Carles Sierra and John Debenham. Information-based agency. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2007. 106
- [32] W. T. Luke Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, March 2006. 3, 13, 36, 69
- [33] Thomas Tran and Robin Cohen. Improving user satisfaction in agent-based electronic marketplaces by reputation modeling and adjustable product quality. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 828–835, 2004. 2
- [34] George Vogiatzis, Ian MacGillivray, and Maria Chli. A probabilistic model for trust and reputation. In *Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 225–232, May 2010. 101
- [35] Yonghong Wang and Munindar P. Singh. Trust representation and aggregation in a distributed agent system. In *AAAI’06*, 2006. 82, 83
- [36] Michael Wooldridge. Intelligent agents. In Gerhard Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 2000. 1
- [37] Pinar Yolum and Munindar P. Singh. Dynamic communities in referral networks. *Web Intelligence and Agent Systems*, 2003. 18

- [38] Pinar Yolum and Munindar P. Singh. Engineering self-organizing referral networks for trustworthy service selection. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(3):396–407, May 2005. 18, 19
- [39] Bin Yu and Munindar P. Singh. A social mechanism of reputation management in electronic communities. In *Proceedings of Fourth International Workshop on Cooperative Information Agents*, pages 154–165, 2000. 2, 17, 49
- [40] Bin Yu and Munindar P. Singh. Detecting deception in reputation management. In *AAMAS '03*, 2003. xviii, 5, 17, 19, 78, 79, 81
- [41] Bin Yu and Munindar P. Singh. Searching social networks. In *AAMAS '03*, 2003. 80, 81, 82, 84, 85
- [42] Jie Zhang. *Promoting Honesty in E-Marketplaces: Combining Trust Modeling and Incentive Mechanism Design*. PhD thesis, University of Waterloo, 2009. xix, 1, 3, 4, 11, 12, 15, 17, 23, 24, 27, 29, 98, 104
- [43] Jie Zhang and Robin Cohen. Trusting advice from other buyers in e-marketplaces: The problem of unfair ratings. In *Proceedings of the Eighth International Conference on Electronic Commerce (ICEC)*, 2006. 2, 3, 7, 27, 28, 29, 69
- [44] Jie Zhang, Murat Şensoy, and Robin Cohen. A detailed comparison of probabilistic approaches for coping with unfair ratings in trust and reputation systems. In *Sixth Annual Conference on Privacy, Security and Trust*, 2008. 13, 36, 72, 102