



# **Trust Evaluation in Service-Oriented Environments**

by

**Lei Li**

A thesis submitted in fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
in the  
Department of Computing  
Faculty of Science  
Macquarie University

Supervisor: A/Prof. Yan Wang

2011

© Copyright by

Lei Li

2011

---

# Statement of Candidate

---

I certify that the work in this thesis entitled “**Trust Evaluation in Service-Oriented Environments**” has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree to any other university or institution other than Macquarie University.

I also certify that the thesis is an original piece of research and it has been written by me. Any help and assistance that I have received in my research work and the preparation of the thesis itself have been appropriately acknowledged.

In addition, I certify that all information sources and literature used are indicated in the thesis.

---

Lei Li

17 November 2011

*To my parents,  
Guoen Li and Xiulan Zhang,  
who dedicated all their life to me*

---

# Abstract

---

In service-oriented applications, effective and efficient trust evaluation is crucial. It provides valuable information to service clients, enabling them to select trustworthy service providers, consume high quality services and prevent monetary loss.

One aspect of the work presented in this thesis is trust vector based approaches to trust rating aggregations in service-oriented environments. In the single trust vector approach, a trust vector of three values is computed, which can predict the trust trend and represent a set of ratings distributed within a time interval (e.g. a week or a month etc.). With trust vectors, two service providers with similar final trust values can be compared. In the multiple trust vector approach, a two dimensional aggregation is performed, which consists of both vertical and horizontal aggregations of trust ratings. The vertical aggregation calculates the aggregated rating representing the trust level for the services delivered in a small time period. The horizontal aggregation applies our proposed multiple time interval (MTI) greedy and optimal algorithms to determine the minimal number of time intervals, within each of which a trust vector can be calculated to represent all the ratings in that time interval. The proposed algorithms can return a small set of MTI to represent a large set of trust ratings and preserve the trust features well.

The other aspect of the work presented in this thesis is trust-oriented composite service selection and discovery. In service-oriented environments, a service may invoke other services offered by different providers forming composite services. The complex invocations in composite services greatly increase the complexity of trust-oriented service selection and discovery. To evaluate the subjective global trust of a composite service based on subjective probability theory, a set of subjective trust evaluation approaches have been proposed, which can map the non-binary discrete subjec-

tive ratings to trust values, evaluate the global trust value deductively and maintain the subjective property of trust ratings and trust results. In addition, two algorithms have been proposed for trust-oriented service selection and discovery.

The properties of the above proposed algorithms have been studied both analytically and empirically. These studies illustrate that the proposed algorithms are effective and efficient.

---

# Acknowledgments

---

First of all, I would like to express my sincere appreciation to my supervisor A/Prof. Yan Wang for his kindness and patience. He has led me in the correct direction over the past few years not only with his knowledge and experience, but also with thoughtfulness about a young man's personal growth. Literally, without his continuous support and endless guidance, this work would not have been possible. It is my great fortune to have him as my supervisor at Macquarie University.

I also would like to thank Prof. Ee-Peng Lim for his help in various stages of my work. This academic journey would not have been so rewarding without his kindness and wisdom.

In addition, my colleagues have helped me to develop this work. I wish to express my thanks to Guanfeng Liu, Yi Wang, Zheng Chen, Hadi Mashinchi, Joe Zou, Weiliang Zhao, Haibin Zhang and Daisy He for providing a friendly and enjoyable environment during these years.

Many thanks to the staff in the Department of Computing for their administrative help. I would like to thank Sylvian Chow, Fiona Yang, Melina Chan, Donna Hua and Jackie Walsh for their warm support and help.

Most important of all, I would like to thank my family. My parents, Guoen Li and Xiulan Zhang, have always been there for me. Their love, support and encouragement have been the foundation for my life. I wish to thank them for all the opportunities they have made available to me, and for the support they have given me during my life. Thanks to my de facto wife, Jun Zhang, for her love, understanding and support. Without their love, unwavering support and inspiration, this work could never have been accomplished.



---

# Publications

---

This thesis is based on research I have performed with the help of my supervisor and other colleagues during my PhD program at the Department of Computing, Macquarie University between 2008 and 2011. Some parts of my research have been published in the following papers:

- [1] Lei Li and Yan Wang, The Study of Trust Vector Based Trust Rating Aggregation in Service-Oriented Environments, *World Wide Web Journal (WWWJ)* (ERA rank A journal<sup>1</sup>) (Impact Factor 1.0), accepted on 03/10/2011.
- [2] Yan Wang and Lei Li, Trust Rating Aggregations in Service-Oriented Applications, *IEEE Transactions on Services Computing (TSC)* (regarded as the best journal in the field of services computing), 4(4), pages 257-271, 2011.
- [3] Lei Li and Yan Wang, A Subjective Probability Based Deductive Approach to Global Trust Evaluation in Composite Services, *IEEE International Conference on Web Services (ICWS 2011)* (ERA rank A conference), pages 604-611, 4-9 July, 2011, Washington DC, USA.
- [4] Hadi Mashinchi, Lei Li, Mehmet Orgun and Yan Wang, The Prediction of Trust Rating Based on the Quality of Services Using Fuzzy Linear Regression, *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)* (ERA rank A conference), pages 1953-1959, 27-30 June, 2011, Taipei, Taiwan.
- [5] Lei Li, Yan Wang and Ee-Peng Lim, Trust-Oriented Composite Service Selection with QoS Constraints, *Journal of Universal Computer Science (J.UCS)*, 16(13), pages 1720-1744, 2010.

---

<sup>1</sup>refer to [http://www.arc.gov.au/era/era\\_2010/archive/era\\_journal\\_list.htm](http://www.arc.gov.au/era/era_2010/archive/era_journal_list.htm)

- [6] Lei Li and Yan Wang, Context Based Trust Normalization in Service-Oriented Environments, International Conference on Autonomic and Trusted Computing (ATC 2010), Springer, LNCS Vol. 6407, pages 122-138, 26-29 October, 2010, Xi'an, China.
- [7] Lei Li and Yan Wang, Subjective Trust Inference in Composite Services, Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010) (ERA rank A conference), pages 1377-1384, 11-15 July, 2010, Atlanta, Georgia, USA.
- [8] Lei Li, Yan Wang and Ee-Peng Lim, Trust-Oriented Composite Service Selection and Discovery, International Conference on Service Oriented Computing and European ServiceWave Conference series (ICSOC/ServiceWave 2009) (ERA rank A conference), Springer, LNCS Vol. 5900, pages 50-67, 24-27 November, 2009, Stockholm, Sweden.
- [9] Lei Li and Yan Wang, Trust Evaluation in Composite Services Selection and Discovery, IEEE International Conference on Services Computing (SCC 2009) (ERA rank A conference), pages 482-485, 21-25 September, 2009, Bangalore, India.
- [10] Lei Li, Yan Wang and Vijay Varadharajan, Fuzzy Regression Based Trust Prediction in Service-Oriented Applications, International Conference on Autonomic and Trusted Computing (ATC-09), Springer, LNCS Vol. 5586, pages 221-235, 7-9 July, 2009, Brisbane, Australia.
- [11] Lei Li and Yan Wang, A Trust Vector Approach to Service-Oriented Applications, IEEE International Conference on Web Services (ICWS 2008) (ERA rank A conference), pages 270-277, 23-26 September, 2008, Beijing, China.
- [12] Yan Wang, Lei Li and Ee-Peng Lim, Price Trust Evaluation in E-Service Oriented Applications, IEEE Conference on E-Commerce Technology and IEEE Conference on Enterprise Computing, E-Commerce and E-Services (CEC/EEE 2008), pages 165-172, 21-24 July, 2008, Crystal City, Washington DC, USA.

---

# Contents

---

<b>Abstract</b>	<b>v</b>
<b>Acknowledgments</b>	<b>vii</b>
<b>Publications</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 A Brief Review of Service-Oriented Computing . . . . .	2
1.2 A Brief Preview of Trust . . . . .	3
1.3 Contributions of the Work . . . . .	4
1.4 Roadmap of the Thesis . . . . .	8
<b>2 Literature Review on Trust</b>	<b>11</b>
2.1 General Structure of Trust . . . . .	12
2.1.1 Basic Trust . . . . .	13
2.1.2 A Priori Generalized Trust . . . . .	14
2.1.3 Context-Specific Trust . . . . .	14
2.1.4 Inner Dialogicality . . . . .	15
2.2 Bases of Trust . . . . .	15
2.2.1 Dispositional Trust . . . . .	15
2.2.2 History-based Trust . . . . .	16
2.2.3 Third Parties as Conduits of Trust . . . . .	16
2.2.4 Category-based Trust . . . . .	17
2.2.5 Role-based Trust . . . . .	18
2.2.6 Rule-based Trust . . . . .	18
2.3 Concept of Trust in Multiple Disciplines . . . . .	19

2.4	Trust Evaluation in Online Applications . . . . .	20
2.4.1	Trust Evaluation in E-Commerce Environments . . . . .	21
2.4.2	Trust Evaluation in P2P Information Sharing Networks . . . . .	21
2.4.3	Trust Evaluation in Service-Oriented Environments . . . . .	22
2.4.4	Trust Evaluation in Multi-Agent Systems . . . . .	23
2.4.5	Trust Evaluation in Social Networks . . . . .	24
2.5	Trust Evaluation Taxonomy . . . . .	25
2.5.1	Trust Evaluation Technique Based Taxonomy . . . . .	25
2.5.2	Trust Structure Based Taxonomy . . . . .	26
2.5.3	Trust Bases Based Taxonomy . . . . .	26
2.6	Conclusions . . . . .	27
<b>3</b>	<b>A Trust Vector Approach in Service-Oriented Applications</b>	<b>29</b>
3.1	Service Trust Vector and Its Evaluation . . . . .	30
3.1.1	Final Trust Level ( <i>FTL</i> ) Evaluation . . . . .	31
3.1.2	Service Trust Trend ( <i>STT</i> ) Evaluation . . . . .	31
3.1.3	Service Performance Consistency Level ( <i>SPCL</i> ) Evaluation . . . . .	34
3.1.4	Service Trust Vector . . . . .	36
3.1.5	Experiments on Trust Vector . . . . .	38
3.2	Fuzzy Regression Based Trust Prediction . . . . .	46
3.2.1	Background on Fuzzy Regression . . . . .	46
3.2.2	Fuzzy Regression Model Parameters . . . . .	49
3.2.3	Fuzzy Regression Methodology . . . . .	51
3.2.4	Fuzzy Regression Based Service Trust Vector . . . . .	53
3.2.5	Experiments on Fuzzy Regression Based Trust Prediction . . . . .	55
3.3	Conclusions . . . . .	60
<b>4</b>	<b>Two Dimensional Trust Rating Aggregations</b>	<b>61</b>
4.1	Vertical Trust Rating Aggregation . . . . .	64
4.1.1	Vertical Aggregation without Service Rating Reputation . . . . .	64

4.1.2	Vertical Aggregation with Service Rating Reputation (SRR) . . . . .	67
4.1.3	Experiment on Vertical Rating Aggregation . . . . .	71
4.2	Multiple Time Intervals (MTI) Analysis . . . . .	74
4.2.1	Boundaries of MTI . . . . .	75
4.2.2	Boundary Included Greedy MTI Algorithm . . . . .	77
4.2.3	Bisection-based Boundary Excluded Greedy MTI Algorithm . . . . .	79
4.2.4	Boundary Included Optimal MTI Algorithm . . . . .	83
4.2.5	Boundary Excluded Optimal MTI Algorithm . . . . .	85
4.2.6	Boundary Mixed Optimal MTI Algorithm . . . . .	88
4.2.7	Experiments on MTI Analysis . . . . .	91
4.3	Conclusions . . . . .	111
<b>5</b>	<b>Trust-Oriented Composite Service Selection</b>	<b>113</b>
5.1	Related Work on Trust-Oriented Composite Service Selection . . . . .	115
5.2	Service Invocation Model . . . . .	117
5.2.1	Composite Services and Invocation Relation . . . . .	117
5.2.2	An Example: Travel Plan . . . . .	119
5.2.3	Service Invocation Graph . . . . .	121
5.2.4	Service Invocation Matrix . . . . .	122
5.3	Trust Evaluation in Composite Services . . . . .	123
5.3.1	Trust Estimation Model . . . . .	124
5.3.2	Global Trust Computation in Composite Services . . . . .	126
5.4	Trust-Oriented Composite Service Selection . . . . .	128
5.4.1	Longest <i>SEF</i> Algorithm . . . . .	128
5.4.2	Monte Carlo Method Based Algorithm (MCBA) . . . . .	130
5.4.3	Experiments on Trust-Oriented Composite Service Selection . . . . .	133
5.5	Trust-Oriented Composite Service Selection with QoS Constraints . . . . .	139
5.5.1	Related Work on Composite Service Selection with QoS Constraints . . . . .	140

---

5.5.2	QoS Constrained Monte Carlo Method Based Algorithm . . .	141
5.5.3	Experiment on Trust-Oriented Composite Service Selection with QoS Constraints . . . . .	143
5.6	Conclusions . . . . .	147
<b>6</b>	<b>Subjective Trust Evaluation in Composite Services</b>	<b>149</b>
6.1	Trust Normalization in Service-Oriented Environments . . . . .	151
6.1.1	Comprehensive Evaluation Index System . . . . .	153
6.1.2	Fuzzy Comprehensive Evaluation Model . . . . .	154
6.1.3	Experiments on Trust Normalization . . . . .	160
6.2	Subjective Global Trust Evaluation in Composite Services . . . . .	168
6.2.1	Trust Estimation of Service Components . . . . .	169
6.2.2	Probability Interpretation of Trust Dependency . . . . .	174
6.2.3	Joint Subjective Probability Method . . . . .	178
6.2.4	Subjective Probability Based Deductive (SELECTIVE) Ap- proach . . . . .	179
6.2.5	Experiments and Analysis on Subjective Global Trust Evaluation	184
6.3	Conclusions . . . . .	193
<b>7</b>	<b>Conclusions and Future Work</b>	<b>195</b>
	<b>Appendix A Notations Used in This Thesis</b>	<b>199</b>
	<b>Appendix B Technical Report about Subjective Global Trust Evaluation</b>	<b>207</b>
	<b>Bibliography</b>	<b>217</b>

---

# List of Figures

---

2.1	General Structure of Trust . . . . .	12
3.1	Several <i>STT</i> cases . . . . .	32
3.2	Weighted least square linear regression . . . . .	32
3.3	Several <i>SPCL</i> cases . . . . .	35
3.4	Experiment 1 on Trust Vector . . . . .	40
3.5	Experiment 2 on Trust Vector . . . . .	41
3.6	Experiment 3 on Trust Vector . . . . .	44
3.7	Experiment 4 on Trust Vector . . . . .	46
3.8	Experiment 1 on Fuzzy Regression Based Trust Prediction . . . . .	55
3.9	Feedback of a learner's experience of the same teacher in a course between 2005 and 2008 from the Centre for Professional Development at Macquarie University . . . . .	57
4.1	$R_{SRR_j}^{(t_i)}$ function . . . . .	69
4.2	Service rating values in Experiment on Vertical Rating Aggregation . . . . .	72
4.3	Rating frequency at $t_{10}$ in Experiment on Vertical Rating Aggregation . . . . .	72
4.4	Error bars and <i>STT</i> in Experiment on Vertical Rating Aggregation . . . . .	73
4.5	Several MTI examples . . . . .	75
4.6	MTI examples with different boundaries . . . . .	76
4.7	The properties of <i>SPCL</i> function $T_{SPCL}^{[t_1, t]}$ . . . . .	79
4.8	Rating frequency examples in Experiment 1 of Section 4.2.7.1 . . . . .	92
4.9	Single trust vector in Experiment 1 of Section 4.2.7.1 . . . . .	93
4.10	MTI in Experiment 1 of Section 4.2.7.1 . . . . .	94
4.11	MTI in Experiment 2 of Section 4.2.7.2 . . . . .	95

---

4.12	Comparison of five MTI algorithms in Experiment 2 of Section 4.2.7.2	96
4.13	Case 1 in Experiment 3 of Section 4.2.7.3 with the boundary mixed optimal MTI algorithm . . . . .	98
4.14	Case 2 in Experiment 3 of Section 4.2.7.3 with the boundary mixed optimal MTI algorithm . . . . .	98
4.15	Case 3 in Experiment 3 of Section 4.2.7.3 with the boundary mixed optimal MTI algorithm . . . . .	99
5.1	Atomic invocations . . . . .	118
5.2	Complex invocations examples . . . . .	118
5.3	The <i>SIG</i> for the travel plan of Smith . . . . .	120
5.4	A service execution flow . . . . .	120
5.5	Histogram of $O_T$ for each <i>SEF</i> . . . . .	134
5.6	$O_T$ in the travel plan example . . . . .	135
5.7	CPU time of simulation times . . . . .	136
5.8	$O_T$ in composite service of 100 vertices . . . . .	138
5.9	Histogram of $O_{T_{QoS}}$ for each <i>SEF</i> in the travel plan example . . . . .	145
5.10	CPU time of <i>QC_MCBA</i> in the travel plan example . . . . .	145
5.11	$O_{T_{QoS}}$ in the travel plan composite service . . . . .	146
5.12	$O_{T_{QoS}}$ in the composite service of 52 vertices . . . . .	146
6.1	Triangular membership functions with five levels . . . . .	156
6.2	Ratings from Epinions in Trust Normalization Experiment . . . . .	160
6.3	Experimental results about Type 1 risk . . . . .	163
6.4	Experimental results about Type 2 risk . . . . .	165
6.5	Experimental results about Type 3 risk . . . . .	167
6.6	The service invocation graph ( <i>SIG</i> ) for the travel plan . . . . .	173
6.7	A service execution flow ( <i>SEF</i> ) in the <i>SIG</i> . . . . .	173
6.8	A simple <i>SEF</i> example . . . . .	180
6.9	Certainty with fixed ratio of $x_4$ and $x_5$ . . . . .	185

---

6.10	Certainty with fixed $x_4 + x_5$ . . . . .	185
6.11	Certainty with $x_4$ and $x_5$ when $x_1 = x_2 = x_3 = 0$ . . . . .	187
6.12	Certainty with $x_4$ and $x_5$ when $x_1 = x_2 = 0$ and $x_3 = 10$ . . . . .	188
6.13	Service execution flow $SEF_2$ . . . . .	188



---

# List of Tables

---

3.1	Trust vectors in Experiment 1 on Trust Vector . . . . .	39
3.2	Experiment 2 on Trust Vector results . . . . .	43
3.3	Experiment 3 on Trust Vector results with $w_{t_i}$ . . . . .	44
3.4	Experiment 3 on Trust Vector results without $w_{t_i}$ . . . . .	44
3.5	Experiment 4 on Trust Vector results . . . . .	47
3.6	Experiment 1 on Fuzzy Regression Based Trust Prediction results . .	56
3.7	Feedback of a learner's experience of a teacher for a course taken between 2005 and 2008 from Centre for Professional Development in Macquarie University . . . . .	58
3.8	Goodness-of-fit results in Experiment 2 on Fuzzy Regression Based Trust Prediction . . . . .	59
4.1	Trust vectors in Experiment on Vertical Rating Aggregation . . . . .	74
4.2	Complexity of MTI algorithms proposed in Section 4.2 . . . . .	89
4.3	MTI algorithms compared in Experiments of Section 4.2.7 . . . . .	95
4.4	Consumed CPU time in seconds and the number of returned MTI in Experiment 3 of Section 4.2.7.3 . . . . .	102
4.5	MTI goodness-of-fit for the data sets used in Experiments 2 & 3 of Section 4.2.7.2 & 4.2.7.3 . . . . .	110
5.1	Non-zeros of <i>SIM</i> in Travel Plan . . . . .	124
5.2	Ratings and subjective belief of each service component in the travel plan . . . . .	133
5.3	Weights of service components in Pa . . . . .	133
5.4	Structure of complex composite services . . . . .	137

---

5.5	CPU time in seconds of different examples . . . . .	137
5.6	QoS attribute values of each service component in the travel plan example . . . . .	144
5.7	CPU time in seconds of different examples with QoS constraints . . . . .	147
6.1	Ratings for service components in the <i>SEF</i> in Fig. 6.7 . . . . .	174
6.2	Trust estimation of service components in the <i>SEF</i> in Fig. 6.7 . . . . .	175
6.3	Trustworthiness of each rating in Table 6.1 . . . . .	175
6.4	Trust dependency in the <i>SEF</i> in Fig. 6.7 . . . . .	178
6.5	Trust results in Experiment 1 . . . . .	190
6.6	Trust results in Experiment 2 . . . . .	191
6.7	Trust results in Experiment 3 . . . . .	192
A.1	Notations Used in Chapter 3 . . . . .	199
A.2	Notations Used in Chapter 3 (continued) . . . . .	200
A.3	Notations Used in Chapter 4 . . . . .	200
A.4	Notations Used in Chapter 4 (continued) . . . . .	201
A.5	Notations Used in Chapter 4 (continued) . . . . .	202
A.6	Notations Used in Chapter 5 . . . . .	202
A.7	Notations Used in Chapter 5 (continued) . . . . .	203
A.8	Notations Used in Chapter 6 . . . . .	204
A.9	Notations Used in Chapter 6 (continued) . . . . .	205

# Chapter 1

---

## Introduction

---

In our daily life, there are many occasions when we have to trust others to behave as they promised or as we expect them to do. For example, we trust a bus driver can take us to our destination on time; we trust a doctor to conduct a physical examination and check whether we have an illness; we trust a motor mechanic to find out whether there is a problem in our car and then repair it; we trust a bank and deposit our money. Each time we trust, we have to put something at risk: our lives, our assets, our properties, and so on. On these occasions, we may use a variety of clues and past experiences to believe these individuals' good intentions towards us and decide on the extent to which we can trust them. This is the general procedure of trust evaluation in daily occasions.

Nowadays, with the development of e-commerce application technologies, from time to time it is necessary to buy some products or services from online e-commerce websites. In both e-commerce and e-service environments, when a client looks for one service from a large pool of services or service providers, in addition to service functionality the trustworthiness of a service or service provider is a key factor in service selection. This makes trust evaluation a very important issue in both e-commerce and e-service environments, especially when the client has to select from unknown sellers or service providers.

In general, in a trust management enabled system, service clients can provide feedback and trust ratings after completed transactions. Based on the ratings, the trust value of a service provider can be evaluated to reflect the quality of services in a certain time period. This trust evaluation approach in service-oriented environments is the focus of

our work in this thesis.

## 1.1 A Brief Review of Service-Oriented Computing

In recent years, with the development of information technologies and distributed systems, Service-Oriented Computing (SOC) has emerged as an increasingly important research area, attracting much attention from both the research and industry communities [57].

Conceptually, SOC is a computing paradigm that utilizes services as basic constructs to support the development of rapid and low-cost composition of distributed applications, even in heterogeneous environments [54, 72]. In service-oriented applications, a variety of services across domains are provided to clients in a loosely-coupled environment. Clients can look for preferred and qualified services via the discovery of service registries, and invoke and receive services from the rich service environments [72]. For example, in a company several departments may develop a variety of services in different implementation languages; with a service-oriented architecture, a department's clients can access and use the services from other departments through a well defined interface.

Conceptually, in SOC environments a service is an autonomous, platform-independent computational entity, which can be described, published, discovered and dynamically assembled for developing massively distributed systems [72]. For example, in SOC environments, a service can refer to a transaction, such as selling a product online (i.e. the traditional online service), or a functional component implemented by Web service technologies [57]. However, when a client looks for a service from a large set of services offered by different providers, in addition to functionality, reputation-based trust is also a key factor for service selection. It is also a critical task for service registries to be responsible for maintaining the list of reputable and trustworthy services and service providers, and bringing them to clients [85].

---

## 1.2 A Brief Preview of Trust

Conceptually, trust is the measure taken by one party of the willingness and ability of another party to act in the interest of the former party in a certain situation [45]. If the trust value is in the range of  $[0,1]$ , it can be taken as the subjective probability by which one party expects that another party can perform a given action [41].

The issue of trust has also been actively studied in Peer-to-Peer (P2P) networks (e.g. [19, 44, 95]), which can be used for information-sharing systems [3]. In a P2P system, it is quite natural for a client peer to doubt if a serving peer can provide the complete file prior to any download action, which may be quite time-consuming and network bandwidth-consuming. Unlike some trust management systems in e-commerce or service-oriented environments, in the P2P trust management system a requesting peer needs to inquire the trust data of a serving peer (target peer) from other peers who may have transacted with the serving peer [44, 65, 95]. The computation of the trust level of the serving peer from the collected trust ratings is then performed by the requesting peer rather than a central management server, because of the decentralized architecture of the P2P system.

Unlike P2P information-sharing networks or the eBay reputation management system where a binary rating system is used [95], in SOC environments a trust rating is usually a value in the range of  $[0,1]$  given by a service client [85, 88, 90], representing the subjective belief of the service client on their satisfaction with a service or a service provider. The trust value of a service or a service provider can be calculated by a trust management authority based on the collected trust ratings representing the reputation of the service or the service provider.

Effective and efficient trust evaluation is highly desirable and critical for service clients to identify potential risks, providing objective trust results and preventing huge monetary loss [87].

### 1.3 Contributions of the Work

This thesis contributes in two main ways to the study of trust evaluation in service-oriented environments.

1. One aspect of the work presented in this thesis is trust vector based approaches to trust rating aggregations in service-oriented environments.

- (a) In most existing trust evaluation models [19, 44, 82, 85, 88, 90, 95, 97, 102], a single trust value (e.g. a value in the range of  $[0,1]$ ) is computed to reflect the global trust level of a target accumulated over a certain time period (e.g. in the last 6 months). It is easy for single trust value based systems to be used in trust-oriented service comparison and selection. However, a single trust value of a service provider computed by a service management authority cannot depict the real trust level very well under certain circumstances. For example, if there are two service providers  $A$  and  $B$  with their final trust values  $T_A \approx 0.7$  and  $T_B \approx 0.7$  (each of which is in the range of  $[0,1]$ ), does this mean that both  $A$  and  $B$  have the same trust level? It is not true if  $A$ 's trust values are turning worse with an accumulated value of 0.7 while  $B$ 's trust values are becoming better. In this case,  $B$  is better than  $A$  in terms of predicting the trust level of a forthcoming transaction.

In this thesis, we propose a trust vector to represent a set of ratings distributed within a time interval with three values [51, 59]: *final trust level*, *service trust trend* and *service performance consistency level*. The *final trust level* is represented by a value in  $[0, 1]$ . The *service trust trend* is computed as a numerical value in  $(-\infty, +\infty)$  representing the trend of service trust changes during a given time interval that can be interpreted as *coherent*, *upgoing*, *dropping* or *uncertain*. The *service performance consistency level* is represented by a numerical value in  $[0,1]$  measuring the

---

extent to which the computed service trust trend fits the given set of trust ratings.

With trust vectors, two service providers with similar final trust values can be compared.

- (b) In most existing studies on trust evaluation, a single trust value is aggregated from the ratings given to the previous services of a service provider, to indicate his/her current trust level. Such a mechanism is useful, but may not be able to depict the trust features of a service provider well under certain circumstances. Alternatively, a complete set of trust ratings can be transferred to a service client for local trust evaluation. However, this incurs a big overhead in communication, since the rating data set is usually large-scale, covering a long service history. The third option is to generate a small data set that should represent the large set of trust ratings over a long time period well.

With our proposed single trust vector approach, a trust vector of three values resulting from a computed regression line can represent a set of ratings distributed within a certain time interval (e.g. a week or a month etc.). However, the computed trust vector can represent the set of ratings well only if these ratings imply consistent trust trend changes and are all very close to the obtained regression line.

In a more general case with trust ratings for a long service history, a two dimensional aggregation is performed, which consists of both vertical and horizontal aggregations of trust ratings. The vertical aggregation calculates the aggregated rating representing the trust level for the services delivered in a small time period. The horizontal aggregation applies our proposed multiple time interval (MTI) greedy and optimal algorithms to determine the minimal number of time intervals, within each of which a trust vector can be obtained and can represent all the corresponding ratings well.

Hence, a small set of data can represent a large set of trust ratings well with well preserved trust features. This is significant for large-scale trust rating transmission, trust evaluation and trust management.

In this thesis, we propose five MTI analysis algorithms [87]. We have also studied the properties of our proposed algorithms both analytically and empirically. These studies illustrate that our algorithms can return a small set of MTI to represent a large set of trust ratings and preserve the trust features well.

2. The second aspect of the work presented in this thesis is trust-oriented composite service selection and discovery.

In SOC environments, to satisfy the specified functionality requirement it is usually necessary to effectively compose different kinds of services across domains forming a composite service, which requires that the involved service can be trusted by service clients and other collaborating services [41]. Given a set of various services, different compositions may lead to different service invocation structures. Although these compositions certainly enrich service provision, they greatly increase the complexity of subjective trust evaluation and thus make a proper subjective global trust evaluation very challenging.

In the literature, there are some existing studies for service composition and quality driven service selection [29, 69, 94, 101, 103]. However, for trust-oriented composite service selection and discovery, some research problems remain open.

- (a) Trust is context dependent, i.e. for different contexts of transactions (e.g. transaction cost, product/service category, clients), there are different factors influencing the trust result [88, 89].

In this thesis, we propose a method for building up a projection from the trust ratings in the transaction history of a service provider to an upcoming

---

transaction depending on the contexts of the previous transactions and the upcoming one. This process is termed *context based trust normalization* [53]. After trust normalization, normalized trust ratings are used for trust evaluation, the results of which would be closely bound to the upcoming transaction.

- (b) The definition of a proper graph representation of composite services including both probabilistic invocations and parallel invocations is still lacking. The corresponding data structure is also essential. It is fundamental and important to define these representations to support the global trust evaluation of composite services.

In this thesis, we present the service invocation graph and service invocation matrix for composite service representation [57, 58].

- (c) From the definitions in [41, 45], trust can be taken as the subjective probability, i.e. the degree of belief an individual has in the truth of a proposition [30, 33], rather than the objective probability or classical probability, which is the occurrence frequency of an event [33]. A subjective probability is derived from an individual's personal judgment about a specific outcome (e.g. the evaluation of teaching quality or service quality). It differs from person to person. Hence, classical probability theory is not a good fit for trust evaluation. Instead, subjective probability theory [30, 33] should be adopted.

In this thesis, a Bayesian inference based subjective trust estimation method for service components is proposed [54, 57, 58].

- (d) Although a variety of trust evaluation methods exist in different research areas [45, 85, 95, 102], they either ignore the subjective probability property of trust ratings, or neglect complex invocation structures. As a result, no proper mechanism exists yet for the subjective global trust evaluation of composite services.

In this thesis, we interpret the trust dependency caused by service invocations as conditional probability, which can be evaluated based on the trust values of service components [54, 55]. Then, we propose a joint subjective probability approach and a subjective probability based deductive approach to evaluate the subjective global trust of a composite service on the basis of trust dependency [54, 55].

- (e) Taking trust evaluation and the complex structure of composite services into account, effective algorithms are needed for trust-oriented composite service selection and discovery. These are expected to be more efficient than the existing approaches [69, 101].

In this thesis, based on Monte Carlo method we propose a service selection and discovery algorithm and a QoS constrained service selection algorithm [57, 58]. Experiments have been conducted on composite services of various sizes to compare the proposed algorithms with the existing exhaustive search method [69]. The results illustrate that our proposed algorithms are effective and more efficient.

## 1.4 Roadmap of the Thesis

This thesis paves some ways to trust evaluation in service-oriented environments, and the structure of this thesis is as follows.

Chapter 2 presents a comprehensive literature review of both the theoretical analysis of trust and trust evaluation methods in online application fields.

One aspect of the work presented in this thesis is trust vector based approaches to trust rating aggregations in service-oriented environments. This is discussed in the following two chapters. Chapter 3 proposes our single trust vector approach. This chapter includes our papers [51, 59] published in ICWS 2008 and ATC 2009<sup>1</sup>. In Chapter 4, with our proposed vertical aggregation approaches and horizontal aggrega-

---

<sup>1</sup>For details of the conferences, please refer to Publications on page ix.

---

gation approaches, a small set of values can represent a large set of trust ratings well with well preserved trust features. This chapter includes our papers [87, 56] published in TSC and WWWJ.

The other aspect of the work presented in this thesis concerns trust-oriented composite service selection and discovery. In Chapter 5, based on Monte Carlo method, we propose a service selection and discovery algorithm and a QoS constrained service selection algorithm. This chapter includes our papers [52, 57, 58] published in SCC 2009, ICSOC 2009 and J.UCS. In Chapter 6, we propose a joint subjective probability approach and a subjective probability based deductive approach to evaluate the subjective global trust of a composite service on the basis of trust dependency. This chapter includes our papers [54, 55, 53] published in AAI 2010, ICWS 2011 and ATC 2010.

Finally, Chapter 7 briefly concludes this work and points out some directions for future research opportunities.



# Literature Review on Trust

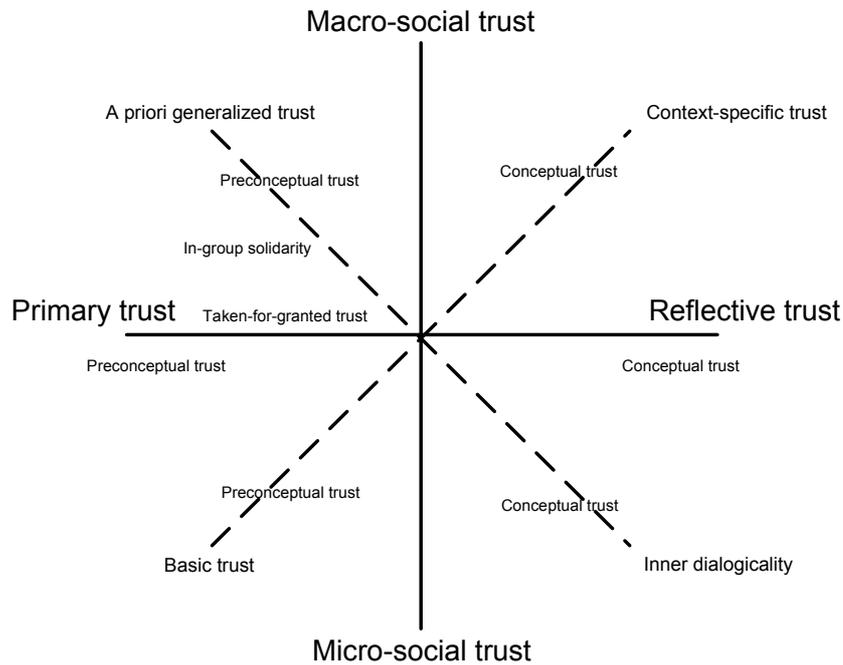
---

In the 1980s and 1990s, a significant number of social science research papers draw attention to trust in many aspects of life [64]. Subsequently, the issue of trust has been discussed not only by social scientists but also by professionals in politics, economics, computer science and so on.

In the literature, trust is a very complicated issue. Related to different networks of concepts on languages, cultures and sociopolitical systems, the term “trust” is highly polysemic. In fact, in different languages there are different semantic distinctions between trust and confidence, reliance, expectation, faith and so on [64].

In this chapter, the literature review on trust is organized as follows:

- Section 2.1 presents a general structure of trust, which provides a general global picture of trust. With this structure, it is easy to start a preliminary theoretical analysis of trust.
- Section 2.2 identifies the bases of trust, with which trust can be established from a variety of diverse sources of trust-related information.
- Section 2.3 briefly introduces the concepts of trust defined in multiple disciplines, including sociology, history, psychology, economics and so on.
- Section 2.4 focuses on trust evaluation models used in different online applications, including e-commerce, P2P networks, service-oriented computing, multi-agent systems and social networks.



**Figure 2.1:** General Structure of Trust

- In Section 2.5, the above mentioned trust evaluation methods can be categorized into different taxonomies with respect to trust evaluation techniques, the structure of trust and the bases of trust respectively.
- Finally, Section 2.6 concludes our work in this chapter.

## 2.1 General Structure of Trust

The general structure of trust has been proposed in [64] and graphically represented in Fig. 2.1. This structure provides a general global picture of trust, with which professionals, scientists and even ordinary citizens can start a preliminary theoretical analysis of trust. With primary trust and reflective trust as the horizontal axis, and micro-social trust and macro-social trust as the vertical axis, this presentation creates four spaces which correspond to four orthogonally placed forms of trust.

- Vertically, passing from the bottom half of Fig. 2.1 toward the top, we move

---

from micro-social trust (i.e. personal, private and interpersonal trust) toward macro-social trust (i.e. professional, group and organizational trust).

- Horizontally, the left-hand side of Fig. 2.1 is characterized by trust as feelings, either based on the interdependence between the self and other, or associated with security or social cohesion [6]. As we move toward the right-hand part of Fig. 2.1, trust becomes conceptualized and rationalized [6]. Trust in the right-hand part of Fig. 2.1 is contractual, and is based on obligations and morality.

In other words, in the left-hand side we focus on primary trust (i.e. immediately apprehended [preconceptual] forms of trust), while in the right-hand side trust is established between strangers or institutions and between organizations and groups of various kinds (i.e. reflective trust [6]). However, once trust has been established, it transforms into common knowledge and becomes taken-for-granted and commonly understood. In contrast to the left-hand side of Fig. 2.1, this taken-for-grantedness arises from reflective thinking. There is also a case whereby, as a result of an individual's doubt trust is brought back into discourse explicitly. When trust is explicitly verbalized, it is no longer taken-for-granted and is partly or fully destroyed. It is necessary to establish trust from the very beginning again.

### **2.1.1 Basic Trust**

Now let us focus on the bottom left quadrant of Fig. 2.1, the boundaries of which are determined by micro-social and primary trust. In the bottom left corner of this quadrant, there is what developmental psychologists describe as *basic trust* between mother and baby.

Basic trust is the first mark of an individual's mental life, even before feelings of autonomy and initiative develop [22]. Through the mutuality between mother and baby, basic trust evolves through mutual somatic experiences and "unmistakable communication" that creates security and continuity. With the presupposition that humans

possess the capacity to make distinctions, the child, equipped with an innate capacity for intersubjectivity, learns through actions, experiences and communications to differentiate between the mental states of others, between feelings, and between trustworthy and untrustworthy relations [79].

### **2.1.2 A Priori Generalized Trust**

Moving to the second quadrant in the top left part of Fig. 2.1, we can see that it is circumscribed by primary trust and macro-social trust. In the top left quadrant of Fig. 2.1, a *a priori generalized trust* is above all a fundamental psychosocial feeling, and it is instantaneously apprehended, quite often without the awareness of those concerned [63]. Generally speaking, the top left quadrant contains trust which is characterized by the kinds of social relations in a society where individuals have certain kinds of social activities. In society, they conceive, create and communicate their social relations and trust with others. Social differentiation leads to the formation of social groups, associations and institutions in which individuals are bound together by impersonal relations. Particularly, in a heterogeneous and complex society like ours, trust is person-specific and content-specific [79]. In our society, during daily life we have to deal with strangers all the time, but here we only deal with one aspect of a stranger and not with the whole person.

In this quadrant, somewhere more towards the intersection, we can place *in-group solidarity*, which can be taken as a form of trust [16]. It includes the social binding and bounding of close in-groups, such as the social cohesion and social ties within family, friends, neighbors, coactivists, and other communities.

### **2.1.3 Context-Specific Trust**

Now we focus on the third quadrant of Fig. 2.1, and it is bounded by macro-social and reflective trust. This quadrant includes trust resulting from a variety of forms, ranging from cooperation to audits, strategies, calculations and so on. The type of trust located

---

in this quadrant is *context-specific trust*, which is derived from contextual information [41]. Context-specific trust is conceptualized and symbolically communicable, and it can be implicitly presented in interactions, relationships and communication.

#### **2.1.4 Inner Dialogicality**

Finally, we arrive at the bottom right quadrant of Fig. 2.1, and in this quadrant we can place *inner dialogicality* [9]. By inner dialogicality, we mean the capacity of humans to carry out internal dialogues (i.e. dialogues within the self). For example, it could include evaluations of one's own and others' past experiences and present conduct, which reflects personal issues and predicts the future conduct. Inner dialogues include not only self-confidence but also self-doubt [64]. With inner dialogues, individuals can develop an awareness of how, where, when and why they can trust or have confidence in specific others (or in themselves).

## **2.2 Bases of Trust**

Research on identifying the bases of trust attempts to establish the conditions which lead to the emergence of trust, including psychological, social, and organizational factors that influence individuals' expectations about others' trustworthiness and their willingness to behave trustworthily during a transaction [6, 49].

### **2.2.1 Dispositional Trust**

Individuals behave differently in their general predisposition to trust different people [49]. To explain the origins of such dispositional trust, Rotter [76] proposed that people tend to build up general trustworthiness about other people from their early trust-related experiences (e.g. the basic trust proposed in Section 2.1.1). In addition, we usually assume that an individual has a relatively stable personality characteristic [76], i.e. a relatively stable dispositional trust.

### **2.2.2 History-based Trust**

In the literature, it has been pointed out that individuals' willingness to engage in trusting others is largely a history dependent process [11]. Interactional histories provide decision makers with useful trust information on the estimation of others' dispositions, intentions and motivations. With the assumption of a relatively stable personality characteristic, this historical information also provides a basis for making predictions about others' future behaviors.

Interactional histories have a significant effect on two psychological facets of trust judgment.

- First, individuals' estimations about others' trustworthiness depend on their prior expectations about others' behaviors.
- Second, these expectations vary with subsequent experience, which either validates or discredits the expectations.

In this regard, history-based trust can be taken as an important basis for establishing knowledge-based or personalized trust [50].

Personalized knowledge can provide important information for trust estimation. However, such knowledge is usually hard to obtain. In most situations, it is impossible for decision makers to accumulate sufficient knowledge about the potential individuals with whom they would like to transact. As a consequence, a variety of substitutes for such direct personalized knowledge have to be utilized [18] and many other bases of trust have to be introduced.

### **2.2.3 Third Parties as Conduits of Trust**

Considering the importance of personalized knowledge regarding others' trustworthiness and its difficulty to obtain, third parties are introduced as conduits of trust because of their diffusion of trust-related information.

In our daily life, simple examples of using third parties as conduits of trust are gossip and word-of-mouth. These ways can provide a valuable source of second-hand knowledge about others [13], but the effects of these ways on trust estimations are complex and do not always have positive effects on the estimation of others' trustworthiness. That is because third parties usually tend to disclose only partial information about others [13]. In particular, when an individual has a strong relation to a prospective trustee, third parties usually prefer to convey the information which they believe the individual wants to hear, i.e. the information which strengthens the tie [49]. This will increase the certainty about the trustee's trustworthiness. Thus, third parties tend to amplify such trust.

Third parties also play an important role in the development and diffusion of trust in social networks [83]. When there is no sufficient knowledge or transactional history available, individuals can turn to third parties for transferring their well-established trust relationships. This provides a base of trust which will be validated or discredited with subsequent experience.

#### **2.2.4 Category-based Trust**

Category-based trust refers to trust estimation based on the information regarding a trustee's membership in a social or organizational category. For example, we can take gender, race or age as a social category. This information usually unknowingly influences others' estimations about the trustee's trustworthiness. Due to the cognitive consequences of categorization and ingroup bias, individuals tend to attribute positive characteristics such as cooperativeness and trustworthiness to other ingroup members [12]. As a result, individuals can establish a kind of depersonalized trust on other ingroup members based only on awareness of their shared category membership.

### **2.2.5 Role-based Trust**

Role-based trust focuses on trust estimation based on the knowledge that a trustee occupies a particular role in an organization rather than that trusters have specific knowledge about the trustee's dispositions, intentions and motivations. To some extent, it is believable that technically competent role performance is usually aligned with corresponding roles in organizations [10]. For example, in the case of vehicle maintenance, we usually trust a motor mechanic to find out whether there is a problem with the car. Therefore, individuals can establish a kind of trust based on the knowledge of role relations, even without personalized knowledge or transactional history.

Role-based trust is established from the fact that there are some prerequisites to occupy a role in an organization, such as the training and socialization processes that role occupants have undergone, and their intentions to ensure their technically competent role performance.

Role-based trust can also be quite vulnerable, especially during organizational crises or when novel situations occur which confuse organizational roles or break down role-based transactions.

### **2.2.6 Rule-based Trust**

Both formal and informal rules capture much of the knowledge about tacit understandings regarding transaction behaviors, interactional routines, and exchange practices [62]. Formal rules are determined by a trust management authority to establish trust between truster and trustee. For example, with the help of PayPal [4], a buyer can trust an unknown seller for a certain transaction. In contrast, informal rules are not explicitly determined by any trust management authority. Instead, they are formed by tradition, religion or routines. For example, in academic environments, early career researchers usually trust senior researchers to help them and guide their research path.

Rule-based trust is estimated not on a conscious calculation of consequences, but rather on shared understandings regarding rules of appropriate behaviors. Regard-

---

ing the effects of rules on individuals' self-perceptions and expectations about other organizational members, rules can create and sustain high levels of trust within an organization [62].

## **2.3 Concept of Trust in Multiple Disciplines**

Complex social phenomena like trust cannot be properly understood from the perspective of a single discipline or in separation from other social phenomena [64]. Although considerable attention to the problem of defining trust has been afforded [49], it is understandable that a single researcher cannot master all the knowledge related to trust in all related disciplines. Thus, a concise and universally accepted definition of trust has remained elusive, and the concept of trust is usually based on analysis from the viewpoint of a single discipline. For example, the basic concept of trust has been widely explored in multiple disciplines, as discussed below.

From the perspective of sociology and history, according to Seligman [78], “trust enters into social interaction in the interstices of systems, when for one reason or another systematically defined role expectations are no longer viable”. If people play their roles according to role expectations, we can safely conduct our own transaction accordingly. The problem of trust emerges only in cases where there is “role negotiability”, i.e. there is “open space” between roles and role expectations [78].

Seligman [78] also points out that trust is a modern phenomenon. What might appear as trust in premodern societies is nothing but “confidence in well-regulated and heavily sanctioned role expectations”. Modernity saw the rise of individualism and the proliferation of societal roles. There was thus a greater degree of negotiability of role expectations and greater possibility for role conflicts, and this resulted in a greater potential for the development of trust in modern society.

From the perspective of sociology, Coleman [15] proposes a four-part definition of trust.

- Placement of trust allows actions that otherwise are not possible, i.e. trust allows

actions to be conducted based on incomplete information on the case in hand.

- If the person in whom trust is placed (trustee) is trustworthy, then the trustor will be better off than if s/he had not trusted. Conversely, if the trustee is not trustworthy, then the trustor will be worse off than if s/he had not trusted.
- Trust is an action that involves a voluntary transfer of resources (e.g. physical, financial, intellectual, or temporal) from the trustor to the trustee with no real commitment from the trustee.
- A time lag exists between the extension of trust and the result of the trusting behavior.

This definition allows for the discussion of trust behavior, which is useful in reasoning about human-computer trust and trust behaviors in social institutions.

From the perspective of psychology, trust is the belief in the person who you trust to do what you expect. Individuals in relationships characterized by high levels of social trust are more apt to exchange information and to act with benevolence toward others than those in relationships lacking trust. Misztal [70] points out three basic things that trust does in the lives of people: It makes social life predictable, creates a sense of community, and makes it easier for people to work together.

From the perspective of economics, trust is often conceptualized as reliability in transactions [64].

In all cases, trust involves many heuristic decision rules, requiring the trust management authority to handle a lot of complex information with great effort in rational reasoning [14].

## **2.4 Trust Evaluation in Online Applications**

The issue of trust has been studied in some online application fields.

### 2.4.1 Trust Evaluation in E-Commerce Environments

Trust is an important issue in e-commerce (EC) environments. At eBay [1], after each transaction, a buyer can give feedback with a rating of “positive”, “neutral” or “negative” to the system according to the service quality of the seller. eBay calculates the feedback score  $S = P - N$ , where  $P$  is the number of positive ratings left by buyers and  $N$  is the number of negative ratings. The positive feedback rate  $R = \frac{P}{P+N}$  (e.g.  $R = 99.1\%$ ) is then calculated and displayed on web pages. This is a simple trust management system providing valuable reputation information to buyers.

In [102], the Sporas system is introduced to evaluate trust for EC applications based on the ratings of transactions in a recent time period. In this method, the ratings of later transactions are given higher weights as they are more important in trust evaluation. The Histos system proposed in [102] is a more personalized reputation system compared to Sporas. Unlike Sporas, the reputation of a user in Histos depends on who makes the query, and how that person rated other users in the online community. In [80], Song et al. apply fuzzy logic to trust evaluation. Their approach divides sellers into multiple classes of reputation ranks (e.g. a 5-star seller, or a 4-star seller). In [88], Wang et al. propose an approach to evaluate situational transaction trust, which binds the trust ratings of a forthcoming transaction with previous transactions. Since the situational trust vector includes service specific trust, service category trust, transaction amount category specific trust and price trust, it can deliver more objective transaction specific trust information to buyers and prevent some typical attacks. In [89], Wang and Lin present some reputation-based trust evaluation mechanisms to more objectively depict the trust level of sellers on forthcoming transactions and the relationship between interacting entities.

### 2.4.2 Trust Evaluation in P2P Information Sharing Networks

The issue of trust has been actively studied in Peer-to-Peer (P2P) information sharing networks as a client peer needs to know prior to download actions which serving peer

can provide complete files. In [19], Damiani et al. propose an approach for evaluating the reputation of peers through a distributed polling algorithm and the *XRep* protocol before initiating any download action. This approach adopts a binary rating system and is based on the Gnutella [3] query broadcasting method. EigenTrust [44] adopts a binary rating system as well, and aims to collect the local trust values of all peers to calculate the global trust value of a given peer. Some other P2P studies also adopted the binary rating system. In [95], Xiong et al. propose a *PeerTrust* model which has two main features. First, they introduce three basic trust parameters (i.e. the feedback that a peer receives from other peers, the total number of transactions that a peer performs, the credibility of the feedback sources) and two adaptive factors in computing the trustworthiness of peers (i.e. transaction context factor and the community context factor). Second, they define some general trust metrics and formulas to aggregate these parameters into a final trust value. In [65], Marti et al. propose a voting reputation system that collects responses from other peers on a target peer. The final reputation value is calculated by aggregating the values returned by responding peers and the requesting peer's experience with the target peer. In [105], Zhou et al. discover a power-law distribution in peer feedbacks, and develop a reputation system with a dynamical selection on a small number of power nodes that are the most reputable in the system.

### **2.4.3 Trust Evaluation in Service-Oriented Environments**

In the literature, the issue of trust also has received much attention in the field of service-oriented computing (SOC). In [85], Vu et al. present a model to evaluate service trust by comparing the advertised service quality and the delivered service quality. If the advertised service quality is as good as the delivered service quality, the service is reputable. In [90], Wang et al. propose some trust evaluation metrics and a formula for trust computation with which a final trust value is computed. In addition, they propose a fuzzy logic based approach for determining reputation ranks that particularly

differentiate the service periods of new and old (long-existing) service providers. The aim is to provide incentives to new service providers and penalize those old service providers with poor service quality. In [61], Malik et al. propose a set of decentralized techniques aiming at evaluating reputation-based trust with the ratings from clients to facilitate the trust-oriented selection and composition of Web services. In [17], Conner et al. present a trust model that allows service clients with different trust requirements to use different weight functions that place emphasis on different transaction attributes. This customized trust evaluation provides flexibility for service clients to have different trust values from the same feedback data.

#### **2.4.4 Trust Evaluation in Multi-Agent Systems**

Trust has also drawn much attention in the field of multi-agent systems. In [40], Jøsang describes a framework for combining and assessing subjective ratings from different sources based on Dempster-Shafer belief theory, which is a generalization of the Bayesian theory of subjective probability. In [82], Teacy et al. introduce the TRAVOS system (Trust and Reputation model for Agent-based Virtual OrganisationS) which calculates an agent's trust on an interaction partner using probability theory, taking into account the past interactions between agents. In [26], Griffiths proposes a multi-dimensional trust model which allows agents to model the trust value of others according to various criteria. In [77], Sabater et al. propose a model discussing trust development between groups. When calculating the trust from individual  $A$  to individual  $B$ , a few factors are considered, e.g. the interaction between  $A$  and  $B$ , the evaluation of  $A$ 's group to  $B$  and  $B$ 's group, and  $A$ 's evaluation to  $B$ 's group. In [20], a community-wide trust evaluation method is proposed where the final trust value is computed by aggregating the ratings (termed as votes in [20]) and other aspects (e.g. the rater's location and connection medium). In addition, this approach computes the trust level of an assertion (e.g. trustworthy or untrustworthy) as the aggregation of multiple fuzzy values representing the trust resulting from human interactions. In [36], in trust evalu-

ation, the motivations of agents and the dependency relationships among them are also taken into account.

### **2.4.5 Trust Evaluation in Social Networks**

In the literature, the issue of trust becomes increasingly important in social networks. In [86], Walter et al. identify that network density, the similarity of preference between agents, and the sparseness of knowledge about the trustworthiness of recommendations are crucial factors for trust-oriented recommendations in social networks. However, the trust-oriented recommendation can be attacked in various ways, such as sybil attack, where the attacker creates a potentially unlimited number of identities to provide feedback and increase trust level. In [99], Yu et al. present SybilGuard, a protocol for limiting the corruptive influences of sybil attacks, which depends on the established trust relationship between users in social networks.

Trust propagation, during which the trust of a target agent can be estimated from the trust of other agents, is an important problem in social networks. In [25], Golbeck et al. present trust propagation algorithms based on binary ratings. In social networks, many more non-binary trust propagation approaches have been proposed. In [28], Guha et al. develop a framework dealing with not only trust propagation but also distrust propagation. In [32], Hang et al. propose an algebraic approach to propagating trust in social networks, including a concatenation operator for the trust aggregation of sequential invocation, an aggregation operator for the trust aggregation of parallel invocation, and a selection operator for trust-oriented multiple path selection. In [84], Victor et al. present a trust propagation model, which takes into account fuzzified trust, fuzzified distrust, unavailable trust information and contradictory trust information simultaneously.

## 2.5 Trust Evaluation Taxonomy

Trust evaluation is based on the trusters' knowledge of trust, which is only in the trusters' minds. This makes the analysis process highly human-dependent and therefore prone to errors. Knowledge of trust can be abstract/general, or domain/application specific, etc. From different viewpoints, the trust evaluation approaches presented in Section 2.4 can be categorized into different taxonomies as follows.

### 2.5.1 Trust Evaluation Technique Based Taxonomy

Similar to the taxonomy in [20], we can categorize the above mentioned trust evaluation approaches presented in Section 2.4 as follows according to their computation techniques. Some approaches may correlate to more than one category.

- *Category 1* adopts the approach of calculating the summation or weighted average of ratings, like the models in [20, 26, 88, 90, 95, 102].

In addition, based on the additive approach, a few studies address how to compute the final trust value by considering appropriate metrics. For example, later transactions are more important [102]; the evaluation approach should provide incentive to consistently good quality services and punish malicious service providers [90, 95]. Some other studies also consider context factors, e.g. the new transaction amount and service category [88], the rater's profile and location [20], or the relationship between the rater's group and the ratee [26].

- *Category 2* addresses the subjective property of trust for trust rating aggregation, e.g. the work in [40, 54], where subjective probability theory is adopted in trust evaluation.
- The approaches in *Category 3* (e.g. [82]) adopt Bayesian systems, which take binary ratings as input and compute reputation scores by statistically updating beta probability density functions (PDF).

- *Category 4* uses flow models, e.g. in [17, 25, 28, 32, 84, 86, 99, 102, 105], which compute the trust of a target through some intermediate participants and the trust dependency between them.
- While each of the above categories calculates a crisp value, *the last category* adopts fuzzy models, e.g. in [20, 90], where membership functions are used to determine the trustworthiness of targets.

### 2.5.2 Trust Structure Based Taxonomy

According to the general structure of trust described in Section 2.1, the trust evaluation approaches presented in Section 2.4 can be categorized into the *first quadrant* of Fig. 2.1. This is not a big surprise since each trust evaluation approach in Section 2.4 focuses on trust in a specific environment (e.g. e-commerce, P2P networks, service-oriented computing, multi-agent systems or social networks), and reflective and macro-social trust belongs to the first quadrant. In contrast, the second and third quadrants focus on primary (taken-for-granted) trust, and there is no necessity to have any trust evaluation approach in these quadrants. The fourth quadrant focuses on self trust evaluation.

As the topic of my thesis is about trust evaluation in service-oriented environments, all my proposed trust evaluation approaches belong to the first quadrant of Fig. 2.1.

### 2.5.3 Trust Bases Based Taxonomy

According to the bases of trust proposed in Section 2.2, the trust evaluation approaches presented in Section 2.4 can be analyzed as follows to find out which base of trust is adopted in each trust evaluation approach. Some approaches may be based on more than one bases of trust.

- *Dispositional Trust* focuses on the personality of a truster, with the assumption of a relatively stable personality characteristic, like the model in [82].

- *History-based Trust* is the most widely adopted trust base in trust evaluation. For example, it has been taken into account in [17, 19, 36, 44, 61, 65, 77, 80, 85, 88, 89, 90, 95, 102, 105].
- *Third Parties as Conduits of Trust* is another widely adopted trust base to evaluate trust. For example, it has been adopted by the models in [25, 28, 32, 36, 40, 77, 82, 84, 86, 99, 105].
- *Category-based Trust* addresses the information regarding a trustee's membership in a social or organizational category, e.g. in [77].
- *Role-based Trust* uses the knowledge that a trustee occupies a particular role in the organization, e.g. the work in [20, 36, 89].
- *Rule-based Trust* specifies formal or informal rules, which can determine trust, like the models in [26, 40, 90].

## 2.6 Conclusions

This chapter provides a general overview of the research studies on trust. Conceptually, we present the general structure of trust, the bases of trust and the concepts of trust in different disciplines. The general structure of trust presents a general cross-disciplinary analysis of trust, and provides a general picture containing all kinds of trust. The bases of trust illustrate what leads to the emergence of trust. The concepts of trust present different aspects of trust from the different viewpoints of different disciplines. In addition, the typical trust evaluation methods are introduced in a variety of online applications, including e-commerce, P2P networks, service-oriented computing, multi-agent systems and social networks. Finally, these trust evaluation methods can be categorized into different taxonomies.



# A Trust Vector Approach in Service-Oriented Applications

---

Trust is an important factor in service-oriented applications that can be used to indicate the trustworthiness of future services. In most existing trust evaluation models [19, 44, 82, 85, 88, 90, 92, 95, 97, 102], a single trust value (e.g. a value in the range of [0,1]) is computed to reflect the global trust level of a target accumulated in a certain time period (e.g. in the latest 6 months). The calculation of the final trust value is based on either all the ratings given for the latest time period [44, 95] or the current trust value for previous transactions and the rating for the latest transaction [88, 92].

Single trust value mechanisms are easy to use in trust-oriented service comparison and selection. However, a single trust value computed by a service management authority cannot depict the real trust level very well under certain circumstances. For example, if there are two service providers  $A$  and  $B$  with their final trust values  $T_A \approx 0.7$  and  $T_B \approx 0.7$  (each of which is in the range of [0,1]), does it mean that both  $A$  and  $B$  have the same trust level? It is not true if  $A$ 's trust values are turning worse with an accumulated value of 0.7 while  $B$ 's trust values are becoming better. In this case,  $B$  is better than  $A$  in terms of predicting the trust level of a forthcoming transaction.

Thus, a good trust management system requires more comprehensive trust evaluation approaches providing more objective trust information that indicates not only the global trust level, but also the trust prediction relevant to forthcoming transactions. To serve this purpose, in this chapter we propose a *service trust vector* consisting of a set

of values, such as *final trust level*, *service trust trend* and *service performance consistency level*, which is applicable to e-commerce or e-service environments. We also conduct empirical experiments to study the properties of our proposed approaches. In addition, fuzzy regression is adopted in trust vector evaluation instead of classical regression. In classical regression analysis, the deviations between the observed and estimated data are assumed to be subject to random errors. However, in trust evaluation, these deviations are frequently caused by human subjective judgement or imprecise observations [66]. All these reasons make it necessary to introduce fuzzy regression.

This chapter is organized as follows. In Section 3.1, we propose our service trust vector, which consists of three values: *final trust level*, *service trust trend* and *service performance consistency level*. In addition, some empirical studies are also presented to further illustrate the properties of our model. In Section 3.2, we adopt fuzzy regression instead of classical regression to compute the service trust vector. Finally, Section 3.3 concludes our work in this chapter.

### 3.1 Service Trust Vector and Its Evaluation

In cognitive science, it has been pointed out that people are motivated to maintain consistency among their actions [93]. This consistency provides the rationale that the performance of a service provider during a time interval can be consistent, implying a consistent trust level.

In this section, a trust vector approach is proposed to depict the trust level with three values: *Final Trust Level (FTL)*, *Service Trust Trend (STT)* and *Service Performance Consistency Level (SPCL)*.

### 3.1.1 Final Trust Level (*FTL*) Evaluation

The calculation of *FTL* follows a common principle, which is termed the *recency effect*<sup>1</sup> in cognitive science [93]. This principle appears in a number of studies in service-oriented applications [51, 59, 102].

**Principle 1:** The final trust value should be computed by taking the trust ratings in a recent time period into account, with more weight given to the ratings of later services.

**Definition 1:** Based on this principle, the *FTL* value for the time interval  $[t_1, t_n]$  can be calculated as:

$$T_{FTL}^{[t_1, t_n]} = \frac{\sum_{i=1}^n w_{t_i} \cdot R^{(t_i)}}{\sum_{i=1}^n w_{t_i}}, \quad (3.1)$$

where  $t_i \in [t_1, t_n]$  and  $w_{t_i}$  can be calculated as the exponential moving average [33]:

$$w_{t_i} = \alpha^{t_n - t_i}, \quad 0 < \alpha \leq 1. \quad (3.2)$$

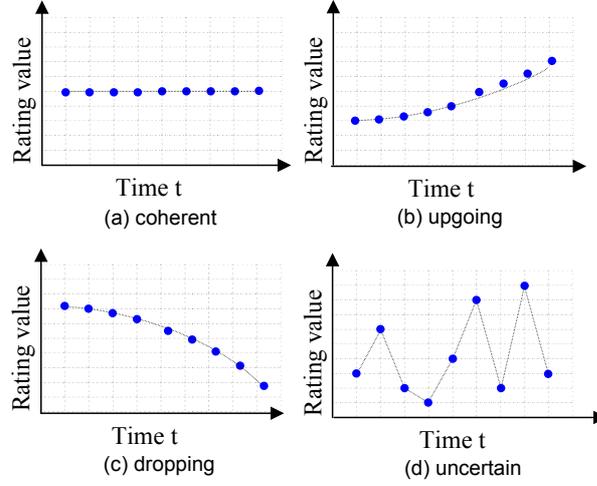
Actually, most existing single trust value methods (e.g. the methods proposed in [90, 102]) can be adopted to compute the *FTL* value if they are based on non-binary ratings  $\{R^{(t_i)}\}$  and follow Principle 1.

### 3.1.2 Service Trust Trend (*STT*) Evaluation

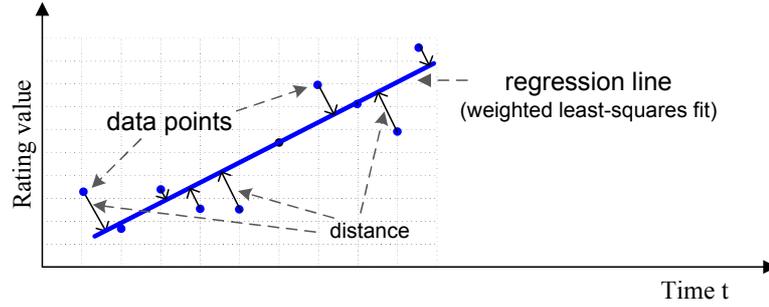
*STT* aims to illustrate the trend of service trust value changes in a given time interval. Some typical cases of *STT* are depicted in Fig. 3.1, which are “*coherent*”, “*upgoing*”, “*dropping*” and “*uncertain*” in sequence.

In order to evaluate the *STT* of a set of ratings  $\{R^{(t_i)} | t_i \in [t_1, t_n]\}$  for the time interval  $[t_1, t_n]$ , following Principle 1, we design a *weighted least square linear regression* method [51], as illustrated in Fig. 3.2. This method is used to obtain the best-fit straight line from a set of data points  $\{(t_i, R^{(t_i)})\}$ . It is characterized by the sum of weighted squared residuals with its least value, where a residual is the distance from a

<sup>1</sup>In the literature, recency effect is also known as *temporal sensitivity* [61].



**Figure 3.1:** Several *STT* cases



**Figure 3.2:** Weighted least square linear regression

data point to the regression line (see Fig. 3.2). Once the regression line is obtained, its slope is taken as the *STT* value.

Let  $(t_1, R^{(t_1)})$ ,  $(t_2, R^{(t_2)})$ ,  $\dots$ ,  $(t_n, R^{(t_n)})$  be the given data points within a time interval  $[t_1, t_n]$ , where  $R^{(t_i)} \in [0, 1]$  is the trust rating for the service delivered during a short time period  $t_i$  ( $t_i < t_{i+1}$ ,  $t_1 = t_{start}$  and  $t_n = t_{end}$ ). In general,  $R^{(t_i)}$  can be the value aggregated from a set of ratings for the services delivered at  $t_i$  (e.g. a day, or a week) [87]. In addition, the *regression line* can be represented as

$$R = p_{a_0} + p_{a_1}t, \quad (3.3)$$

where  $p_{a_0}$  and  $p_{a_1}$  are constants to be determined, and  $p_{a_1}$  represents the *STT* value. As the distance from point  $(t_i, R^{(t_i)})$  to the regression line is

$$d_{t_i} = \frac{|R^{(t_i)} - p_{a_0} - p_{a_1}t_i|}{\sqrt{1 + p_{a_1}^2}}, \quad (3.4)$$

the *sum of squares of the distance* can be defined as follows.

**Definition 2:** Based on the method of weighted least squares, the *sum of squares of the distance* can be calculated as follows:

$$S_{dis} = \sum_{i=1}^n w_{t_i}^2 d_{t_i}^2 = \sum_{i=1}^n \frac{w_{t_i}^2 (R^{(t_i)} - p_{a_0} - p_{a_1}t_i)^2}{1 + p_{a_1}^2}. \quad (3.5)$$

Now the task is to minimize the *sum of squares of the distance*  $S_{dis}$  with respect to the parameters  $p_{a_0}$  and  $p_{a_1}$ , with the method of undetermined coefficients.

As function  $S_{dis}$  is continuous and differentiable, based on Fermat's theorem in real analysis, the minimization point of  $S_{dis}$  makes the first derivative of function  $S_{dis}$  zero, and the second derivative positive, which could be easily proved since  $p_{a_1}$  is very small here. To serve this purpose, we differentiate  $S_{dis}$  with respect to  $p_{a_0}$  and  $p_{a_1}$ , and set the results to zero, which gives

$$\frac{\partial S_{dis}}{\partial p_{a_0}} = -2 \sum_{i=1}^n \frac{w_{t_i}^2 (R^{(t_i)} - p_{a_0} - p_{a_1}t_i)}{1 + p_{a_1}^2} = 0, \quad (3.6)$$

and

$$\frac{\partial S_{dis}}{\partial p_{a_1}} = -2 \sum_{i=1}^n \frac{w_{t_i}^2 (R^{(t_i)} - p_{a_0} - p_{a_1}t_i)(p_{a_1}R^{(t_i)} - p_{a_0}p_{a_1} + t_i)}{(1 + p_{a_1}^2)^2} = 0, \quad (3.7)$$

i.e.

$$\sum_{i=1}^n w_{t_i}^2 R^{(t_i)} - p_{a_0} \sum_{i=1}^n w_{t_i}^2 - p_{a_1} \sum_{i=1}^n w_{t_i}^2 t_i = 0, \quad (3.8)$$

and

$$\begin{aligned}
& \sum_{i=1}^n w_{t_i}^2 t_i R^{(t_i)} + p_{a_0} p_{a_1}^2 \sum_{i=1}^n w_{t_i}^2 t_i + p_{a_0}^2 p_{a_1} \sum_{i=1}^n w_{t_i}^2 \\
& \quad - 2p_{a_0} p_{a_1} \sum_{i=1}^n w_{t_i}^2 R^{(t_i)} - p_{a_1}^2 \sum_{i=1}^n w_{t_i}^2 t_i R^{(t_i)} \\
& + p_{a_1} \sum_{i=1}^n w_{t_i}^2 R^{(t_i)^2} - p_{a_0} \sum_{i=1}^n w_{t_i}^2 t_i - p_{a_1} \sum_{i=1}^n w_{t_i}^2 t_i^2 = 0. \tag{3.9}
\end{aligned}$$

Eqs. (3.8) and (3.9) can be solved for the unknown  $p_{a_0}$  and  $p_{a_1}$ , by substituting  $p_{a_0}$  from Eq. (3.8) into Eq. (3.9) to obtain

$$p_{a_1}^2 + \frac{S_{wr2} S_w - S_{wr}^2 + S_{wt}^2 - S_{wt2} S_w}{S_{wr} S_{wt} - \sum_{i=1}^n w_{t_i}^2 t_i R^{(t_i)} S_w} p_{a_1} - 1 = 0, \tag{3.10}$$

where  $S_w = \sum_{i=1}^n w_{t_i}^2$ ,  $S_{wt} = \sum_{i=1}^n w_{t_i}^2 t_i$ ,  $S_{wr} = \sum_{i=1}^n w_{t_i}^2 R^{(t_i)}$ ,  $S_{wt2} = \sum_{i=1}^n w_{t_i}^2 t_i^2$ , and  $S_{wr2} = \sum_{i=1}^n w_{t_i}^2 R^{(t_i)^2}$ . Obviously, it is easy to obtain a very small real solution of Eq. (3.10) about  $p_{a_1}$ .

Furthermore, by substituting the solution of  $p_{a_1}$  back to Eq. (3.8), we can obtain

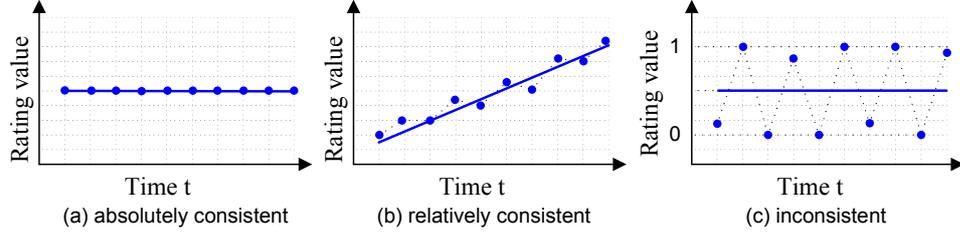
$$p_{a_0} = \frac{\sum_{i=1}^n w_{t_i}^2 R^{(t_i)} - p_{a_1} \sum_{i=1}^n w_{t_i}^2 t_i}{\sum_{i=1}^n w_{t_i}^2}. \tag{3.11}$$

Thus, based on the weighted least square linear regression method, we can obtain the  $STT$  value  $T_{STT} = p_{a_1}$ , which is determined from Eq. (3.10). However, in order to determine the four cases of  $STT$  depicted in Fig. 3.1, another factor  $SPCL$  should be taken into account.

### 3.1.3 Service Performance Consistency Level ( $SPCL$ ) Evaluation

The  $SPCL$  value indicates the consistency level of the service trust ratings in a certain time interval. Some typical  $SPCL$  cases are depicted in Fig. 3.3. In sequence, they are “absolutely consistent”, “relatively consistent” and “inconsistent”.

Prior to presenting the  $SPCL$  evaluation method in detail, we firstly introduce some



**Figure 3.3:** Several *SPCL* cases

definitions.

**Definition 3:** The *predicted value of regression line* is

$$V_{pr}(t_i) = p_{a_0} + p_{a_1} t_i, \quad (3.12)$$

where  $p_{a_0}$  and  $p_{a_1}$  are decided by Eqs. (3.11) and (3.10) respectively.

**Definition 4:** Following Principle 1, the *weighted average distance* for the time interval  $[t_1, t_n]$  is

$$V_{t_{dis}}^{[t_1, t_n]} = \frac{\sum_{i=1}^n w_{t_i} |R^{(t_i)} - V_{pr}(t_i)|}{\sqrt{1 + p_{a_1}^2} \sum_{i=1}^n w_{t_i}} \quad (3.13)$$

$$= \frac{\sum_{i=1}^n w_{t_i} |R^{(t_i)} - (p_{a_0} + p_{a_1} t_i)|}{\sqrt{1 + p_{a_1}^2} \sum_{i=1}^n w_{t_i}} \quad (3.14)$$

for  $n$  trust ratings  $\{R^{(t_i)}\}$  of services delivered in the time interval  $[t_1, t_n]$ .

Now let us introduce a principle about the *SPCL* evaluation as follows.

**Principle 2:** The *SPCL* value  $T_{SPCL}^{[t_1, t_n]}$  is a monotonically decreasing function of the weighted average distance  $V_{t_{dis}}^{[t_1, t_n]}$ .

According to Principle 2, we have the following *SPCL* evaluation formula.

**Definition 5:** The *SPCL* value for the time interval  $[t_1, t_n]$  is

$$T_{SPCL}^{[t_1, t_n]} = 1 - 2V_{t_{dis}}^{[t_1, t_n]} = 1 - 2 \frac{\sum_{i=1}^n w_{t_i} |R^{(t_i)} - (p_{a_0} + p_{a_1} t_i)|}{\sqrt{1 + p_{a_1}^2} \sum_{i=1}^n w_{t_i}}. \quad (3.15)$$

Obviously, we have  $T_{SPCL}^{[t_1, t_n]} \in [0, 1]$ .

From the above definition, the cases of *SPCL* can be determined as follows:

1. If  $\epsilon_{SPCL_1} < T_{SPCL} \leq 1$  ( $0 \ll \epsilon_{SPCL_1} < 1$  is the threshold), *SPCL* is *absolutely consistent* (refer to Fig. 3.3(a)),
2. If  $\epsilon_{SPCL_2} < T_{SPCL} < \epsilon_{SPCL_1}$  ( $0 < \epsilon_{SPCL_2} \ll 1$  is the threshold), *SPCL* is *relatively consistent* (refer to Fig. 3.3(b)), i.e. the service performance is consistent in a certain level.
3. If  $T_{SPCL} < \epsilon_{SPCL_2}$ , *SPCL* is *inconsistent* (refer to Fig. 3.3(c)), i.e. the service performance is not consistent.

Now, with  $T_{STT}$  and  $T_{SPCL}$ , the four cases of *STT* can be determined as follows.

1. If  $T_{SPCL} > \epsilon_{SPCL_2}$  and  $|T_{STT}| < \epsilon_{STT}$  ( $0 < \epsilon_{STT} \ll 1$  is the threshold), *STT* is *coherent* (refer to Fig. 3.1(a)), i.e. the service trust rating remains at the same level.
2. If  $T_{SPCL} > \epsilon_{SPCL_2}$  and  $T_{STT} > \epsilon_{STT}$ , *STT* is *up-going* (refer to Fig. 3.1(b)), i.e. the service trust rating is becoming better.
3. If  $T_{SPCL} > \epsilon_{SPCL_2}$  and  $T_{STT} < -\epsilon_{STT}$ , *STT* is *dropping* (refer to Fig. 3.1(c)), i.e. the service trust rating is turning worse.
4. If  $T_{SPCL} < \epsilon_{SPCL_2}$ , *STT* is *uncertain* (refer to Fig. 3.1(d)), i.e. the service trust rating is not reliable.

### 3.1.4 Service Trust Vector

Based on the above discussions, we can define the service trust vector as follows.

**Definition 6:** The *service trust vector*  $\overline{T}^{[t_1, t_n]}$  for the trust ratings given in time interval  $[t_1, t_n]$  is

$$\overline{T}^{[t_1, t_n]} = \langle T_{FTL}^{[t_1, t_n]}, T_{STT}^{[t_1, t_n]}, T_{SPCL}^{[t_1, t_n]} \rangle, \quad (3.16)$$

where  $T_{FTL}^{[t_1, t_n]}$  is defined in Eq. (3.1),  $T_{STT}^{[t_1, t_n]}$  is decided by Eq. (3.10), and  $T_{SPCL}^{[t_1, t_n]}$  is defined in Eq. (3.15).

With trust vectors, all service providers form a partial order set. Given two service providers  $P_j$  and  $P_k$  with their service trust vectors

$$\bar{T}_j^{[t_1, t_n]} = \langle T_{FTL_j}^{[t_1, t_n]}, T_{STT_j}^{[t_1, t_n]}, T_{SPCL_j}^{[t_1, t_n]} \rangle$$

and

$$\bar{T}_k^{[t_1, t_n]} = \langle T_{FTL_k}^{[t_1, t_n]}, T_{STT_k}^{[t_1, t_n]}, T_{SPCL_k}^{[t_1, t_n]} \rangle,$$

they are comparable in the following cases:

**Property 1:** If  $T_{STT_j}^{[t_1, t_n]} = T_{STT_k}^{[t_1, t_n]}$ ,  $T_{SPCL_j}^{[t_1, t_n]} = T_{SPCL_k}^{[t_1, t_n]}$ , and  $T_{FTL_j}^{[t_1, t_n]} < T_{FTL_k}^{[t_1, t_n]}$ ,  $P_k$  is preferable, which is denoted as  $P_k > P_j$  or  $P_j < P_k$ .

**Property 2:** If  $T_{FTL_j}^{[t_1, t_n]} = T_{FTL_k}^{[t_1, t_n]}$ ,  $T_{SPCL_j}^{[t_1, t_n]} = T_{SPCL_k}^{[t_1, t_n]}$ , and  $T_{STT_j}^{[t_1, t_n]} < T_{STT_k}^{[t_1, t_n]}$ , then  $P_j < P_k$ .

**Property 3:** If  $T_{FTL_j}^{[t_1, t_n]} = T_{FTL_k}^{[t_1, t_n]}$ ,  $T_{STT_j}^{[t_1, t_n]} = T_{STT_k}^{[t_1, t_n]}$ , and  $T_{SPCL_j}^{[t_1, t_n]} < T_{SPCL_k}^{[t_1, t_n]}$ , then  $P_j < P_k$ .

In addition, when the two values of a service vector element are approximately equal, we can compare the two trust vectors in the following cases.

**Property 4:** If

$$|T_{STT_j}^{[t_1, t_n]} - T_{STT_k}^{[t_1, t_n]}| < \epsilon_{Vector_1}, \quad (3.17)$$

$$|T_{SPCL_j}^{[t_1, t_n]} - T_{SPCL_k}^{[t_1, t_n]}| < \epsilon_{Vector_2}, \quad (3.18)$$

and

$$|T_{FTL_j}^{[t_1, t_n]} - T_{FTL_k}^{[t_1, t_n]}| < \epsilon_{Vector_3}, \quad (3.19)$$

$P_j$  and  $P_k$  are both preferable, which is denoted as  $P_k = P_j$ , where  $0 < \epsilon_{Vector_1}, \epsilon_{Vector_2}, \epsilon_{Vector_3} \ll 1$  are thresholds that can be specified by service clients or trust management authorities.

**Property 5:** If

$$|T_{STT_j}^{[t_1, t_n]} - T_{STT_k}^{[t_1, t_n]}| < \epsilon_{Vector_1}, \quad (3.20)$$

$$|T_{SPCL_j}^{[t_1, t_n]} - T_{SPCL_k}^{[t_1, t_n]}| < \epsilon_{Vector_2}, \quad (3.21)$$

and

$$T_{FTL_j}^{[t_1, t_n]} + \epsilon_{Vector_3} < T_{FTL_k}^{[t_1, t_n]}, \quad (3.22)$$

$P_k$  is preferable, which is denoted as  $P_k > P_j$  or  $P_j < P_k$ .

**Property 6:** If

$$|T_{FTL_j}^{[t_1, t_n]} - T_{FTL_k}^{[t_1, t_n]}| < \epsilon_{Vector_3}, \quad (3.23)$$

$$|T_{SPCL_j}^{[t_1, t_n]} - T_{SPCL_k}^{[t_1, t_n]}| < \epsilon_{Vector_2}, \quad (3.24)$$

and

$$T_{STT_j}^{[t_1, t_n]} + \epsilon_{Vector_1} < T_{STT_k}^{[t_1, t_n]}, \quad (3.25)$$

then  $P_j < P_k$ .

**Property 7:** If

$$|T_{FTL_j}^{[t_1, t_n]} - T_{FTL_k}^{[t_1, t_n]}| < \epsilon_{Vector_3}, \quad (3.26)$$

$$|T_{STT_j}^{[t_1, t_n]} - T_{STT_k}^{[t_1, t_n]}| < \epsilon_{Vector_1}, \quad (3.27)$$

and

$$T_{SPCL_j}^{[t_1, t_n]} + \epsilon_{Vector_2} < T_{SPCL_k}^{[t_1, t_n]}, \quad (3.28)$$

then  $P_j < P_k$ .

### 3.1.5 Experiments on Trust Vector

In this section, we illustrate the results of conducted simulations to study the proposed service trust vector approach, and explain why the service trust vector is necessary and important.

In these experiments, we set  $\epsilon_{SPCL_1} = 0.94$  and  $\epsilon_{SPCL_2} = 0.85$ , which are the

**Table 3.1:** Trust vectors in Experiment 1 on Trust Vector

	$T_{FTL}$	$T_{STT}$	$STT$	$T_{SPCL}$	$SPCL$
$P_1$	0.6048	0.0029	<i>up-going</i>	0.9417	<i>absolutely consistent</i>
$P_2$	0.6034	0.0047	<i>up-going</i>	0.8929	<i>relatively consistent</i>
$P_3$	0.6036	0.0001	<i>coherent</i>	0.9573	<i>absolutely consistent</i>
$P_4$	0.6066	0.0004	<i>coherent</i>	0.9101	<i>relatively consistent</i>
$P_5$	0.6055	-0.0017	<i>dropping</i>	0.9510	<i>absolutely consistent</i>
$P_6$	0.6004	-0.0024	<i>dropping</i>	0.9337	<i>relatively consistent</i>
$P_7$	0.6084	0.0033	<i>up-going</i>	0.9468	<i>absolutely consistent</i>
$P_8$	0.6010	0.0037	<i>up-going</i>	0.9068	<i>relatively consistent</i>
$P_9$	0.6031	0.0002	<i>coherent</i>	0.9618	<i>absolutely consistent</i>
$P_{10}$	0.6097	0.0005	<i>coherent</i>	0.8904	<i>relatively consistent</i>
$P_{11}$	0.6034	-0.0022	<i>dropping</i>	0.9583	<i>absolutely consistent</i>
$P_{12}$	0.6055	-0.0015	<i>dropping</i>	0.9047	<i>relatively consistent</i>

thresholds to determine *absolutely consistent*, *relatively consistent*, *inconsistent SPCL* and *uncertain STT* (refer to the properties introduced in Section 3.1.3), and set  $\epsilon_{STT} = 0.0006$ , which is the threshold for determining *coherent*, *up-going* and *dropping STT* together with  $\epsilon_{SPCL_2}$  (refer to the properties introduced in Section 3.1.3). Meanwhile, we set the parameter  $\alpha = 0.95$  in the weighting function in Eq. (3.2).

### 3.1.5.1 Experiment 1 on Trust Vector

In this experiment, we aim to illustrate why the service trust vector is necessary by comparing our method with two existing approaches in [90, 102], because they are also based on non-binary ratings only and applied to service-oriented environments. Both the ratings and corresponding regression lines are plotted in Fig. 3.4. The computed trust vectors are listed in Table 3.1.

In comparison with the Sporas approach proposed in [102], we evaluate the trust level of six service providers  $P_1$  to  $P_6$ , with constant  $\theta = 5$ , acceleration factor  $\sigma = 25$ , the reputation of ratee  $R_i^{other} = 1$  and initial reputation  $R_0 = 0.1$ . According to Table 3.1, all six service providers  $P_1$  to  $P_6$  (see Fig. 3.4(a)-(f)) have almost the same  $T_{FTL}$ .

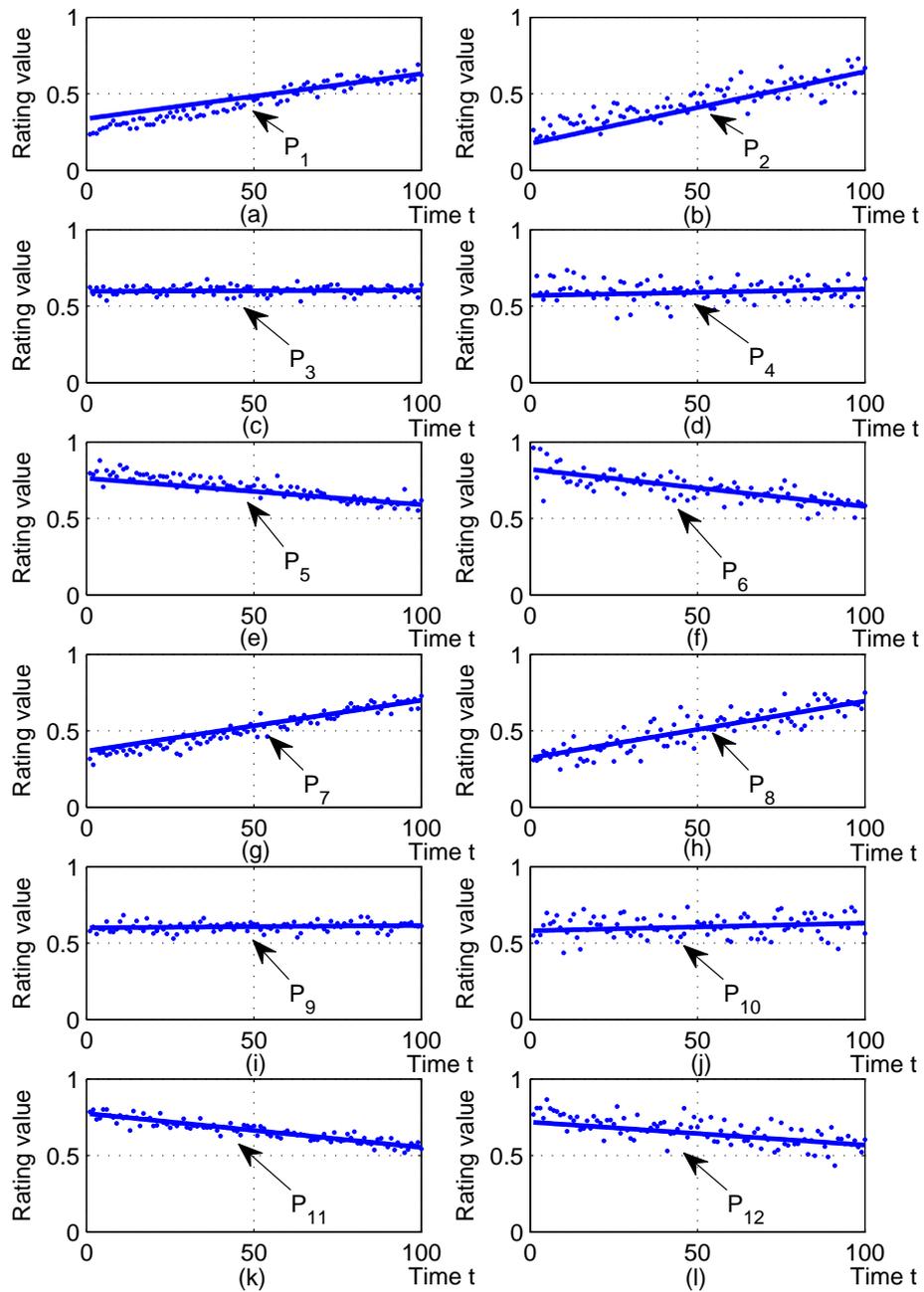
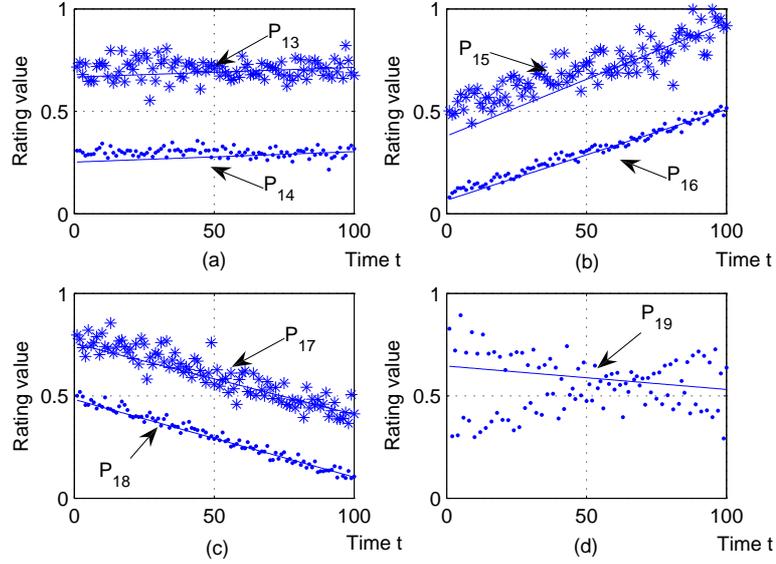


Figure 3.4: Experiment 1 on Trust Vector



**Figure 3.5:** Experiment 2 on Trust Vector

Therefore, they seemingly have the same trust level. However, they have different  $T_{STT}$  or  $T_{SPCL}$ . Based on the properties introduced in Section 3.1.3, we can determine the  $STT$  and  $SPCL$  as listed in Table 3.1, with which the six service providers  $P_1$  to  $P_6$  can be partially ordered:  $P_1 > P_3 > P_5$ ,  $P_2 > P_4 > P_6$ ,  $P_1 > P_2$ ,  $P_3 > P_4$  and  $P_5 > P_6$ .

Similarly, we compare our method with the approach proposed in [90] for the trust evaluation of six service providers  $P_7$  to  $P_{12}$  (see Fig. 3.4(g)-(l)), with the scale control factor  $\lambda = 1$ , and parameters  $\alpha = 2$  and  $\beta = 20$ . From Table 3.1, we can notice that the six service providers  $P_7$  to  $P_{12}$  have almost the same  $T_{FTL}$  but different  $T_{STT}$  or  $T_{SPCL}$ . Hence, the six service providers  $P_7$  to  $P_{12}$  can be partially ordered:  $P_7 > P_9 > P_{11}$ ,  $P_8 > P_{10} > P_{12}$ ,  $P_7 > P_8$ ,  $P_9 > P_{10}$  and  $P_{11} > P_{12}$ .

From this experiment, we can observe that under some circumstances a service trust vector can depict the trust level more precisely than a single trust value.

### 3.1.5.2 Experiment 2 on Trust Vector

In this experiment, we focus on four cases about *STT*. The computed results are listed in Table 3.2, and the best fit straight line for each service provider in this experiment is plotted in Fig. 3.5.

**Case 1:** In this case, as plotted in Fig. 3.5(a), there are two service providers  $P_{13}$  and  $P_{14}$ . According to Table 3.2, as  $T_{SPCL_{13}} > \epsilon_{SPCL_2}$ ,  $|T_{STT_{13}}| < \epsilon_{STT}$ ,  $T_{SPCL_{14}} > \epsilon_{SPCL_2}$  and  $|T_{STT_{14}}| < \epsilon_{STT}$ , based on the property introduced in Section 3.1.3, they both have *coherent STT*. In contrast, according to Table 3.2, as  $\epsilon_{SPCL_2} < T_{SPCL_{13}} < \epsilon_{SPCL_1}$ , based on the property introduced in Section 3.1.3,  $P_{13}$  has *relatively consistent SPCL*. Meanwhile, as  $T_{SPCL_{14}} > \epsilon_{SPCL_1}$ ,  $P_{14}$  has *absolutely consistent SPCL*.

**Case 2:** As plotted in Fig. 3.5(b), there are two service providers  $P_{15}$  and  $P_{16}$  in this case. Based on Table 3.2, as  $T_{SPCL_{15}} > \epsilon_{SPCL_2}$ ,  $T_{STT_{15}} > \epsilon_{STT}$ ,  $T_{SPCL_{16}} > \epsilon_{SPCL_2}$  and  $T_{STT_{16}} > \epsilon_{STT}$ , according to the property introduced in Section 3.1.3, they both have *up-going STT*. In contrast, based on Table 3.2, as  $\epsilon_{SPCL_2} < T_{SPCL_{15}} < \epsilon_{SPCL_1}$ , according to the property introduced in Section 3.1.3,  $P_{15}$  has *relatively consistent SPCL*. Meanwhile, as  $T_{SPCL_{16}} > \epsilon_{SPCL_1}$ ,  $P_{16}$  has *absolutely consistent SPCL*.

**Case 3:** In this case, as plotted in Fig. 3.5(c), there are two service providers  $P_{17}$  and  $P_{18}$ . According to Table 3.2, as  $T_{SPCL_{17}} > \epsilon_{SPCL_2}$ ,  $T_{STT_{17}} < -\epsilon_{STT}$ ,  $T_{SPCL_{18}} > \epsilon_{SPCL_2}$  and  $T_{STT_{18}} < -\epsilon_{STT}$ , according to the property introduced in Section 3.1.3, they both have *dropping STT*. In contrast, according to Table 3.2, as  $\epsilon_{SPCL_2} < T_{SPCL_{17}} < \epsilon_{SPCL_1}$ , based on the property introduced in Section 3.1.3,  $P_{17}$  has *relatively consistent SPCL*. Meanwhile, as  $T_{SPCL_{18}} > \epsilon_{SPCL_1}$ ,  $P_{18}$  has *absolutely consistent SPCL*.

**Case 4:** As plotted in Fig. 3.5(d), there is one service provider  $P_{19}$  in this case. According to Table 3.2, as  $T_{SPCL_{19}} < \epsilon_{SPCL_2}$ , according to the properties introduced

in Section 3.1.3, it has *uncertain STT* and *inconsistent SPCL*.

**Table 3.2:** Experiment 2 on Trust Vector results

	$T_{FTL}$	$T_{STT}$	$STT$	$T_{SPCL}$	$SPCL$
$P_{13}$	0.7069	0.0005	<i>coherent</i>	0.9286	<i>relatively consistent</i>
$P_{14}$	0.2968	0.0005	<i>coherent</i>	0.9573	<i>absolutely consistent</i>
$P_{15}$	0.8364	0.0055	<i>up-going</i>	0.8948	<i>relatively consistent</i>
$P_{16}$	0.4288	0.0044	<i>up-going</i>	0.9693	<i>absolutely consistent</i>
$P_{17}$	0.4759	-0.0035	<i>dropping</i>	0.9301	<i>relatively consistent</i>
$P_{18}$	0.1737	-0.0038	<i>dropping</i>	0.9710	<i>absolutely consistent</i>
$P_{19}$	0.5472	-0.0011	<i>uncertain</i>	0.7874	<i>inconsistent</i>

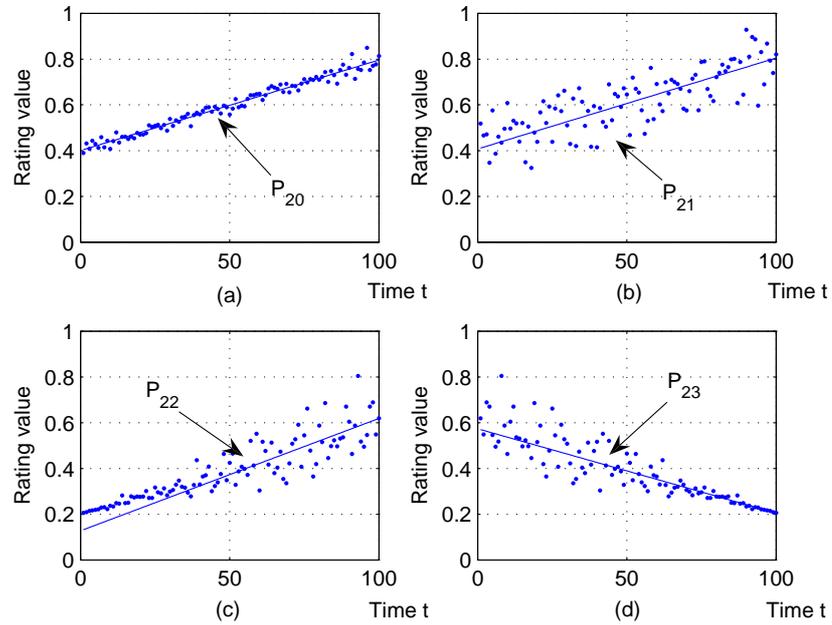
### 3.1.5.3 Experiment 3 on Trust Vector

In this experiment, we introduce two cases. In Case 1, we conduct an experiment to illustrate why  $SPCL$  is necessary and important. In Case 2, we aim to explain by examples why the introduction of a weight function in Eq. (3.2) is important. The computed results are listed in Table 3.3 and Table 3.4, and the best fit straight line for each service provider in the experiment is also plotted in Fig. 3.6.

**Case 1:** In this case, as plotted in Fig. 3.6(a)(b), there are two service providers  $P_{20}$  and  $P_{21}$ . According to Table 3.3, as  $T_{FTL_{20}} \approx T_{FTL_{21}}$ ,  $T_{STT_{20}} = T_{STT_{21}}$ , and  $T_{SPCL_{20}} > T_{SPCL_{21}}$ , we can conclude that  $P_{20} > P_{21}$ . This case indicates that the  $SPCL$  model is useful for depicting the trust history.

**Case 2:** As plotted in Fig. 3.6(c)(d), there are two service providers  $P_{22}$  and  $P_{23}$  in this case, which have the following property: the service trust rating  $R^{(t_i)}$  at time  $t_i$  of  $P_{22}$  in Fig. 3.6(c) is the same as the one at time  $100 - t_i$  of  $P_{23}$  in Fig. 3.6(d).

Without the weight function in Eq. (3.2), from Table 3.4 we obtain  $T_{FTL_{22}} = T_{FTL_{23}}$ ,  $T_{SPCL_{22}} = T_{SPCL_{23}}$  and  $T_{STT_{22}} = -T_{STT_{23}}$ . We can conclude that without



**Figure 3.6:** Experiment 3 on Trust Vector

**Table 3.3:** Experiment 3 on Trust Vector results with  $w_{t_i}$

	$T_{FTL}$	$T_{STT}$	$T_{SPCL}$
$P_{20}$	0.7232	0.0040	0.9591
$P_{21}$	0.7305	0.0040	0.8621
$P_{22}$	0.5299	0.0049	0.8661
$P_{23}$	0.2734	-0.0037	0.9731

**Table 3.4:** Experiment 3 on Trust Vector results without  $w_{t_i}$

	$T_{FTL}$	$T_{STT}$	$T_{SPCL}$
$P_{20}$	0.6003	0.0040	0.9687
$P_{21}$	0.6133	0.0035	0.8635
$P_{22}$	0.3971	0.0041	0.9126
$P_{23}$	0.3971	-0.0041	0.9126

the weight function in Eq. (3.2),  $P_{22}$  and  $P_{23}$  have the same  $T_{FTL}$ ,  $T_{SPCL}$  and the absolute value of  $T_{STT}$ .

However, after introducing the weight function, we have markedly different values, which are listed in Table 3.3. As plotted in Fig. 3.6(c)(d), the latest  $R^{(t_i)}$  of  $P_{22}$  is larger than the one of  $P_{23}$ , which is proven by  $T_{FTL_{22}} = 0.5299 > 0.3971$  (refer to  $T_{FTL_{22}}$  and  $T_{FTL_{23}}$  in Table 3.4)  $> T_{FTL_{23}} = 0.2734$  in Table 3.3 and Table 3.4. Similarly,  $T_{STT_{22}} = 0.0049 > 0.0041 > |T_{STT_{23}}| = 0.0037$  proves that  $P_{22}$  is becoming better and  $P_{23}$  is turning worse. Meanwhile, as  $T_{SPCL_{22}} = 0.8661 < 0.9126 < T_{SPCL_{23}} = 0.9731$ , it proves that  $P_{22}$  is turning less consistent and  $P_{23}$  is becoming more consistent.

So we can see that with the weight function in Eq. (3.2), the service trust rating history can be described more precisely.

#### 3.1.5.4 Experiment 4 on Trust Vector

Until now we have considered only one time interval in all cases. In order to evaluate the trust history of service trust ratings better, multiple time intervals may have to be introduced, where each interval corresponds to the trust trend with a high  $T_{SPCL}$  value. There are four cases in this experiment. The computed results are listed in Table 3.5. The best fit straight line for each case with one service provider is plotted in Fig. 3.7.

With the  $T_{STT}$  in Table 3.5, according to the properties introduced in Section 3.1.3, we can determine the typical cases of  $STT$  listed in Table 3.5. It is easy to see that the trust ratings in each time interval have *absolutely consistent SPCL* (i.e. a high  $T_{SPCL}$  value).

From the above results, we can see that in the case of a dynamic trust trend, proper multiple time intervals should be determined so as to describe the transaction history better, which will be our focus in Section 4.2.

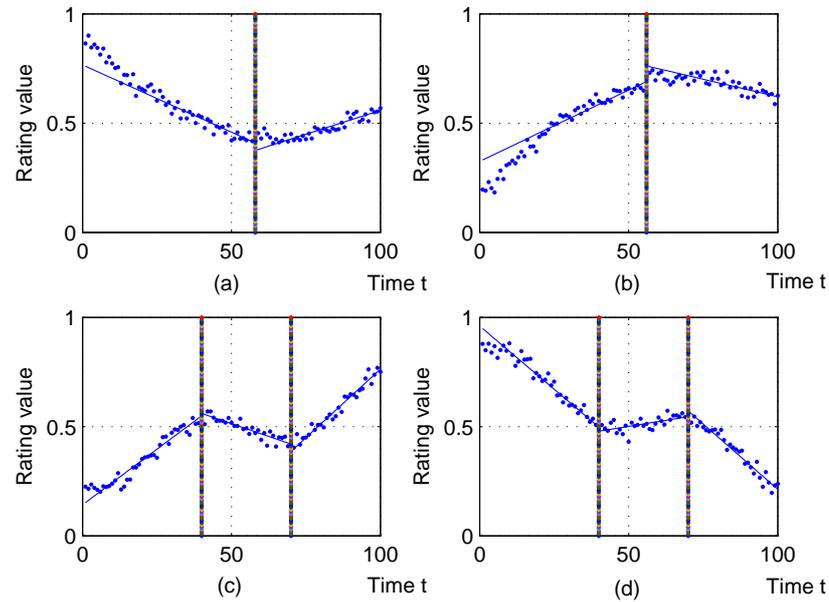


Figure 3.7: Experiment 4 on Trust Vector

## 3.2 Fuzzy Regression Based Trust Prediction

### 3.2.1 Background on Fuzzy Regression

In SOC environments, the service provider usually provides the QoS values before a transaction, which are the advertised QoS values, and then receives the aggregated rating value in  $[0, 1]$  about the delivered QoS values after the transaction. Such a rating of the delivered service is assumed to be able to indicate the success possibility or trustworthiness of the transaction. Therefore, this rating of the delivered service can be taken as a trust value. Assuming both the service providers and service clients are honest, if the transaction history data and the newly advertised QoS values are already known, the rating of the forthcoming delivered service can be predicted, since the rating of the delivered service is inherently related to the corresponding existing advertised QoS values prior to the forthcoming transaction.

For example, at eBay we plan to buy a new battery for HP pavilion dv2000 and

**Table 3.5:** Experiment 4 on Trust Vector results

		$T_{FTL}$	$T_{STT}$	$T_{STT}$	$T_{SPCL}$
Fig. 3.7(a)	[0, 58]	0.5107	-0.0062	<i>dropping</i>	0.9590
	[58, 100]	0.4998	0.0044	<i>up-going</i>	0.9663
Fig. 3.7(b)	[0, 56]	0.5842	0.0065	<i>up-going</i>	0.9534
	[56, 100]	0.6646	-0.0032	<i>dropping</i>	0.9615
Fig. 3.7(c)	[0, 40]	0.4159	0.0102	<i>up-going</i>	0.9545
	[40, 70]	0.4717	-0.0047	<i>dropping</i>	0.9683
	[70, 100]	0.6246	0.0126	<i>up-going</i>	0.9655
Fig. 3.7(d)	[0, 40]	0.6477	-0.0116	<i>dropping</i>	0.9622
	[40, 70]	0.5206	0.0022	<i>up-going</i>	0.9649
	[70, 100]	0.3449	-0.0120	<i>dropping</i>	0.9567

dv6000 laptops<sup>2</sup>. First of all, we focus on the transaction history and try to find out whether the seller is trustworthy. Before each transaction, the seller provides the product's price, shipping price, delivery time and all other QoS values. After the transaction, the buyer gives a feedback rating about the transaction quality. If a function between the advertised QoS values and the rating of the delivered service can be determined, then with the new QoS values (such as: Price US \$58.75; Shipping US \$10.95 and so on) the forthcoming rating can be predicted. That is very useful and important important to receive before the transaction.

Regression analysis [33] is a statistical technique for modeling and investigating the relationship between two or more variables. For example, here regression analysis can be used to build up a model that represents the rating of the delivered service as a function of a set of advertised QoS values. This model can then be used to predict the rating of the forthcoming delivered service with a new set of advertised QoS values.

In the classical regression method, a set of parameters of an unknown function  $f(x, \omega)$  can be estimated by making measurements of the function with an error at any

<sup>2</sup>[http://cgi.ebay.com/NEW-Battery-for-Hp-pavilion-dv2000-dv6000-V6000-12-cell0QQitemZ370143793214QQcmdZViewItemQQptZLH\\_DefaultDomain\\_0?hash=\\_Witem370143793214&\\_trksid=p3286.c0.m14&\\_trkparms=72%3A1234%7C66%3A2%7C65%3A12%7C39%3A1%7C240%3A1308%7C301%3A1%7C293%3A1%7C294%3A50](http://cgi.ebay.com/NEW-Battery-for-Hp-pavilion-dv2000-dv6000-V6000-12-cell0QQitemZ370143793214QQcmdZViewItemQQptZLH_DefaultDomain_0?hash=_Witem370143793214&_trksid=p3286.c0.m14&_trkparms=72%3A1234%7C66%3A2%7C65%3A12%7C39%3A1%7C240%3A1308%7C301%3A1%7C293%3A1%7C294%3A50)

point  $x_i$ :

$$y_i = f(x_i, \omega) + \epsilon_i, \quad (3.29)$$

where the error  $\epsilon_i$  is independent of  $x_i$  and is distributed according to a known density  $p_\omega(\epsilon)$ . Based on the observed data sample  $S = \{(x_i, y_i), i = 1, 2, \dots, n\}$ , the likelihood is given by:

$$P(S|\omega) = \sum_{i=1}^n \ln p_\omega(y_i - f(x_i, \omega)). \quad (3.30)$$

Assuming that the error is normally distributed with mean 0 and variance  $\delta$  [59], the likelihood is given by:

$$P(S|\omega) = -\frac{1}{2\delta^2} \sum_{i=1}^n (y_i - f(x_i, \omega))^2 - n \ln(\sqrt{2\pi}\delta) \quad (3.31)$$

Maximizing the likelihood in Eq. (3.31) is equivalent to minimizing

$$E(\omega) = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i, \omega))^2, \quad (3.32)$$

which is in fact the same as the estimation by the method of least squares. Namely, the regression line is estimated such that the sum of squares of the deviations between the observations and the regression line is minimized.

In classical regression analysis, the deviations between the observed and estimated data are assumed to be due to random errors. However, frequently these deviations are caused by the indefinite structure of the system, by imprecise observations or by human subjective judgement [66], which makes it necessary to introduce fuzzy regression.

Fuzzy regression can be quite useful in estimating the relationships among variables where the available data are very limited and imprecise, and variables are interacting in an uncertain, qualitative, and fuzzy way. Thus, it has considerably practical applications in many management and engineering problems.

Particularly, it is reported that the uncertainty in a system can be due to several

reasons [66]:

- The high complexity of the environment, which necessitates the adaptation of abstraction (granulation of information) for generalization purposes.
- The influence of human subjective judgement in the decision process or the involvement of human-machine interactions.
- Partially available information, due to miss-recording or inaccurate measurements.

The trust terms in service-oriented environments come with uncertainty as they are derived from the *human subjective judgement* - one of the above reasons. Therefore, fuzzy regression is useful to deal with trust in service-oriented environments.

### 3.2.2 Fuzzy Regression Model Parameters

Prior to presenting the fuzzy regression model in detail, some definitions about the parameters of the model should be introduced.

#### 3.2.2.1 Membership Function

The fuzzy number mentioned in this chapter is  $\tilde{A}(\alpha, C)$  with the following *membership function* [37],

$$\mu_{\tilde{A}}(x) = \begin{cases} \begin{cases} 1 - \frac{|x-\alpha|}{C}, & |x - \alpha| \leq C, \\ 0, & \text{otherwise,} \end{cases} & C > 0, \\ \begin{cases} 1, & x = \alpha, \\ 0, & \text{otherwise,} \end{cases} & C = 0, \end{cases} \quad (3.33)$$

According to Eq. (3.33), it is easy to prove that  $\lambda\tilde{A}$  is fuzzy number  $\tilde{A}(\lambda\alpha, |\lambda|C)$  and  $\tilde{A}_1 + \tilde{A}_2$  is fuzzy number  $\tilde{A}(\alpha_1 + \alpha_2, C_1 + C_2)$ . So if

$$\tilde{T}_i^* = \tilde{A}_0 + q_{i1}\tilde{A}_1 + \dots + q_{in}\tilde{A}_n, \quad (3.34)$$

then  $\tilde{T}_i^*$  is the fuzzy number

$$\tilde{T}_i^*(\alpha_0 + \sum_{j=1}^n q_{ij}\alpha_j, C_0 + \sum_{j=1}^n |q_{ij}|C_j). \quad (3.35)$$

### 3.2.2.2 Goodness-of-fit

**Definition 7:** Let  $A, B$  be the fuzzy sets in real space  $\mathbf{R}$ , then

$$h = \bigvee_{x \in \mathbf{R}} \{\mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x)\} \quad (3.36)$$

is the *goodness-of-fit* from  $A$  to  $B$ .

In fact, the goodness-of-fit is defined as the inner product here. According to Definition 7, the goodness-of-fit from fuzzy number  $\tilde{A}(\alpha_1, C_1)$  to fuzzy number  $\tilde{B}(\alpha_2, C_2)$  is

$$h = \begin{cases} 1 - \frac{|\alpha_1 - \alpha_2|}{C_1 + C_2}, & |\alpha_1 - \alpha_2| \leq C_1 + C_2, \\ 0, & \text{otherwise,} \end{cases} \quad (3.37)$$

Hence, the goodness-of-fit  $h_i$  from  $\tilde{T}_i(T_i, e_i)$  to  $\tilde{T}_i^*(\alpha_0 + \sum_{j=1}^n q_{ij}\alpha_j, C_0 + \sum_{j=1}^n |q_{ij}|C_j)$  in Eq. (3.35) is

$$h_i = \begin{cases} 1 - \frac{|T_i - (\alpha_0 + \sum_{j=1}^n q_{ij}\alpha_j)|}{C_0 + \sum_{j=1}^n |q_{ij}|C_j + e_i}, & |T_i - (\alpha_0 + \sum_{j=1}^n q_{ij}\alpha_j)| \leq C_0 + \sum_{j=1}^n |q_{ij}|C_j + e_i, \\ 0 & \text{otherwise,} \end{cases} \quad (3.38)$$

### 3.2.2.3 Fuzziness

**Definition 8:** Let  $\tilde{A}(\alpha, C)$  be the fuzzy number, then the *fuzziness* [48] of  $\tilde{A}$  is

$$S_{\tilde{A}} = \frac{1}{2}C. \quad (3.39)$$

The fuzziness measures how fuzzy, vague or unclear the fuzzy set is. According to

Definition 8, the fuzziness of  $\tilde{T}_i^*$  in Eq. (3.34) is

$$S_{\tilde{T}_i^*} = \frac{1}{2} \left( C_0 + \sum_{j=1}^n |q_{ij}| C_j \right). \quad (3.40)$$

### 3.2.3 Fuzzy Regression Methodology

Let the sample data be

$$\begin{aligned} q_{11}, q_{12}, \dots, q_{1n}; T_1 \\ q_{21}, q_{22}, \dots, q_{2n}; T_2 \\ \dots \\ q_{m1}, q_{m2}, \dots, q_{mn}; T_m \end{aligned} \quad (3.41)$$

where  $q_{i1}, q_{i2}, \dots, q_{in}$  are the advertised QoS values at time  $i$ , which are the input data.  $T_i$  is the corresponding rating of the delivered service, which is the output data, and  $i = 1, 2, \dots, m$ .

As there may be no established relation between the input and the output, in order to dovetail the model nicely with the real application the output is transformed into fuzzification, which makes the fuzzy output  $\tilde{T}_i$  be fuzzy number  $\tilde{T}_i(T_i, e_i)$ , where  $e_i$  depends on the application environment. In this chapter, the relation between the input and the output is estimated by the fuzzy linear regression as follows.

The corresponding general fuzzy linear regression model is

$$\tilde{T}_i^* = \tilde{A}_0 + q_{i1}\tilde{A}_1 + \dots + q_{in}\tilde{A}_n, \quad (3.42)$$

where  $\tilde{A}_j$  is fuzzy number  $\tilde{A}_j(\alpha_j, C_j)$ , which has the membership function in Eq. (3.33), and  $\tilde{T}_i^*$  is the fuzzy number defined in Eq. (3.35).

For parameters estimation, when the goodness-of-fit  $h_i$  is large enough, we try to minimize the fuzziness

$$\max_{1 \leq i \leq m} \left\{ \frac{1}{2} \left( C_0 + \sum_{j=1}^n |q_{ij}| C_j \right) \right\} \quad (3.43)$$

to estimate  $\alpha_0, \alpha_1, \dots, \alpha_n$  and  $C_0, C_1, \dots, C_n$ .

Since

$$\max_{1 \leq i \leq m} \left\{ \frac{1}{2} (C_0 + \sum_{j=1}^n |q_{ij}| C_j) \right\} \leq \frac{1}{2} C_0 + \frac{1}{2} \sum_{j=1}^n (\max_{1 \leq i \leq m} |q_{ij}|), \quad (3.44)$$

parameters estimation is transformed to linear programming

$$\min S = W_0 C_0 + W_1 C_1 + \dots + W_n C_n, \quad (3.45)$$

such that

$$h_i > H, \quad i = 1, 2, \dots, m, \quad (3.46)$$

$$C_j \geq 0, \quad j = 0, 1, \dots, n, \quad (3.47)$$

where  $H$  is established at the beginning, and

$$W_0 = \frac{1}{\sum_{k=1}^n (\max_{1 \leq i \leq m} |q_{ik}|)}, \quad (3.48)$$

$$W_j = \frac{\max_{1 \leq i \leq m} |q_{ij}|}{\sum_{k=1}^n (\max_{1 \leq i \leq m} |q_{ik}|)}. \quad (3.49)$$

From Eq. (3.38), the above multiple linear programming problem is

$$\min S = \sum_{j=1}^n W_j C_j \quad (3.50)$$

such that

$$\alpha_0 + \sum_{j=1}^n q_{ij} \alpha_j + (1-H)(C_0 + \sum_{j=1}^n |q_{ij}| C_j) \geq T_i - (1-H)e_i, \quad i = 1, 2, \dots, m \quad (3.51)$$

$$\alpha_0 + \sum_{j=1}^n q_{ij} \alpha_j - (1-H)(C_0 + \sum_{j=1}^n |q_{ij}| C_j) \leq T_i + (1-H)e_i, \quad i = 1, 2, \dots, m \quad (3.52)$$

$$C_j \geq 0, \quad j = 1, 2, \dots, n \quad (3.53)$$

From the linear programming above, the parameters can be estimated. Hence, the

prediction model is

$$\tilde{T}^* = \tilde{A}_0 + q_1 \tilde{A}_1 + \dots + q_n \tilde{A}_n, \quad (3.54)$$

and the center of  $\tilde{T}_i^*$  is

$$T_i^* = \alpha_0 + \sum_{j=1}^n q_{ij} \alpha_j. \quad (3.55)$$

Based on existing advertised QoS values and the rating of the delivered service, the fuzzy regression line can be obtained with Eq. (3.54). With the obtained fuzzy regression line and new advertised QoS values  $\{q_{m+1\ 1}, q_{m+1\ 2}, \dots, q_{m+1\ n}\}$ , the rating of the forthcoming delivered service can be predicted as

$$T_{m+1}^* = \alpha_0 + \sum_{j=1}^n q_{m+1\ j} \alpha_j. \quad (3.56)$$

This is valuable for the decision-making of service clients prior to transactions.

### 3.2.4 Fuzzy Regression Based Service Trust Vector

In Section 3.1 [51], only a single variable is considered. However, in this section, multiple variables are introduced, which generates a multi-variable fuzzy linear regression. It is easy to introduce the weight in Eq. (3.2) without complexity. Therefore, for the sake of simplicity, we omit the weight function in this section. Based on the results in Section 3.2.3, *FTL*, *STT* and *SPCL* are redefined as follows.

**Definition 9:** The *FTL* value can be calculated as:

$$T_{FTL} = \frac{\sum_{i=1}^m T_i}{m}, \quad (3.57)$$

where  $T_i$  is the trust value at time  $i$  ( $i = 1, 2, \dots, m$ ).

**Definition 10:** The *STT* value can be evaluated by the parameters of the first order of Eq. (3.54), i.e.

$$T_{STT} = \sum_{j=1}^n \alpha_j. \quad (3.58)$$

**Definition 11:** The *SPCL* value can be evaluated by the goodness-of-fit, which is defined in Eq. (3.38), i.e.

$$T_{SPCL} = \frac{\sum_{i=1}^m h_i}{m}, \quad (3.59)$$

where  $h_i$  is defined in Eq. (3.38).

Hence, the definition of the service trust vector is as follows.

**Definition 12:** The service trust vector  $\bar{T}$  consists of the *FTL* value  $T_{FTL}$ , the *STT* value  $T_{STT}$ , and the *SPCL* value  $T_{SPCL}$

$$\bar{T} = \langle T_{FTL}, T_{STT}, T_{SPCL} \rangle, \quad (3.60)$$

where  $T_{FTL}$  is defined in Eq. (3.57),  $T_{STT}$  is decided by Eq. (3.58), and  $T_{SPCL}$  is defined in Eq. (3.59).

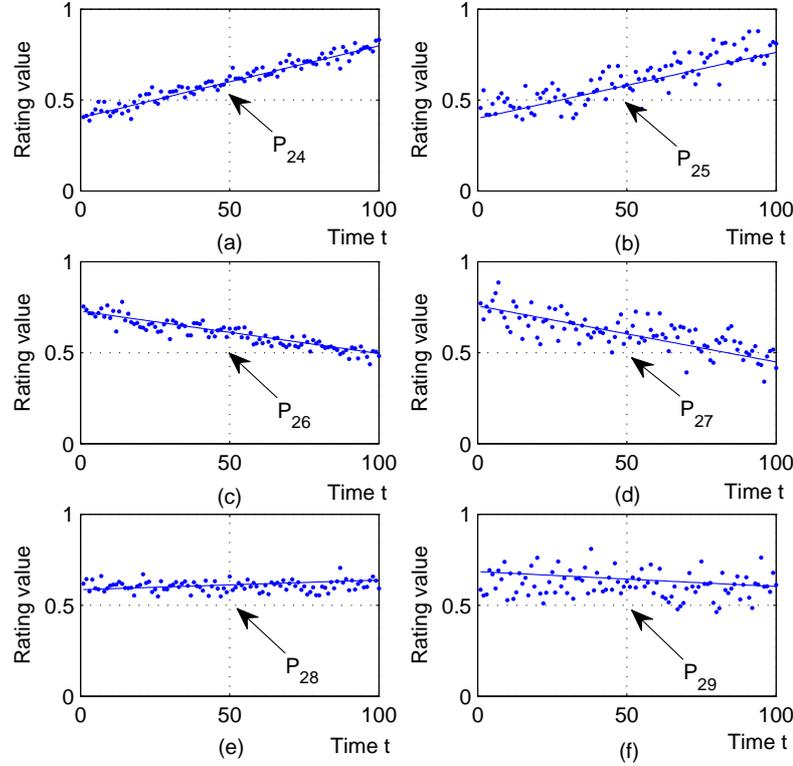
Moreover, with trust vectors, all service providers form a partial order set. Given two service providers  $P_i, P_j$  with service trust vectors  $\bar{T}_i = \langle T_{FTL_i}, T_{STT_i}, T_{SPCL_i} \rangle$ , and  $\bar{T}_j = \langle T_{FTL_j}, T_{STT_j}, T_{SPCL_j} \rangle$  respectively, they are comparable in the following cases:

**Property 8:** If  $T_{STT_i} = T_{STT_j}$ ,  $T_{SPCL_i} = T_{SPCL_j}$ , and  $T_{FTL_i} < T_{FTL_j}$ ,  $P_j$  is preferable. We denote it as  $P_j > P_i$  or  $P_i < P_j$ .

**Property 9:** If  $T_{FTL_i} = T_{FTL_j}$ ,  $T_{SPCL_i} = T_{SPCL_j}$ , and  $T_{STT_i} < T_{STT_j}$ ,  $P_j$  is preferable. This is denoted as  $P_j > P_i$  or  $P_i < P_j$ .

**Property 10:** If  $T_{FTL_i} = T_{FTL_j}$ ,  $T_{STT_i} = T_{STT_j}$ , and  $T_{SPCL_i} < T_{SPCL_j}$ ,  $P_j$  is preferable. We denote it as  $P_j > P_i$  or  $P_i < P_j$ .

In Section 3.1 [51], a regression line is built up to indicate the service trust level, based on the time variable and the service rating values. In contrast to the previous work in Section 3.1, in this section the fuzzy regression line is determined in multiple dimensional space, which consists of multiple independent variable axes of advertised QoS values and one dependent variable axis of the rating of the delivered service.



**Figure 3.8:** Experiment 1 on Fuzzy Regression Based Trust Prediction

### 3.2.5 Experiments on Fuzzy Regression Based Trust Prediction

In this section, we illustrate the results of conducted simulations to study the proposed service trust vector approach, and explain why the service trust vector is necessary and important. In addition, we explain why we adopt the fuzzy regression in this work.

#### 3.2.5.1 Experiment 1 on Fuzzy Regression Based Trust Prediction

In this experiment, we conduct an experiment with six cases to illustrate why the trust vector is necessary and important. Here we only take one QoS value, which is the time, in order to illustrate the fuzzy regression method in a two dimensional figure. We set  $H = 0.6$  and  $e = [0.01 \ 0.01 \ \dots \ 0.01]_{1 \times 100}$ . The computed results are listed in Table 3.6, and the center of the regression line for each service provider in this experiment

**Table 3.6:** Experiment 1 on Fuzzy Regression Based Trust Prediction results

	$T_{FTL}$	$T_{STT}$	$T_{SPCL}$
$P_{24}$	0.6096	0.0040	0.8760
$P_{25}$	0.6092	0.0036	0.8562
$P_{26}$	0.6027	-0.0023	0.8846
$P_{27}$	0.6104	-0.0031	0.8556
$P_{28}$	0.6057	0.0005	0.8640
$P_{29}$	0.6131	-0.0008	0.8474

is plotted in Fig. 3.8.

In each case, as plotted in Fig. 3.8, there is one service provider. According to Table 3.6, all six cases have almost the same  $T_{FTL}$ , but different  $T_{STT}$  or  $T_{SPCL}$ . Meanwhile, we can conclude that  $P_{24} > P_{25}$ ,  $P_{24} > P_{26}$ ,  $P_{25} > P_{27}$ ,  $P_{26} > P_{27}$ ,  $P_{28} > P_{26}$ ,  $P_{28} > P_{29}$ , and  $P_{29} > P_{27}$ . Namely, with solo  $FTL$ , it is not likely to depict the trust history exactly and compare service providers well.

Therefore, in this experiment, we can notice that the trust vector including  $T_{FTL}$ ,  $T_{STT}$  and  $T_{SPCL}$  can describe the history of trust data more precisely than the solo  $FTL$ .

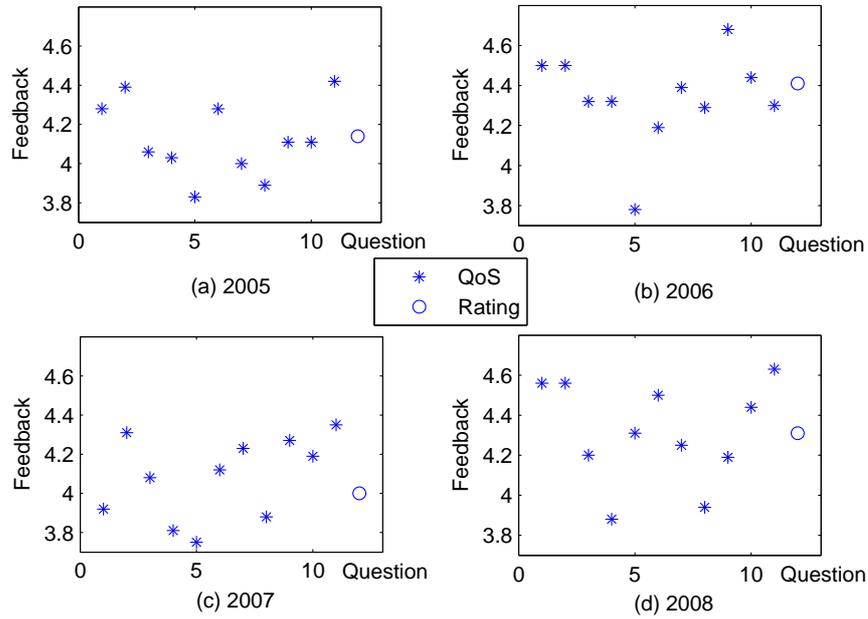
### 3.2.5.2 Experiment 2 on Fuzzy Regression Based Trust Prediction

In this experiment, we apply our model to predict the feedback of a learner's experience of a teacher at Macquarie University<sup>3</sup>, Sydney, Australia. Table 3.7 illustrates an example of the feedback of a learner's experience<sup>4</sup> of the same teacher in a course between 2005 and 2008, obtained from the Centre for Professional Development<sup>5</sup> at Macquarie University. The questionnaire is designed to collect students' feedback on a teacher's teaching quality. Questions 1 to 11 are based on generic attributes of teaching quality, such as: communicated clearly, enthusiasm, good learning atmosphere, constructive feedback etc, which can be taken as QoS values. Question 12 is "I would

<sup>3</sup><http://www.mq.edu.au/>

<sup>4</sup>[http://www.mq.edu.au/learningandteachingcentre/for\\_staff/teaching\\_eval/let.htm](http://www.mq.edu.au/learningandteachingcentre/for_staff/teaching_eval/let.htm)

<sup>5</sup><http://www.cpd.mq.edu.au/>



**Figure 3.9:** Feedback of a learner’s experience of the same teacher in a course between 2005 and 2008 from the Centre for Professional Development at Macquarie University

recommend a unit taught by this teacher to other students”, which can be considered as the overall quality value. These values are illustrated in Fig. 3.9.

Therefore, a fuzzy regression model can be built up to describe the relation between the input, i.e. the time and the feedback to Questions 1-11, and the output, i.e. the feedback to Question 12. In addition, the fuzzy regression model parameters are determined by the data from 2005 to 2007. Based on the fuzzy regression model, with the input QoS data of 2008, the corresponding output is then predicted. Compared with the feedback to Question 12, the goodness-of-fit result can be obtained.

Let  $H = 0.6$  and  $e_i = 0.1$ , and Eq. (3.50) becomes

$$S = 0.01962C_0 + 0.05886C_1 + 0.08829C_2 + 0.08829C_3 + 0.08476C_4 \quad (3.61)$$

$$+0.08476C_5 + 0.07514C_6 + 0.08397C_7 + 0.08613C_8 + 0.08417C_9 \quad (3.62)$$

$$+0.09182C_{10} + 0.08711C_{11} + 0.08672C_{12} \quad (3.63)$$

**Table 3.7:** Feedback of a learner's experience of a teacher for a course taken between 2005 and 2008 from Centre for Professional Development in Macquarie University

Year	QoS <sub>1</sub>	QoS <sub>2</sub>	QoS <sub>3</sub>	QoS <sub>4</sub>	QoS <sub>5</sub>	QoS <sub>6</sub>	QoS <sub>7</sub>	QoS <sub>8</sub>	QoS <sub>9</sub>	QoS <sub>10</sub>	QoS <sub>11</sub>	Overall rating	Number of respondents
2005	4.28	4.39	4.06	4.03	3.83	4.28	4.00	3.89	4.11	4.11	4.42	4.14	36
2006	4.50	4.50	4.32	4.32	3.78	4.19	4.39	4.29	4.68	4.44	4.30	4.41	28
2007	3.92	4.31	4.08	3.81	3.75	4.12	4.23	3.88	4.27	4.19	4.35	4.00	26
2008	4.56	4.56	4.20	3.88	4.31	4.50	4.25	3.94	4.19	4.44	4.63	4.31	16

From the linear programming, the fuzzy regression model is

$$\tilde{T}_i^* = \tilde{A}_0 + q_{i1}\tilde{A}_1 + \dots + q_{in}\tilde{A}_n, \quad (3.64)$$

where

$$\mu_{\tilde{A}_0}(x) = \begin{cases} 1, & x = -0.1036, \\ 0, & x \neq -0.1036, \end{cases} \quad (3.65)$$

$$\mu_{\tilde{A}_1}(x) = \begin{cases} 1, & x = -0.03569, \\ 0, & x \neq -0.03569, \end{cases} \quad (3.66)$$

$$\mu_{\tilde{A}_2}(x) = \begin{cases} 1, & x = 0.2128, \\ 0, & x \neq 0.2128, \end{cases} \quad (3.67)$$

$$\mu_{\tilde{A}_3}(x) = \begin{cases} 1, & x = -0.1469, \\ 0, & x \neq -0.1469, \end{cases} \quad (3.68)$$

$$\mu_{\tilde{A}_4}(x) = \begin{cases} 1, & x = 0.07865, \\ 0, & x \neq 0.07865, \end{cases} \quad (3.69)$$

$$\mu_{\tilde{A}_5}(x) = \begin{cases} 1, & x = 0.2084, \\ 0, & x \neq 0.2084, \end{cases} \quad (3.70)$$

**Table 3.8:** Goodness-of-fit results in Experiment 2 on Fuzzy Regression Based Trust Prediction

Year	Original rating	Prediction rating	Error percentage
2008	4.31	4.1418	3.9030%

$$\mu_{\tilde{A}_6}(x) = \begin{cases} 1, & x = -0.1315, \\ 0, & x \neq -0.1315, \end{cases} \quad (3.71)$$

$$\mu_{\tilde{A}_7}(x) = \begin{cases} 1, & x = -0.06863, \\ 0, & x \neq -0.06863, \end{cases} \quad (3.72)$$

$$\mu_{\tilde{A}_8}(x) = \begin{cases} 1, & x = 0.1396, \\ 0, & x \neq 0.1396, \end{cases} \quad (3.73)$$

$$\mu_{\tilde{A}_9}(x) = \begin{cases} 1, & x = 0.2033, \\ 0, & x \neq 0.2033, \end{cases} \quad (3.74)$$

$$\mu_{\tilde{A}_{10}}(x) = \begin{cases} 1, & x = 0.01623, \\ 0, & x \neq 0.01623, \end{cases} \quad (3.75)$$

$$\mu_{\tilde{A}_{11}}(x) = \begin{cases} 1, & x = 0.1504, \\ 0, & x \neq 0.1504, \end{cases} \quad (3.76)$$

$$\mu_{\tilde{A}_{12}}(x) = \begin{cases} 1, & x = 0.3659, \\ 0, & x \neq 0.3659, \end{cases} \quad (3.77)$$

From Eq. (3.64), the corresponding goodness-of-fit results are listed in Table 3.8 with an error percentage of 3.9030%. Obviously, we can see that the fuzzy regression model predicts well. With the data from more years, the prediction will become more accurate.

### 3.3 Conclusions

In Section 3.1 of this chapter, we propose a trust vector approach to service-oriented applications, which includes *final trust level (FTL)*, *service trust trend (STT)*, and *service performance consistency level (SPCL)*. Corresponding *STT* and *SPCL* evaluation methods are also proposed. From our analytical and empirical experiments, we can see that the proposed approach can depict trust history more precisely than exiting single trust value approaches. It offers more information to service clients for their decision-making in the selection of trustworthy service providers.

In addition, in Section 3.2, a fuzzy regression based trust vector approach is proposed. From our analytical and empirical experiments, we can see that the proposed fuzzy regression based trust vector approach can depict the trust level with more indications than exiting single trust value approaches, and predict the trustworthiness of a forthcoming transaction with its advertised QoS values and transaction history.

---

# Two Dimensional Trust Rating

## Aggregations

---

In the literature, in most existing trust evaluation models [19, 44, 82, 85, 88, 90, 95, 102], a single *final trust level (FTL)* is computed to reflect the general or global trust level of a service provider accumulated in a certain time period (e.g. in the latest 6 months). This *FTL* may be presumably taken as a prediction of trustworthiness for forthcoming transactions. Single trust value approaches are easily adopted in trust-oriented service comparison and selection. However, a single trust value cannot preserve the trust features well, e.g. whether and how the trust trend changes. Alternatively, a complete set of trust ratings can serve this purpose, but it is usually a large data set as it should cover a long service period. A good option is to compute a small data set to present a large set of trust ratings and preserve its trust features well.

In Chapter 3, we propose a trust vector with three values, including *final trust level (FTL)*, *service trust trend (STT)* and *service performance consistency level (SPCL)*, to depict a set of trust ratings. In addition to *FTL*, the *service trust trend* indicates whether the service trust ratings are becoming worse or better. *STT* is obtained from the slope of a regression line that best fits the set of ratings  $\{R^{(t_i)} | R^{(t_i)} \in [0, 1], t_i \in [t_1, t_n]\}$  distributed over a time interval  $[t_1, t_n]$ . The *service performance consistency level* indicates the extent to which the computed *STT* fits the given set of trust ratings.

A computed service trust vector proposed in Chapter 3 is meaningful only if the *SPCL* value is high, which indicates that all the ratings distributed in the time in-

terval are very close to the computed regression line. Only in such a situation can the *STT* represent the trend of service trust changes very well. Assuming there is a two-dimensional diagram with time  $t$  as the x-axis and rating value  $R$  as the y-axis,  $R^{(t_i)}$  represents the trust rating at  $t_i$ . Given a large set of trust ratings  $\{R^{(t_i)}\}$ , if the trust trend changes greatly in the whole time interval  $[t_{start}, t_{end}]$  ( $t_i \in [t_{start}, t_{end}]$ ),  $[t_{start}, t_{end}]$  should be divided into multiple time intervals, each of which corresponds to a subset of ratings that can be represented by one trust vector with a high *SPCL* value. Thus, as all the service trust vectors cross multiple time intervals from  $t_{start}$  to  $t_{end}$ , we term the trust vector computation process and the multiple time intervals (MTI) analysis the *horizontal aggregation of trust ratings*. This task requires efficient algorithms that can determine MTI. Meanwhile, the set of computed time intervals is expected to be the minimal.

Furthermore, we assume that there is a large amount of transactions for each service provider in the whole time interval  $[t_{start}, t_{end}]$ . Thus, if we process the ratings of all transactions occurring at different times separately, it will lead to too many trust vectors. Hence, in this chapter, we take  $t_i$  as a small time period (e.g. a day) and propose a *vertical rating aggregation* approach to aggregate all the ratings  $\{r_j^{(t_i)}\}$  for the services delivered during the same small time period  $t_i$ . We use  $R^{(t_i)}$  to denote the aggregated rating at  $t_i$ . As time  $t$  is the horizontal axis and all ratings  $\{r_j^{(t_i)}\}$  vertically distribute at the same small time period  $t_i$ , this computation process is termed the *vertical aggregation of trust ratings*.

With all the vertically aggregated ratings  $\{R^{(t_i)} | R^{(t_i)}$  is the vertical aggregation of all ratings at time  $t_i, t_i \in [t_{start}, t_{end}]\}$ , we then apply the horizontal aggregation of trust ratings and obtain multiple trust vectors. Consequently, with both vertical and horizontal rating aggregations, a small set of trust vectors can represent a large set of trust ratings for a long service transaction history while the trust features can be highly preserved.

In this chapter, our contributions can be briefly summarized as follows.

1. For the vertical aggregation of trust ratings, we propose a Gaussian distribution based analysis method and a clustering based analysis method. The former applies to cases where all ratings conform to the Gaussian distribution, whereas the latter applies if the condition does not hold. In addition, we also propose an approach to evaluate service rating reputation that can be incorporated with the above two methods to generate more objective results.
  
2. For the horizontal aggregation of trust ratings, we propose five multiple time interval (MTI) algorithms for generating multiple trust vectors (i.e. multiple time intervals) from a large set of trust ratings: boundary included greedy MTI algorithm, bisection-based boundary excluded greedy MTI algorithm, boundary included optimal MTI algorithm, boundary excluded optimal MTI algorithm and boundary mixed optimal MTI algorithm. The difference between the boundary excluded MTI algorithm and the boundary included MTI algorithm is that in the computed MTI, two adjacent time intervals can or can not have common boundaries. When we study the properties of our proposed algorithms both analytically and empirically, the following conclusions can be obtained.
  - (a) The boundary included greedy MTI algorithm is to include as many ratings as possible in one time interval under the condition of included boundary, and it may not return the minimal set of MTI.
  - (b) The bisection-based boundary excluded greedy MTI algorithm consumes much less CPU time than any other four MTI algorithms. However, it may not return the minimal set of MTI.
  - (c) The boundary included optimal MTI algorithm can return the minimal set of boundary included MTI.
  - (d) The boundary excluded optimal MTI algorithm can return the minimal set of boundary excluded MTI.
  - (e) The boundary mixed optimal MTI algorithm returns a minimal set of bound-

ary mixed MTI. This set is no larger than the set returned by any of the other four MTI algorithms.

(f) With any of our proposed algorithms, a small set of data can represent well a large set of trust ratings with well preserved trust features.

3. With our proposed vertical aggregation approach and horizontal aggregation approach, a small set of values can represent well a large set of trust ratings with well preserved trust features. This is significant for large-scale trust rating transmission and trust evaluation.

This chapter is organized as follows. Section 4.1 presents the vertical rating aggregation approach. In Section 4.2, two greedy and three optimal multiple time intervals analysis algorithms are proposed, and our analytical and empirical studies illustrate the effectiveness and efficiency of our proposed algorithms. Finally Section 4.3 concludes our work in this chapter.

## 4.1 Vertical Trust Rating Aggregation

In this section, we introduce our proposed vertical rating aggregation approach, which aggregates the ratings  $\{r_j^{(t_i)} | j = 1, \dots, m\}$  for the services delivered at a small time period  $t_i$  (e.g. a day) to a single trust value  $R^{(t_i)}$ .

### 4.1.1 Vertical Aggregation without Service Rating Reputation

In order to aggregate the ratings  $\{r_j^{(t_i)}\}$  vertically, we first consider the distribution of these ratings given by different clients. Let  $\bar{r}^{(t_i)}$  denote the rating that would ideally represent the trust level of the service delivered at the time period  $t_i$ . If most clients are honest, then their ratings are close to  $\bar{r}^{(t_i)}$ . Raters of these ratings are taken as belonging to the mainstream. Thus, each rating with a clear distance to  $\bar{r}^{(t_i)}$  is taken as marginal. As pointed out in cognitive science, in the process of decision-making,

cognitive and personal preference is usually needed to be taken into account [47]. One of the most commonly existing cognitive preferences is a willingness to believe what we have been told most often and by the greatest number of different sources [47]. Following this principle of cognitive science, marginal ratings can be identified and discarded. If we can determine the upper control limit  $R_u^{(t_i)}$  and the lower control limit  $R_l^{(t_i)}$  properly, marginal ratings out side of the range  $[R_l^{(t_i)}, R_u^{(t_i)}]$  can be filtered out. Then, an aggregated rating  $R^{(t_i)}$  can be derived from the remaining ratings in  $[R_l^{(t_i)}, R_u^{(t_i)}]$  and taken as the *estimation* of  $\bar{r}^{(t_i)}$ .

In this section, we propose a *Gaussian distribution based analysis method* and a *Clustering based analysis method* to compute the aggregated rating  $R^{(t_i)}$  in different situations.

#### 4.1.1.1 Gaussian distribution based analysis method

As illustrated in [35], if all service clients give feedback after transactions, the provided ratings conform to the Gaussian distribution. A complete set of honest ratings can be collected based on honest-feedback-incentive mechanisms [42, 43]. Therefore, the computation method introduced in this subsection applies to cases where the ratings given for the services delivered at  $t_i$  can approximately conform to the Gaussian distribution. In order to determine if ratings conform to the Gaussian distribution, we adopt the formal goodness-of-fit testing procedure based on the chi-square distribution [33]. If ratings do not conform to the Gaussian distribution, we will adopt our clustering based analysis method for vertical aggregation.

If ratings conform to the Gaussian distribution, according to the empirical rule in statistics [33], about 95.45% values in the Gaussian distribution are within  $[\mu - 2\sigma, \mu + 2\sigma]$  where  $\sigma$  is the standard deviation and  $\mu$  is the mean. Hence, based on the control chart in the statistical quality control [33], we can adopt  $\mu + 2\sigma$  as the upper control limit  $R_u^{(t_i)}$  and  $\mu - 2\sigma$  as the lower control limit  $R_l^{(t_i)}$ .

**Definition 13:** Based on unbiased estimation [33], the centerline  $R_c^{(t_i)}$ , the upper control limit  $R_u^{(t_i)}$  and the lower control limit  $R_l^{(t_i)}$  of trust ratings for the services delivered

at the small time period  $t_i$  are defined in sequence as follows:

$$R_c^{(t_i)} = \frac{1}{m} \sum_{j=1}^m r_j^{(t_i)}, \quad (4.1)$$

$$R_u^{(t_i)} = R_c^{(t_i)} + 2\sqrt{\frac{\sum_{j=1}^m (r_j^{(t_i)} - R_c^{(t_i)})^2}{m-1}}, \quad (4.2)$$

$$R_l^{(t_i)} = R_c^{(t_i)} - 2\sqrt{\frac{\sum_{j=1}^m (r_j^{(t_i)} - R_c^{(t_i)})^2}{m-1}}, \quad (4.3)$$

where  $r_j^{(t_i)} \in [0, 1]$  is the trust rating from client  $j$  ( $j = 1, \dots, m$ ) for a service delivered at time  $t_i$  ( $i = 1, \dots, n$ ).

The trust ratings out of the range  $[R_l^{(t_i)}, R_u^{(t_i)}]$  are therefore taken as marginal ratings.

**Definition 14:** If there are  $m'$  trust ratings  $\{r_k^{(t_i)'}\}$  in the range of  $[R_l^{(t_i)}, R_u^{(t_i)}]$ , the vertically aggregated rating  $R^{(t_i)}$  can be calculated by

$$R^{(t_i)} = \frac{1}{m'} \sum_{k=1}^{m'} r_k^{(t_i)'}. \quad (4.4)$$

#### 4.1.1.2 Clustering based analysis method

If ratings do not conform to the Gaussian distribution, the clustering based analysis method will be applied.

In this method, we adopt the hierarchical clustering method [31], which creates a hierarchical cluster of the given data set by either clustering from one cluster or  $n_r$  clusters ( $n_r$  is the size of the data set) until all clusters become stable. In order to determine marginal rating clusters, the *divisive hierarchical clustering approach* [31] is selected, which starts the decomposition from one cluster.

Before presenting the divisive hierarchical clustering approach, we first introduce some definitions.

**Definition 15:** The *relative rating density* from  $r_h^{(t_i)}$  to  $r_j^{(t_i)}$  at the small time period  $t_i$  is

$$D_r^{(t_i)}(r_h^{(t_i)}, r_j^{(t_i)}) = \frac{\sum_{r_h^{(t_i)} < r_k^{(t_i)} < r_j^{(t_i)}} fre(r_k^{(t_i)})}{r_j^{(t_i)} - r_h^{(t_i)}}, \quad (4.5)$$

where  $r_h^{(t_i)}$  and  $r_j^{(t_i)}$  ( $r_h^{(t_i)} < r_j^{(t_i)}$ ) are ratings, and  $fre(r_k^{(t_i)})$  is the frequency of  $r_k^{(t_i)}$ .

**Definition 16:** The *marginal rating percentage* at the small time period  $t_i$  is

$$P_{marginal}^{(t_i)} = \frac{n_{marginal}^{(t_i)}}{n_{total}^{(t_i)}}, \quad (4.6)$$

where  $n_{marginal}^{(t_i)}$  is the number of marginal ratings for the services delivered at the small time period  $t_i$  and  $n_{total}^{(t_i)}$  is the total number of ratings for the services delivered at the small time period  $t_i$ .

The divisive hierarchical clustering approach works as follows. Initially, all the ratings are placed in one cluster, and the centerline  $R_c^{(t_i)}$  is calculated with all ratings by Eq. (4.1). Then the cluster is split according to relative rating density in the cluster. This process repeats until  $D_r^{(t_i)}$  or  $P_{marginal}^{(t_i)}$  reaches the corresponding threshold (such as  $D_r^{(t_i)} = 0.10$  and  $P_{marginal}^{(t_i)} = 0.10$ ). All ratings that are not in the centered cluster are taken as marginal ratings. In the centered cluster, the two boundary ratings are taken as  $R_l^{(t_i)}$  and  $R_u^{(t_i)}$ . With all ratings in  $[R_l^{(t_i)}, R_u^{(t_i)}]$ , the vertically aggregated rating  $R^{(t_i)}$  can be computed according to Eq. (4.4).

### 4.1.2 Vertical Aggregation with Service Rating Reputation (SRR)

The client's rating reputation is important for estimating the ideal rating  $\bar{r}^{(t_i)}$ . This reputation can be evaluated with the distance from the client's rating to  $R^{(t_i)}$ . Obviously, the smaller the distance, the better the service rating reputation (SRR). With the SRR of all ratings,  $R^{(t_i)}$  can be recalculated. This leads to an iterative process until all computed values become stable.

#### 4.1.2.1 Service Rating Reputation (SRR) Evaluation

The calculation of *SRR* follows Principle 1 in Section 3.1.1, which appears in a number of studies [51, 59, 92, 102].

**Definition 17:**  $V_{SRR_j}^{(t_i)}$ , the *SRR* value for client  $j$  from  $t_1$  to  $t_i$ , can be calculated as follows:

$$V_{SRR_j}^{(t_i)} = \frac{\sum_{k=1}^i w_{t_k} \cdot R_{SRR_j}^{(t_k)}}{\sum_{k=1}^i w_{t_k}}, \quad (4.7)$$

where  $R_{SRR_j}^{(t_k)}$  is the *SRR* value for client  $j$  at the small time period  $t_k$  ( $k = 1, \dots, i$ ), and  $w_{t_k}$  is the weight for  $R_{SRR_j}^{(t_k)}$ , which can be calculated as the exponential moving average [33] and defined in Eq. (3.2).

In addition, because the smaller the distance between a rating for a client and the estimation of  $\bar{r}^{(t_i)}$ , the bigger and better the *SRR* and vice versa, a principle about the  $R_{SRR_j}^{(t_i)}$  evaluation is introduced as follows.

**Principle 3:**  $R_{SRR_j}^{(t_i)}$ , the *SRR* value for client  $j$  at the small time period  $t_i$  ( $i = 1, \dots, n$ ), is a monotonically decreasing function of the distance

$$r_{jdis}^{(t_i)} = |r_j^{(t_i)} - R_c^{(t_i)}|, \quad (4.8)$$

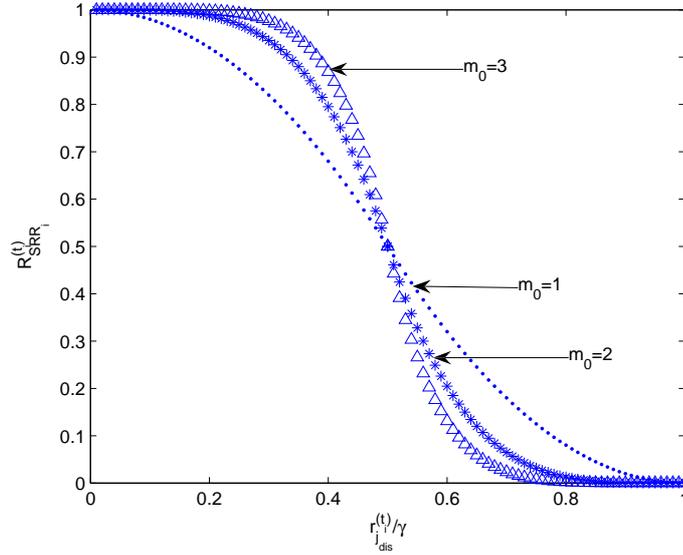
where  $r_j^{(t_i)}$  is the trust rating from client  $j$  for the service delivered at the small time period  $t_i$  and  $R_c^{(t_i)}$  is the weighted average of  $r_j^{(t_i)}$ .

According to Principle 3,  $R_{SRR_j}^{(t_i)}$  can be calculated by the following formula.

**Definition 18:**  $R_{SRR_j}^{(t_i)}$ , the *SRR* value for client  $j$  at the small time period  $t_i$  ( $i = 1, \dots, n$ ), can be evaluated as

$$R_{SRR_j}^{(t_i)} = \begin{cases} 1 - 2^{2m_0-1} \left( \frac{r_{jdis}^{(t_i)}}{\gamma} \right)^{2m_0} & \text{if } 0 \leq r_{jdis}^{(t_i)} \leq \frac{\gamma}{2} \\ 2^{2m_0-1} \left( \frac{r_{jdis}^{(t_i)}}{\gamma} - 1 \right)^{2m_0} & \text{if } \frac{\gamma}{2} < r_{jdis}^{(t_i)} \leq \gamma \end{cases}, \quad (4.9)$$

where  $\gamma = \max r_{jdis}^{(t_i)}$ , and  $m_0$  is the argument to control the function curve.



**Figure 4.1:**  $R_{SRR_j}^{(t_i)}$  function

When setting  $m_0 = 1, 2$  or  $3$ , the changes of the function curve in Eq. (4.9) are depicted in Fig. 4.1. It is easy to see that in all cases  $R_{SRR_j}^{(t_i)}$  is the monotonically decreasing function of  $r_{dis}^{(t_i)}$ , which follows Principle 3.

#### 4.1.2.2 Vertical Aggregation of Trust Ratings

With  $SRR$  taken into account, we should refine the Gaussian distribution based analysis method and the clustering based analysis method.

**Definition 19:** Based on the weighted unbiased estimation [33], the centerline  $R_c^{(t_i)'}$ , the upper control limit  $R_u^{(t_i)'}$  and the lower control limit  $R_l^{(t_i)'}$  can be calculated in sequence as follows:

$$R_c^{(t_i)'} = \frac{\sum_{j=1}^n V_{SRR_j}^{(t_i)} \cdot r_j^{(t_i)}}{\sum_{j=1}^n V_{SRR_j}^{(t_i)}}, \quad (4.10)$$

$$R_u^{(t_i)'} = R_c^{(t_i)'} + 2\sqrt{\frac{\sum_{j=1}^n V_{SRR_j}^{(t_i)} \cdot (r_j^{(t_i)} - R_c^{(t_i)'})^2}{\sum_{j=1}^n V_{SRR_j}^{(t_i)} - 1}}, \quad (4.11)$$

$$R_l^{(t_i)'} = R_c^{(t_i)'} - 2\sqrt{\frac{\sum_{j=1}^n V_{SRR_j}^{(t_i)} \cdot (r_j^{(t_i)} - R_c^{(t_i)'})^2}{\sum_{j=1}^n V_{SRR_j}^{(t_i)} - 1}}, \quad (4.12)$$

where  $r_j^{(t_i)}$  is the trust rating from client  $j$  for the service delivered at the small time period  $t_i$  ( $i = 1, \dots, n$ ), and  $V_{SRR_j}^{(t_i)}$  is the service rating reputation (*SRR*) value for client  $j$  defined in Eq. (4.7).

**Definition 20:** If there are  $m''$  trust ratings  $\{r_k^{(t_i)''}\}$  in the range of  $[R_l^{(t_i)'}, R_u^{(t_i)'}]$  and  $R_{SRR_k}^{(t_i)} > \epsilon_{SRR}$ , the vertically aggregated rating  $R^{(t_i)'}$  can be defined as

$$R^{(t_i)'} = \frac{\sum_{k=1}^{m''} V_{SRR_k}^{(t_i)} \cdot r_k^{(t_i)''}}{\sum_{k=1}^{m''} V_{SRR_k}^{(t_i)}}, \quad (4.13)$$

where  $R_u^{(t_i)'}$  is the upper control limit defined in Eq. (4.11),  $R_l^{(t_i)'}$  is the lower control limit defined in Eq. (4.12) and  $\epsilon_{SRR}$  ( $\epsilon_{SRR} \in [0, 1]$ ) is a threshold.

As for the clustering based analysis method, all the processes are the same as the method introduced in Section 4.1.1.2 except that *SRR* is added in computation.

In order to evaluate the *SRR*, the centerline  $R_c^{(t_i)'}$  should be known. However, from Eq. (4.10),  $V_{SRR_j}^{(t_i)}$  is necessary for determining  $R_c^{(t_i)'}$ . Therefore, it is an iterative process to compute  $R_c^{(t_i)'}$  and  $V_{SRR_j}^{(t_i)}$ .

Here we take the Gaussian distribution based analysis method as an example to illustrate the iterative vertical rating aggregation process in Algorithm 1. Obviously, the iterative process is similar in the vertical rating aggregation on top of the clustering based analysis method.

In summary, the main difference between the vertically aggregated rating evaluation introduced in Section 4.1.1 and the one introduced in this subsection is that the latter approach takes *SRR* into account, which leads to more objective results.

**Algorithm 1** Vertical rating aggregation algorithm

---

**Input:** trust ratings  $r_j^{(t_i)}$ ,  
an arbitrary small positive number  $\epsilon_0$  (such as 0.0001).  
**Output:**  $V_{SRR_j}^{(t_i)}$ ,  $R^{(t_i)'}$ .

- 1: Initialize  $R_{SRR_j}^{(t_i)}$  by Eq. (4.9);
- 2:  $r_{i_{a_1}} \Leftarrow 0$ ;
- 3:  $r_{i_{a_2}} \Leftarrow 0$ ;
- 4: **while**  $\max |r_{i_{a_1}} - r_{i_{a_2}}| > \epsilon_0$  **do**
- 5:    $r_{i_{a_2}} \Leftarrow r_{i_{a_1}}$ ;
- 6:   compute  $V_{SRR_j}^{(t_i)}$  with  $R_{SRR_j}^{(t_i)}$  by Eq. (4.7);
- 7:   compute  $r_{i_{a_1}}$  by Eq. (4.10);
- 8:   compute  $R_{SRR_j}^{(t_i)}$  by Eq. (4.9);
- 9: **end while**
- 10:  $R_c^{(t_i)'}$   $\Leftarrow r_{i_{a_1}}$ ;
- 11: compute  $R_u^{(t_i)'}$  by Eq. (4.11);
- 12: compute  $R_l^{(t_i)'}$  by Eq. (4.12);
- 13: compute  $R^{(t_i)'}$  by Eq. (4.13);
- 14: **return**  $V_{SRR_j}^{(t_i)}$ ,  $R^{(t_i)'}$ ;

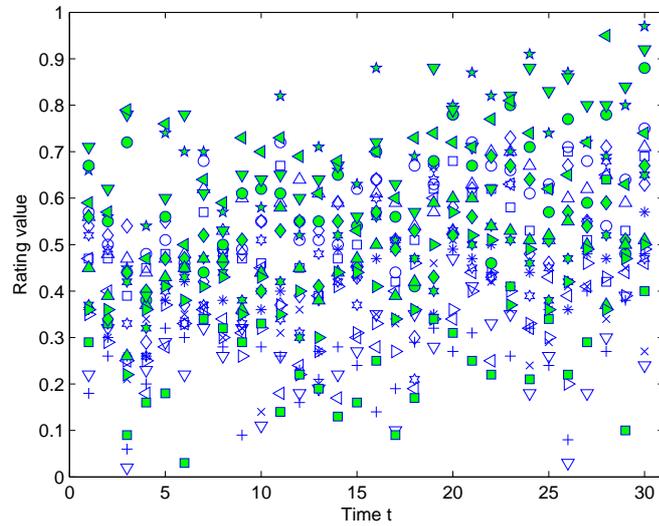
---

### 4.1.3 Experiment on Vertical Rating Aggregation

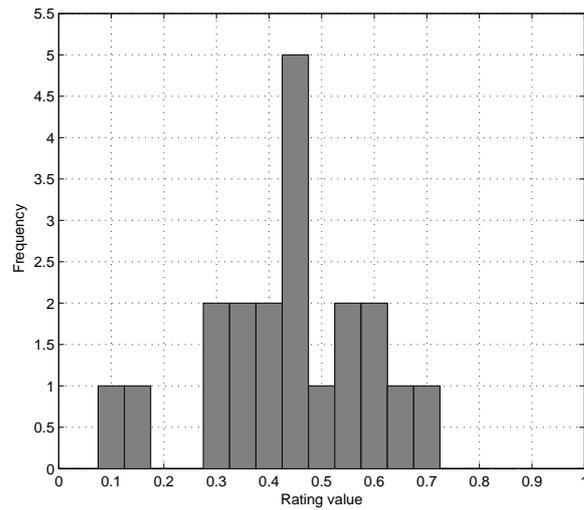
In this experiment, we introduce an example to illustrate our vertical rating aggregation approach. There are 20 service providers who obtain trust ratings for services delivered in the time interval  $[t_1, t_{30}]$ . The trust ratings  $\{r_j^{(t_i)} | r_j^{(t_i)} \in [0, 1]\}$  for the service delivered at time  $t_i$  from provider  $j$  for 20 service providers are plotted in Fig. 4.2.

**Case 1:** In this case, we apply the vertical rating aggregation approach introduced in Section 4.1.1 without considering any service rating reputation (*SRR*). For each rating set, we first check if all the ratings conform to the Gaussian distribution hypothesis by applying the goodness-of-fit test procedure [33]. If the Gaussian distribution hypothesis holds,  $R_c^{(t_i)}$ ,  $R_u^{(t_i)}$  and  $R_l^{(t_i)}$  are computed with Eqs. (4.1), (4.2) and (4.3) respectively. If the Gaussian distribution hypothesis does not hold, then the clustering based analysis method is applied.

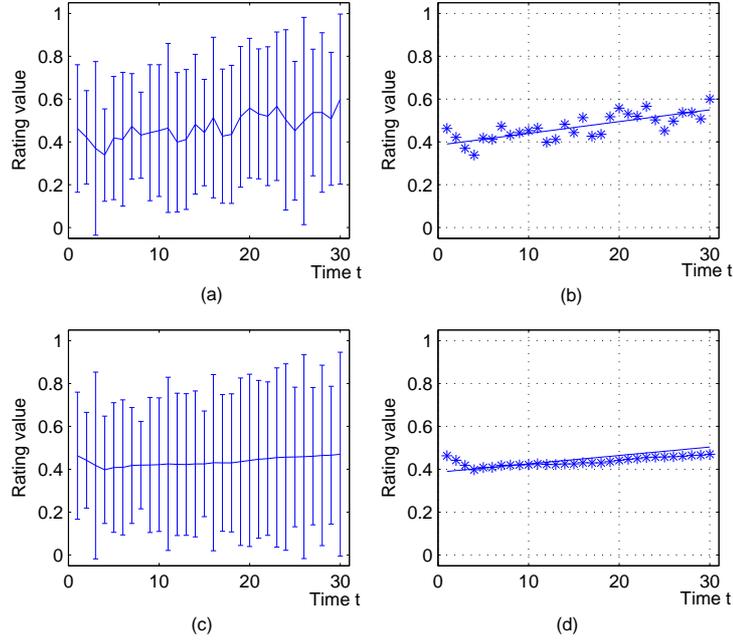
Taking the rating set at the small time period  $t_{10}$  (see Fig. 4.2) as an example, the



**Figure 4.2:** Service rating values in Experiment on Vertical Rating Aggregation



**Figure 4.3:** Rating frequency at  $t_{10}$  in Experiment on Vertical Rating Aggregation



**Figure 4.4:** Error bars and *STT* in Experiment on Vertical Rating Aggregation

corresponding histogram of ratings and frequency is plotted in Fig. 4.3. With the relative rating density threshold  $\epsilon_{RRD} = 0.06$ , the centered cluster is  $[0.25, 0.75]$ , whose  $P_{marginal}^{(t_{10})}$  is no more than the marginal rating percentage threshold  $\epsilon_{MRP} = 0.10$ . Based on this cluster, we have  $R^{(t_{10})} = 0.4526$ . Thus, the ratings out of  $[0.25, 0.75]$  are identified as marginal ratings.

In Fig. 4.4, there are 30 small time periods (i.e.  $\forall t_i, 1 \leq i \leq 30$ ). For each rating set  $\{r_j^{(t_i)}\}$ ,  $R_c^{(t_i)}$ ,  $R_u^{(t_i)}$  and  $R_l^{(t_i)}$  are computed by using either the Gaussian distribution based analysis method or the clustering based analysis method. The obtained values are plotted by the error bars in Fig. 4.4(a).

By applying the *STT* & *SPCL* evaluations, the trust vector is computed as listed in Table 4.1. The corresponding regression line is plotted in Fig. 4.4(b) together with 30 vertically aggregated ratings.

**Case 2:** In this case, we study vertical rating aggregation with service rating reputa-

**Table 4.1:** Trust vectors in Experiment on Vertical Rating Aggregation

$R^{(t_i)}$	$T_{FTL}$	$T_{STT}$	$T_{SPCL}$
<b>Case 1</b> (without <i>SRR</i> )	0.4906	0.0055	0.9344
<b>Case 2</b> (with <i>SRR</i> )	0.4617	0.0039	0.9697

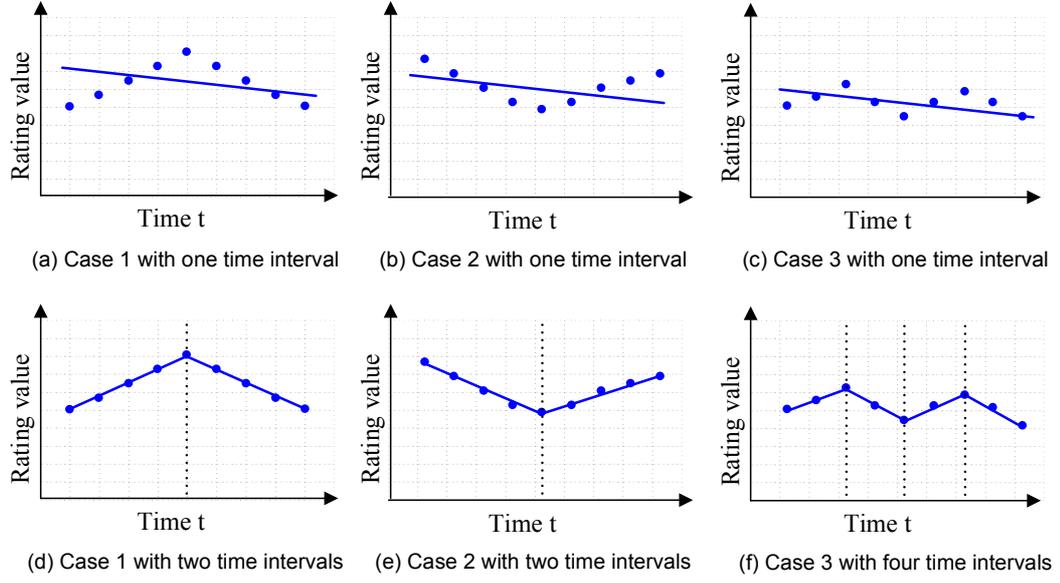
tion (*SRR*) using the threshold  $\epsilon_{MRP} = 0.10$ . The computed  $R_c^{(t_i)}$ ,  $R_u^{(t_i)}$  and  $R_l^{(t_i)}$  are plotted by the error bars in Fig. 4.4(c).

By applying the *STT* & *SPCL* evaluations with the service rating reputation threshold  $\epsilon_{SRR} = 0.5$ , the trust vector is computed as listed in Table 4.1. The corresponding regression line is plotted in Fig. 4.4(d) together with 30 vertically aggregated ratings.

Compared with the curve of the vertically aggregated ratings  $\{R^{(t_i)}\}$  in Case 1 as plotted in Fig. 4.4(a), the curve of  $\{R^{(t_i)}\}$  in Case 2 as plotted in Fig. 4.4(c) is smoother. This is because *SRR* is the accumulated rating reputation and it glues the ratings in two adjacent small time periods (i.e.  $t_i$  and  $t_{i+1}$ ). Therefore,  $T_{SPCL}$  in Case 2 is greater than that in Case 1 (see Table 4.1), indicating that the trust vector in Case 2 can represent the ratings better.

## 4.2 Multiple Time Intervals (MTI) Analysis

A single trust vector with three values can represent the ratings in a given time interval  $[t_1, t_n]$  well if its *SPCL* value is high (i.e. 0.9 or more). However, when the trust trend significantly changes in  $[t_1, t_n]$ , though a single trust vector can be computed, the *SPCL* value will be low, indicating that the obtained trust vector or regression line can not represent all ratings precisely. In such a case, in order to represent all trust ratings well, multiple intervals in  $[t_1, t_n]$  should be determined, within each of which one trust vector with a high *SPCL* value can be obtained to represent the corresponding ratings well.

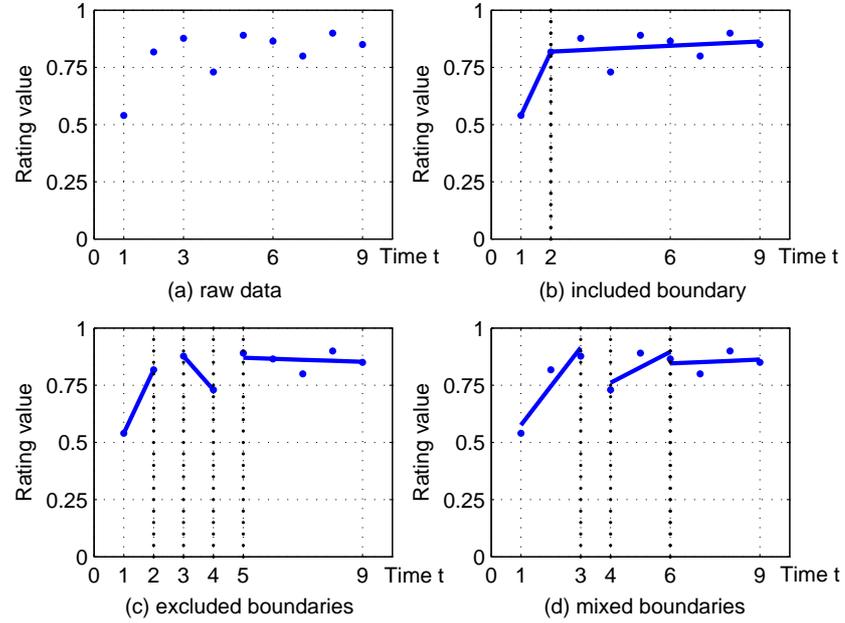


**Figure 4.5:** Several MTI examples

For example, in Fig. 4.5, we can notice that all three cases are quite different from each other in terms of trust trend changes. If only one trust vector is computed in each case, all three cases have approximately the same  $T_{FTL}$ ,  $T_{STT}$  and  $T_{SPCL}$ . However, in each case, most points have clear distances to the obtained regression line. This leads to a low  $SPCL$  value, indicating that the obtained single trust vector (or regression line) cannot represent all trust ratings well. Instead, in each case, the whole time interval can be divided into multiple time intervals (i.e. 2 time intervals in Fig. 4.5(d) & (e), and 4 time intervals in Fig. 4.5(f)). In each sub-time interval, one trust vector (or a regression line) with a high  $SPCL$  value can represent all corresponding ratings well.

### 4.2.1 Boundaries of MTI

In order to determine multiple time intervals, we first need to study the boundaries of time intervals. For example, in Fig. 4.6(b),  $t = 1$  and  $t = 2$  are the time interval boundaries  $[1, 2]$ . In multiple time intervals analysis, there are two types of boundaries as follows.



**Figure 4.6:** MTI examples with different boundaries

**Included boundary:** Two adjacent time intervals have the same boundaries. For example, in Fig. 4.6(b), boundary  $t = 2$  is *included* in both time interval  $[1, 2]$  and time interval  $[2, 9]$ .

**Excluded boundary:** The boundary of a time interval is *excluded* from adjacent time intervals. For example, in Fig. 4.6(c), boundary  $t = 2$  of time interval  $[1, 2]$  is excluded from adjacent time interval  $[3, 4]$ , and boundary  $t = 3$  of time interval  $[3, 4]$  is also excluded from the adjacent time interval  $[1, 2]$ .

With these two types of boundaries, we can have three kinds of MTI algorithms as follows, which determine the multiple time intervals of a given set of ratings.

**Boundary included MTI algorithm:** Adjacent time intervals determined by such an algorithm have a common boundary, i.e. the included boundary (see Fig. 4.6(b)).

**Boundary excluded MTI algorithm:** Adjacent time intervals computed by such an algorithm have no common boundary, i.e. boundaries of MTI are excluded (see

Fig. 4.6(c)).

**Boundary mixed MTI algorithm:** Both included boundaries and excluded boundaries may appear in the adjacent time intervals computed by such an algorithm (see Fig. 4.6(d)).

For example, there are some ratings plotted in Fig. 4.6(a). With a boundary included MTI algorithm, we can assume to have 2 time intervals  $[1, 2]$  and  $[2, 9]$  with an included boundary, which are depicted in Fig. 4.6(b). In contrast, with a boundary excluded MTI algorithm, we can have 3 time intervals  $[1, 2]$ ,  $[3, 4]$  and  $[5, 9]$  with excluded boundaries, which are depicted in Fig. 4.6(c). However, with a boundary mixed MTI algorithm, we can have 3 time intervals  $[1, 3]$ ,  $[4, 6]$  and  $[6, 9]$ , which are depicted in Fig. 4.6(d). Here  $t = 3$  and  $t = 4$  are excluded boundaries while  $t = 6$  is an included boundary.

In this section, we first propose a boundary included greedy MTI algorithm, a bisection-based boundary excluded greedy MTI algorithm, a boundary included optimal MTI algorithm and a boundary excluded optimal MTI algorithm for MTI analysis. Then we further develop a boundary mixed optimal MTI algorithm that can return a minimal set of MTI, which is no larger than the set returned by any of the other four algorithms.

### 4.2.2 Boundary Included Greedy MTI Algorithm

By including as many ratings as possible in one time interval under the condition of included boundary, the boundary included greedy MTI algorithm (Algorithm 2) works as follows.

**Step 1:** Take  $(t_1, R^{(t_1)})$  as the starting point and  $(t_n, R^{(t_n)})$  as the initial ending point (lines 1–3 in Algorithm 2).

- a) If  $T_{SPCL}^{[t_1, t_n]} \geq \epsilon_{MTI}$ , the regression line starts from  $(t_1, R^{(t_1)})$  and ends at  $(t_n, R^{(t_n)})$  (line 4);

b) otherwise,  $(t_{n-1}, R^{(t_{n-1})})$  is taken as the ending point for checking if  $T_{SPCL}^{[t_1, t_{n-1}]} \geq \epsilon_{MTI}$  and so forth. Thus, the first ending point  $(t_i, R^{(t_i)})$  ( $i \in [2, n]$ ) can be determined so that  $T_{SPCL}^{[t_1, t_i]} \geq \epsilon_{MTI}$ . By doing so, the obtained regression line is the longest one starting from  $(t_1, R^{(t_1)})$  (lines 5–7).

**Step 2:** Take  $(t_i, R^{(t_i)})$  as the new starting point and  $(t_n, R^{(t_n)})$  as the ending point.

Repeat the process introduced in Step 1 so that a regression line can be drawn from  $(t_i, R^{(t_i)})$  to  $(t_j, R^{(t_j)})$  ( $j \in [i+1, n]$ ) satisfying  $T_{SPCL}^{[t_i, t_j]} \geq \epsilon_{MTI}$  (lines 4–13).

**Step 3:** Repeat Step 2 until the last regression line reaches point  $(t_n, R^{(t_n)})$ .

---

#### Algorithm 2 Boundary Included Greedy MTI algorithm

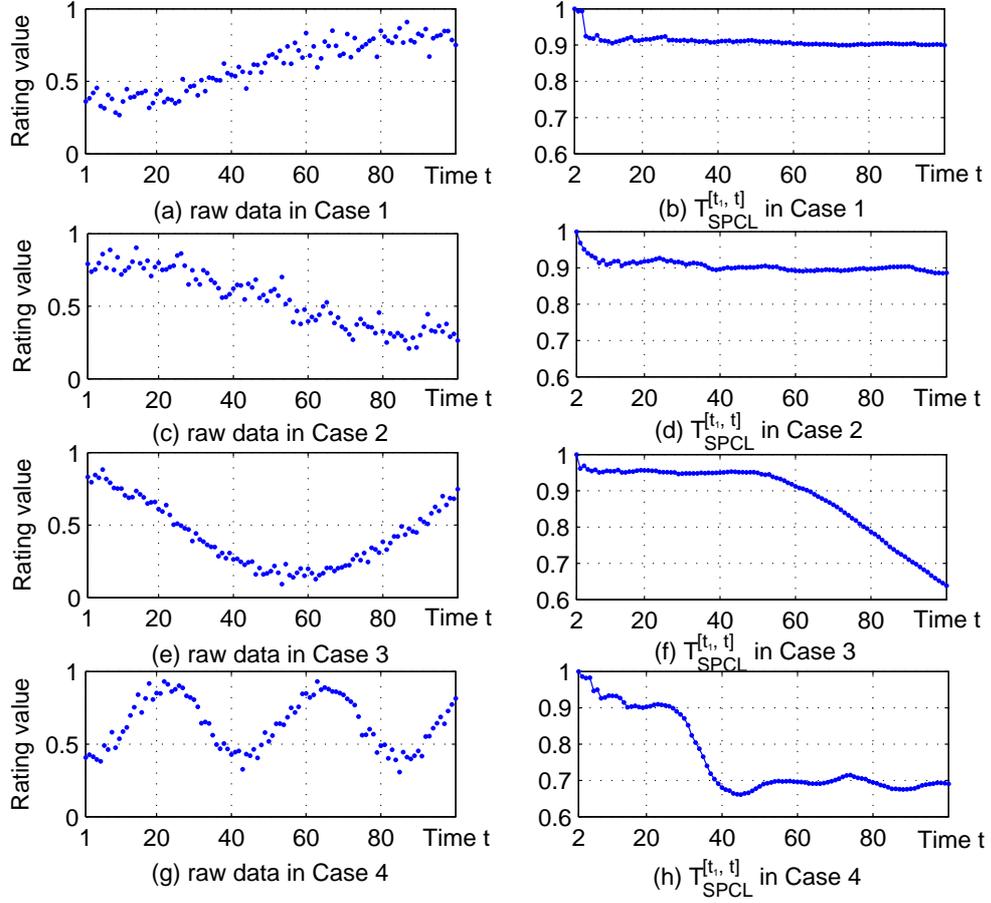
---

**Input:** trust ratings  $R^{(t_i)}$ ,  
the given time interval  $[t_1, t_n]$ ,  
the threshold  $\epsilon_{MTI}$  of  $T_{SPCL}$  (such as 0.9).  
**Output:** the boundary set  $\{[t_{lb_j}, t_{rb_j}]\}$  of MTI.

- 1:  $j \leftarrow 1$ ;
- 2: left time boundary  $t_{lb_j} \leftarrow t_1$ ;
- 3: right time boundary  $t_{rb_j} \leftarrow t_n$ ;
- 4: **while**  $T_{SPCL}^{[t_{lb_j}, t_n]} < \epsilon_{MTI}$  **do**
- 5:     **while**  $T_{SPCL}^{[t_{lb_j}, t_{rb_j}]} < \epsilon_{MTI}$  **do**
- 6:          $t_{rb_j} \leftarrow t_{rb_{j-1}}$ ;
- 7:     **end while**
- 8:     **if**  $t_{rb_j} \neq t_n$  **then**
- 9:          $j \leftarrow j + 1$ ;
- 10:          $t_{lb_j} \leftarrow t_{rb_{j-1}}$ ;
- 11:          $t_{rb_j} \leftarrow t_n$ ;
- 12:     **end if**
- 13: **end while**
- 14: **return**  $\{[t_{lb_j}, t_{rb_j}]\}$ ;

---

The computation of  $T_{SPCL}$  incurs a complexity of  $O(n)$  (line 5), where  $n$  is the number of data points, i.e.  $n = |\{(t_i, R^{(t_i)})\}|$ . As it is contained in the nested loops ( $O(n^2)$ ) (lines 4–13), the boundary included greedy MTI algorithm incurs a complexity of  $O(n^3)$ .



**Figure 4.7:** The properties of SPCL function  $T_{SPCL}^{[t_1, t]}$

### 4.2.3 Bisection-based Boundary Excluded Greedy MTI Algorithm

Take  $(t_1, R^{(t_1)})$  as the starting point and  $(t_n, R^{(t_n)})$  as the ending point. If  $T_{SPCL}^{[t_1, t_n]} \geq \epsilon_{MTI}$ , the regression line starts from  $(t_1, R^{(t_1)})$  and ends at  $(t_n, R^{(t_n)})$ ; otherwise, we need an MTI algorithm which can return a set of MTI.

Now let us focus on the function  $T_{SPCL}^{[t_1, t]}$  in Eq. (3.15), analyze its properties and determine the MTI. We introduce some theorems below to generalize these properties, which are also depicted in Fig. 4.7.

**Lemma 1:**  $\forall i \in [1, n - 1]$ ,

$$T_{SPCL}^{[t_i, t_{i+1}]} = 1. \quad (4.14)$$

**Proof:** As there are only two data points  $(t_i, R^{(t_i)})$  and  $(t_{i+1}, R^{(t_{i+1})})$  in time interval  $[t_i, t_{i+1}]$ , both of them lie on the corresponding regression line from  $(t_i, R^{(t_i)})$  to  $(t_{i+1}, R^{(t_{i+1})})$ . The straight line from  $(t_i, R^{(t_i)})$  to  $(t_{i+1}, R^{(t_{i+1})})$  is the regression line. Hence, following Eq. (3.15), we have  $T_{SPCL}^{[t_i, t_{i+1}]} = 1$ .  $\square$

When  $i = 1$ , following Lemma 1, we have  $T_{SPCL}^{[t_1, t_2]} = 1$ . This confirms the fact that all the  $T_{SPCL}^{[t_1, t]}$  functions ( $t \geq t_2$ ) in Fig. 4.7 (b)(d)(f)(h) start from 1.

**Theorem 1:** With  $1 \leq i < j < k \leq n$ ,  $\forall \epsilon_{MTI} \in (0, 1)$ , if

$$\left(T_{SPCL}^{[t_i, t_j]} - \epsilon_{MTI}\right) \left(T_{SPCL}^{[t_i, t_k]} - \epsilon_{MTI}\right) < 0, \quad (4.15)$$

then  $T_{SPCL}^{[t_i, t]} = \epsilon_{MTI}$  has at least one root in time interval  $[t_j, t_k]$ .

**Proof:** As  $T_{SPCL}^{[t_i, t]}$  is a continuous function of variable  $t$ , according to the *intermediate value theorem* in mathematical analysis [75], the condition in Eq. (4.15) implies that  $T_{SPCL}^{[t_i, t]} = \epsilon_{MTI}$  has at least one root in the interval  $[t_j, t_k]$ .  $\square$

Take the function  $T_{SPCL}^{[t_1, t]}$  in Fig. 4.7(f) as an example. With  $\epsilon_{MTI} = 0.9$ , we have  $T_{SPCL}^{[t_1, t_{40}]} > 0.9$  and  $T_{SPCL}^{[t_1, t_{80}]} < 0.9$ , i.e.  $(T_{SPCL}^{[t_1, t_{40}]} - 0.9)(T_{SPCL}^{[t_1, t_{80}]} - 0.9) < 0$ . In addition, we can observe that  $T_{SPCL}^{[t_1, t]} = 0.9$  has a root in time interval  $[40, 80]$ , which confirms Theorem 1 empirically.

**Theorem 2:** With  $1 \leq i < j \leq n$ ,  $\forall \epsilon_{MTI} \in (0, 1)$ , if  $T_{SPCL}^{[t_i, t_j]} < \epsilon_{MTI}$ , then  $T_{SPCL}^{[t_i, t]} = \epsilon_{MTI}$  has at least one root in time interval  $[t_{i+1}, t_j]$ .

**Proof:** Following Lemma 1, we know  $T_{SPCL}^{[t_i, t_{i+1}]} = 1 > \epsilon_{MTI}$ . In addition, with  $T_{SPCL}^{[t_i, t_j]} < \epsilon_{MTI}$ , we have  $(T_{SPCL}^{[t_i, t_{i+1}]} - \epsilon_{MTI})(T_{SPCL}^{[t_i, t_j]} - \epsilon_{MTI}) < 0$ . Then, following Theorem 1, we can know that  $T_{SPCL}^{[t_i, t]} = \epsilon_{MTI}$  has at least one root in time interval  $[t_{i+1}, t_j]$ .  $\square$

In Fig. 4.7, with  $\epsilon_{MTI} = 0.9$ , for each case we have  $T_{SPCL}^{[t_1, t]} < 0.9$ . In addition, we can observe that for each case,  $T_{SPCL}^{[t_1, t]} = 0.9$  has at least one root in time interval  $[t_1, t_{100}]$ . This confirms Theorem 2 empirically.

Now let us introduce our bisection-based boundary excluded greedy MTI algorithm in detail. Let  $t_{lb_i}$  denote the left boundary of the  $i$ th time interval, and let  $t_{rb_i}$

denote the right boundary of the  $i$ th time interval. In this algorithm, in order to determine the first time interval  $[t_{lb_1}, t_{rb_1}]$  ( $t_{lb_1} = t_1$ ), we need to find the maximal right time boundary  $t_{rb_1}$  satisfying  $T_{SPCL}^{[t_1, t_{rb_1}]} \geq \epsilon_{MTI}$ . If the root  $t^*$  of  $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$  can be obtained, we round down  $t^*$  and let  $t_{rb_1} = \lfloor t^* \rfloor$  (e.g. if  $t^* = 63.25$ , then  $t_{rb_1} = \lfloor 63.25 \rfloor = 63$ ). Then set  $t_{lb_2} = t_{rb_1+1}$  (i.e. if  $t_{rb_1} = 63$ , then  $t_{lb_2} = 64$ ), and repeat the above process until the last time interval reaches  $t_n$ . Thus, all MTI can be determined.

Now the task is to find the root of equation  $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$ . Our method is to repeatedly bisect the time interval that contains a root of  $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$ , until a subinterval can be selected which is smaller than 1. Let us explain this bisection process in detail with an example depicted in Fig. 4.7(f). With  $\epsilon_{MTI} = 0.9$ , as  $T_{SPCL}^{[t_1, t_{100}]} < 0.9$ , according to Theorem 2,  $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$  has a root in time interval  $[t_2, t_{100}]$ . By bisecting time interval  $[t_2, t_{100}]$ , with midpoint  $t_{51}$ , we have  $T_{SPCL}^{[t_1, t_{51}]} > 0.9$ . Thus, according to Theorem 1,  $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$  has a root in time interval  $[t_{51}, t_{100}]$ . By bisecting time interval  $[t_{51}, t_{100}]$  at midpoint  $t_{75.5}$ , we can obtain  $T_{SPCL}^{[t_1, t_{75.5}]} < 0.9$ . According to Theorem 1,  $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$  has a root in time interval  $[t_{51}, t_{75.5}]$ . We repeatedly bisect the time interval containing a root of  $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$ , until we obtain that  $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$  has a root in  $[t_{63.25}, t_{64.0156}]$ , where  $64.0156 - 63.25 < 1$ . Hence, the right boundary of the first time interval can be determined as  $t_{rb_1} = \lfloor t_{63.25} \rfloor = t_{63}$ . Now with the left boundary of the second time interval  $t_{lb_2} = t_{64}$ , we can repeat the above process to determine the right boundary  $t_{rb_2}$  for the second regression line. The whole process terminates when  $t_n$  is determined as the right boundary of a regression line (the last one).

The bisection-based boundary excluded greedy MTI algorithm (Algorithm 3) works as follows.

**Step 1:** Take  $(t_1, R^{(t_1)})$  as the starting point and  $(t_n, R^{(t_n)})$  as the initial ending point (lines 1-3 in Algorithm 3).

- a) If  $T_{SPCL}^{[t_1, t_n]} \geq \epsilon_{MTI}$ , the regression line starts from  $(t_1, R^{(t_1)})$  and ends at  $(t_n, R^{(t_n)})$  (line 4);
- b) otherwise, following Theorem 2, function  $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$  has a root in time

**Algorithm 3** Bisection-based boundary excluded greedy MTI algorithm

---

**Input:** trust ratings  $R^{(t_i)}$ ,  
the threshold  $\epsilon_{MTI}$  of  $T_{SPCL}$  (such as 0.9),  
the given time interval  $[t_1, t_n]$ .

**Output:** the boundary set  $t_b$  of MTI.

- 1:  $j \leftarrow 1$ ;
- 2: left time boundary  $t_{lb_j} \leftarrow t_1$ ;
- 3: right time boundary  $t_{rb_j} \leftarrow t_n$ ;
- 4: **while**  $T_{SPCL}^{[t_{lb_j}, t_n]} < \epsilon_{MTI}$  **do**
- 5:    $t_{left} \leftarrow t_{lb_j}$ ;
- 6:    $t_{right} \leftarrow t_n$ ;
- 7:   **while**  $t_{right} - t_{left} > 1$  **do**
- 8:      $t_{mid} = \frac{t_{left} + t_{right}}{2}$ ;
- 9:     **if**  $T_{SPCL}^{[t_{lb_j}, t_{mid}]} == \lambda$  **then**
- 10:        $t_{rb_j} \leftarrow t_{mid}$ ;
- 11:     **else if**  $T_{SPCL}^{[t_{lb_j}, t_{mid}]} > \lambda$  **then**
- 12:        $t_{left} \leftarrow t_{mid}$ ;
- 13:     **else**
- 14:        $t_{right} \leftarrow t_{mid}$ ;
- 15:     **end if**
- 16:   **end while**
- 17:   find  $t_j^* < t_{left} < t_{right} < t_j^* + 1$ , then let  $t_j^* \leftarrow \lfloor t_{left} \rfloor$  and  $t_{rb_j} \leftarrow t_j^*$ ;
- 18:    $j \leftarrow j + 1$ ;
- 19:    $t_{lb_j} \leftarrow t_{rb_{j-1}+1}$ ;
- 20: **end while**
- 21: **return**  $t_b \leftarrow [t_{lb}^T \ t_{rb}^T]^T$ ;

---

interval  $[t_2, t_n]$ . We initialize the left boundary  $t_{left} \leftarrow t_1$  and the right boundary  $t_{right} \leftarrow t_n$  (lines 5–6).

- c) Time interval  $[t_{left}, t_{right}]$  contains a root of  $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$ , and the midpoint of interval  $[t_{left}, t_{right}]$  is  $t_{mid} \leftarrow \frac{t_{left} + t_{right}}{2}$  (line 8).
- d) If  $T_{SPCL}^{[t_1, t_{mid}]} = \epsilon_{MTI}$ , the first time interval  $[t_1, t_1^*]$  can be determined such that  $t_1^* < t_{mid} < t_1^* + 1$  (lines 9–10).
- e) If  $T_{SPCL}^{[t_1, t_{mid}]} > \epsilon_{MTI}$ ,  $T_{SPCL}^{[t_1, t]}$  has a root in the interval  $[t_{mid}, t_n]$ , and the left boundary  $t_{left}$  is replaced by  $t_{mid}$ , i.e.  $t_{left} \leftarrow t_{mid}$  (lines 11–12);
- f) otherwise,  $T_{SPCL}^{[t_1, t]}$  has a root in the interval  $[t_1, t_{mid}]$ , and the right

boundary  $t_{right}$  is replaced by  $t_{mid}$ , i.e.  $t_{right} \Leftarrow t_{mid}$  (lines 13–15).

g) Procedures c)-f) repeat until the first time interval  $[t_{lb_1} \Leftarrow t_1, t_{rb_1} \Leftarrow t_1^*]$  can be determined such that  $t_1^* < t_{left} < t_{right} < t_1^* + 1$  and  $t_1^* \Leftarrow \lfloor t_{left} \rfloor$ . The corresponding regression line is the longest one that starts from  $(t_1, R^{(t_1)})$  and satisfies  $T_{SPCL}^{[t_1, t_1^*]} \geq \epsilon_{MTI}$  (lines 7–17).

**Step 2:** Take  $(t_1^* + 1, R^{(t_1^*+1)})$  as the new starting point and  $(t_n, R^{(t_n)})$  as the ending point. The time interval  $[t_{lb_2} \Leftarrow t_1^* + 1, t_{rb_2} \Leftarrow t_2^*]$  ( $t_2^* \in [t_1^* + 1, t_n]$ ) can be determined following the same procedure introduced in Step 1, and a regression line can be drawn from  $(t_1^* + 1, R^{(t_1^*+1)})$  to  $(t_2^*, R^{(t_2^*)})$  satisfying  $T_{SPCL}^{[t_1^*+1, t_2^*]} \geq \epsilon_{MTI}$  (lines 4–20).

**Step 3:** Repeat Step 2 until the last regression line reaches  $(t_n, R^{(t_n)})$ .

The computation of  $T_{SPCL}$  incurs a complexity of  $O(n)$  (line 9), where  $n$  is the number of data points, i.e.  $n = |\{(t_i, R^{(t_i)})\}|$ . As it is contained in the bisection process ( $O(n \log n)$ ) (lines 4–20), the bisection-based boundary excluded greedy MTI algorithm incurs a complexity of  $O(n^2 \log n)$ .

#### 4.2.4 Boundary Included Optimal MTI Algorithm

As the above boundary included greedy MTI algorithm and bisection-based boundary excluded greedy MTI algorithm may not find the minimal set of time intervals, now we develop a boundary included optimal MTI algorithm, which can deliver the minimal set of boundary included regression lines.

In this optimal MTI algorithm, each point  $(t_i, R^{(t_i)})$  is taken as a vertex  $v_i$  in a graph. There is a directed edge from  $v_i$  to  $v_j$  ( $i < j$ ) of weight 1 if  $T_{SPCL}^{[t_i, t_j]} \geq \epsilon_{MTI}$ . Thus, the task to obtain a minimal set of MTI is converted to one to find the shortest path from  $v_1$  (i.e. point  $(t_1, R^{(t_1)})$ ) to  $v_n$  (i.e. point  $(t_n, R^{(t_n)})$ ). For this task, we extend Dijkstra's shortest path algorithm [21], and the obtained shortest path from

$v_1$  to  $v_n$  corresponds to the minimum set of boundary included regression lines from  $(t_1, R^{(t_1)})$  to  $(t_n, R^{(t_n)})$ .

The boundary included optimal MTI algorithm (Algorithm 4) works as follows.

**Step 1:** Take  $(t_i, R^{(t_i)})$  as vertex  $v_i$ . Initialize the adjacent matrix  $M$  with  $n$  vertices where the weight of the edge between  $v_i$  and  $v_j$  is  $M_{i,j} \Leftarrow 1$  if  $T_{SPCL}^{[t_i, t_j]} \geq \epsilon$  ( $i < j$ ,  $i, j = 1, \dots, n$ ); otherwise,  $M_{i,j} \Leftarrow \infty$  ( $O(n^3)$ ) (lines 1-9 in Algorithm 4).

**Step 2:** Let  $dis(v_i)$  denote the distance from  $v_1$  to  $v_i$ . Initialize the distance  $dis(v_i)$  for every vertex  $v_i$  according to the adjacent matrix  $M$  ( $O(n)$ ) (lines 10-12).

**Step 3:** Mark all vertices as unvisited. Set  $v_1$  as the current vertex, and mark it as visited ( $O(n)$ ) (lines 13-14).

**Step 4:** For current vertex  $v_i$ , considering all the unvisited vertices with distance 1 to  $v_i$ , denoted as  $\{v_k\}$ , compute the distance  $dis(v_k)$  respectively. If this computed  $dis(v_k)$  is less than the previous recorded  $dis(v_k)$ , overwrite the recorded distance with the computed distance. If all vertices have been visited, go to Step 5. Otherwise, set the unvisited vertex  $v_j$  with the smallest  $dis(v_j)$  as the current vertex  $v_i$ , mark it as visited and go back to the beginning of Step 4. ( $O((n + m) \log n)$ , where  $\sum_{v_i} \deg^-(v_i) = 2m$ ,  $\deg^-(v_i)$  is the indegree of  $v_i$ ) (lines 15-25).

**Step 5:** The recorded  $dis(v_n)$  now is minimized, and the corresponding path from  $v_1$  to  $v_n$  is returned (lines 26-32).

Since  $n^3$  dominates  $m \log n$ , the boundary included optimal MTI algorithm incurs a complexity of  $O(n^3)$ , where  $n$  is the number of data points, i.e.  $n = |\{(t_i, R^{(t_i)})\}|$ .

**Algorithm 4** Boundary Included Optimal MTI algorithm

---

**Input:** trust ratings  $R_i^{(t_i)}$ ,  
the threshold  $\epsilon_{MTI}$  of  $T_{SPCL}$  (such as 0.9),  
the given time interval  $[t_1, t_n]$ .

**Output:** the minimal boundary set  $v_b$  of MTI.

- 1: **for all**  $i \in [1, v_n]$  **do**
- 2:   **for all**  $j \in [i, v_n]$  **do**
- 3:     **if**  $T_{SPCL}^{[i,j]} \geq \epsilon_{MTI}$  **then**
- 4:        $M_{i,j} \leftarrow 1$ ;
- 5:     **else**
- 6:        $M_{i,j} \leftarrow \infty$ ;
- 7:     **end if**
- 8:   **end for**
- 9: **end for**
- 10: **for all**  $v_i \in [v_1, v_n]$  **do**
- 11:    $dis(v_i) \leftarrow M_{v_1, v_i}$ ;
- 12: **end for**
- 13: initialize vector  $unvisit \leftarrow \{v_1, v_2, \dots, v_n\}$ ;
- 14: let  $u$  be  $v_1$ ;
- 15: **while**  $unvisit \neq \emptyset$  **do**
- 16:   let  $u$  be  $v_i \in unvisit$  with the smallest  $dis(v_i)$ ;
- 17:   remove  $u$  from  $unvisit$ ;
- 18:   **for all**  $v_j$  with  $M_{u, v_j} = 1$  **do**
- 19:      $temp \leftarrow dis(u) + M_{u, v_j}$ ;
- 20:     **if**  $temp < dis(v_j)$  **then**
- 21:        $dis(v_j) \leftarrow temp$ ;
- 22:        $previous(v_j) \leftarrow u$ ;
- 23:     **end if**
- 24:   **end for**
- 25: **end while**
- 26:  $v_k \leftarrow v_n$ ;
- 27:  $v_b \leftarrow \emptyset$ ;
- 28: **while**  $previous(v_k) \neq v_1$  **do**
- 29:    $v_k \leftarrow previous(v_k)$ ;
- 30:    $v_b \leftarrow v_b \cup v_k$ ;
- 31: **end while**
- 32: **return**  $v_b$ ;

---

**4.2.5 Boundary Excluded Optimal MTI Algorithm**

In this section, we develop a boundary excluded optimal MTI algorithm, which can deliver the minimal set of boundary excluded regression lines.

In this optimal MTI algorithm, the task to obtain a minimal set of MTI is converted to one to find the shortest path from  $v_1$  (i.e. point  $(t_1, R^{(t_1)})$ ) to  $v_n$  (i.e. point  $(t_n, R^{(t_n)})$ ) with *excluded boundaries*, i.e. if there is a directed edge from  $v_i$  to  $v_j$  in the shortest path, the next edge in the path starts from  $v_{j+1}$ , not  $v_j$ . For this task, we extend Dijkstra's shortest path algorithm [21] as follows.

In the boundary included optimal MTI algorithm, when dealing with the current vertex  $v_i$ , which changes from unvisited to visited, we need to update the distance from  $v_1$  to every unvisited vertex with distance 1 to  $v_i$ .

In contrast, in the boundary excluded optimal MTI algorithm, when dealing with the current vertex  $v_i$ , we need to update the distance from  $v_1$  to every unvisited vertex with distance 1 to  $v_{i+1}$ , not  $v_i$ . The obtained shortest path from  $v_1$  to  $v_n$  with excluded boundaries corresponds with the minimal set of boundary excluded regression lines from  $(t_1, R^{(t_1)})$  to  $(t_n, R^{(t_n)})$ .

In this section, we introduce how our boundary excluded optimal MTI algorithm (Algorithm 5) works.

**Step 1:** Take  $(t_i, R^{(t_i)})$  as vertex  $v_i$ . Initialize the adjacent matrix  $M$  with  $n$  vertices where the weight of the edge between  $v_i$  and  $v_j$  is  $M_{i,j} \Leftarrow 1$  if  $T_{SPCL}^{[t_i, t_j]} \geq \epsilon_{MTI}$  ( $i < j, i, j = 1, \dots, n$ ); otherwise,  $M_{i,j} \Leftarrow \infty$  ( $O(n^3)$ ) (lines 1–9 in Algorithm 5).

**Step 2:** Let  $dis(v_i)$  denote the distance from  $v_1$  to  $v_i$ . Initialize the distance  $dis(v_i)$  for every vertex  $v_i$  according to the adjacent matrix  $M$  ( $O(n)$ ) (lines 10–12).

**Step 3:** Mark all vertices as unvisited. Set  $v_1$  as the current vertex, and mark it as visited ( $O(n)$ ) (lines 13–14).

**Step 4:** For current vertex  $v_i$ , considering all the unvisited vertices with distance 1 to its neighbors  $v_{i+1}$ , denoted as  $\{v_k\}$ , compute the distance  $dis(v_k)$  respectively. If this computed  $dis(v_k)$  is less than the previous recorded  $dis(v_k)$ , overwrite the

**Algorithm 5** Boundary Excluded Optimal MTI algorithm

---

**Input:** trust ratings  $R^{(t_i)}$ ,  
the threshold  $\epsilon_{MTI}$  of  $T_{SPCL}$  (such as 0.9),  
the given time interval  $[t_1, t_n]$ .

**Output:** the minimal boundary set  $v_b$  of MTI.

- 1: **for all**  $i \in [1, v_n]$  **do**
- 2:   **for all**  $j \in [i, v_n]$  **do**
- 3:     **if**  $T_{SPCL}^{[i,j]} \geq \epsilon_{MTI}$  **then**
- 4:        $M_{i,j} \leftarrow 1$ ;
- 5:     **else**
- 6:        $M_{i,j} \leftarrow \infty$ ;
- 7:     **end if**
- 8:   **end for**
- 9: **end for**
- 10: **for all**  $v_i \in [v_1, v_n]$  **do**
- 11:    $dis(v_i) \leftarrow M_{v_1, v_i}$ ;
- 12: **end for**
- 13: initialize vector  $unvisit \leftarrow \{v_1, v_2, \dots, v_n\}$ ;
- 14: let  $u$  be  $v_1$ ;
- 15: **while**  $unvisit \neq \emptyset$  **do**
- 16:   let  $u$  be  $v_i \in unvisit$  with the smallest  $dis(v_i)$ ;
- 17:   remove  $u$  from  $unvisit$ ;
- 18:   **for all**  $v_j$  with  $M_{u+1, v_j} = 1$  **do**
- 19:      $temp \leftarrow dis(u) + M_{u+1, v_j}$ ;
- 20:     **if**  $temp < dis(v_j)$  **then**
- 21:        $dis(v_j) \leftarrow temp$ ;
- 22:        $previous(v_j) \leftarrow u$ ;
- 23:     **end if**
- 24:   **end for**
- 25: **end while**
- 26:  $v_k \leftarrow v_n$ ;
- 27:  $v_b \leftarrow \emptyset$ ;
- 28: **while**  $previous(v_k) \neq v_1$  **do**
- 29:    $v_k \leftarrow previous(v_k)$ ;
- 30:    $v_b \leftarrow v_b \cup v_k$ ;
- 31: **end while**
- 32: **return**  $v_b$ ;

---

recorded distance with the computed distance. If all vertices have been visited, go to Step 5. Otherwise, set the unvisited vertex  $v_j$  with the smallest  $dis(v_j)$  as the current vertex  $v_i$ , mark it as visited and go back to the beginning of Step 4.

( $O((n + m) \log n)$ , where  $\sum_{v_i} \deg^-(v_i) = 2m$ ,  $\deg^-(v_i)$  is the indegree of  $v_i$ ) (lines 15–25).

**Step 5:** The recorded  $dis(v_n)$  now is minimized, and the corresponding path from  $v_1$  to  $v_n$  is returned (lines 26–32).

Since  $n^3$  dominates  $m \log n$ , the boundary excluded optimal MTI algorithm incurs a complexity of  $O(n^3)$ , where  $n$  is the number of data points, i.e.  $n = |\{(t_i, R^{(t_i)})\}|$ .

#### 4.2.6 Boundary Mixed Optimal MTI Algorithm

Our empirical studies can demonstrate that both the boundary included optimal MTI algorithm and the boundary excluded optimal MTI algorithm can return the minimal set of MTI with constraints, namely, the boundaries are included or excluded in adjacent time intervals respectively. If there is no such constraint, there may exist a set of boundary mixed time intervals, which is no larger than the set returned by either the boundary included optimal MTI algorithm or the boundary excluded optimal MTI algorithm. This requires the use of a boundary mixed optimal MTI algorithm.

Let us briefly illustrate the difference between the boundary excluded optimal MTI algorithm and the boundary mixed optimal MTI algorithm.

1. In the boundary excluded optimal MTI algorithm, when dealing with the current vertex  $v_i$ , we need to update the distance from  $v_1$  to every unvisited vertex with distance 1 to  $v_{i+1}$ .
2. In contrast, in the boundary mixed optimal MTI algorithm, when dealing with vertex the current  $v_i$ , we need to update the distance from  $v_1$  to every unvisited vertex with distance 1 to  $v_i$  or  $v_{i+1}$ .

The boundary mixed optimal MTI algorithm (Algorithm 6) works as follows.

**Step 1:** Take  $(t_i, R^{(t_i)})$  as vertex  $v_i$ . Initialize the adjacent matrix  $M$  with  $n$  vertices where the weight of the edge between  $v_i$  and  $v_j$  is  $M_{i,j} \Leftarrow 1$  if  $T_{SPCL}^{[t_i, t_j]} \geq \epsilon_{MTI}$

**Table 4.2:** Complexity of MTI algorithms proposed in Section 4.2

The boundary included greedy MTI algorithm	$O(n^3)$
The bisection-based boundary excluded greedy MTI algorithm	$O(n^2 \log n)$
The boundary included optimal MTI algorithm	$O(n^3)$
The boundary excluded optimal MTI algorithm	$O(n^3)$
The boundary mixed optimal MTI algorithm	$O(n^3)$

$(i < j, i, j = 1, \dots, n)$ ; otherwise,  $M_{i,j} \Leftarrow \infty$  ( $O(n^3)$ ) (lines 1–9 in Algorithm 6).

**Step 2:** Let  $dis(v_i)$  denote the distance from  $v_1$  to  $v_i$ . Initialize the distance  $dis(v_i)$  for every vertex  $v_i$  according to the adjacent matrix  $M$  ( $O(n)$ ) (lines 10–12).

**Step 3:** Mark all vertices as unvisited. Set  $v_1$  as the current vertex, and mark it as visited ( $O(n)$ ) (lines 13–14).

**Step 4:** For current vertex  $v_i$ , considering all the unvisited vertices with distance 1 to its neighbor  $v_{i+1}$  or itself  $v_i$ , denoted as  $\{v_k\}$ , compute the distance  $dis(v_k)$  respectively. If this computed  $dis(v_k)$  is less than the previous recorded  $dis(v_k)$ , overwrite the recorded distance with the computed distance. If all vertices have been visited, go to Step 5. Otherwise, set the unvisited vertex  $v_j$  with the smallest  $dis(v_j)$  as the current vertex  $v_i$ , mark it as visited and go back to the beginning of Step 4. ( $O((n+m) \log n)$ , where  $\sum_{v_i} \deg^-(v_i) = 2m$ ,  $\deg^-(v_i)$  is the indegree of  $v_i$ ) (lines 15–30).

**Step 5:** The recorded  $dis(v_n)$  is now minimized, and the corresponding path from  $v_1$  to  $v_n$  is returned (lines 31–39).

Since  $n^3$  dominates  $m \log n$ , the boundary mixed optimal MTI algorithm incurs a complexity of  $O(n^3)$ , where  $n$  is the number of data points, i.e.  $n = |\{(t_i, R^{(t_i)})\}|$ . The complexity of the above five MTI algorithms is listed in Table 4.2.

**Theorem 3:** The boundary mixed optimal MTI algorithm returns the minimal set of boundary mixed time intervals.

**Proof:** Let  $D(v_i, v_j)$  denote the distance from  $v_i$  to  $v_j$ . In the boundary mixed optimal MTI algorithm, the following two conditions hold.

C1: The directed edge from  $v_i$  to  $v_j$  ( $i < j$ ) weights 1 only if  $T_{SPCL}^{[t_i, t_j]} \geq \epsilon_{MTI}$ , i.e.  $D(v_i, v_j) = 1$ ; otherwise it weights infinite, i.e.  $D(v_i, v_j) = \infty$ .

C2: If there is a directed edge from  $v_i$  to  $v_j$  in the shortest path,

- (a) with included boundary, the next edge in the path starts from  $v_j$ , and  $D(v_j, v_j) = 0$ ;
- (b) with excluded boundary, the next edge in the path starts from  $v_{j+1}$ , not  $v_j$ , and  $D(v_j, v_{j+1}) = 0$ .

With these distances, in the boundary mixed optimal MTI algorithm,  $D(v_1, v_n)$  is obtained from Dijkstra's shortest path algorithm.  $D(v_1, v_n)$  is then the minimal length which corresponds to the shortest path from  $v_1$  to  $v_n$ . According to C1 & C2, a path from  $v_1$  to  $v_n$  corresponds with a set of regression lines from  $v_1$  to  $v_n$ , i.e. a set of boundary mixed MTI. Hence, the boundary mixed optimal MTI algorithm returns the minimal set of boundary mixed MTI, and the number of MTI is  $D(v_1, v_n)$ .  $\square$

Similar to Theorem 3, we can prove that our proposed boundary included optimal MTI algorithm returns the minimal set of boundary included MTI, and our proposed boundary excluded optimal MTI algorithm returns the minimal set of boundary excluded MTI.

**Theorem 4:** The boundary mixed optimal MTI algorithm returns a set of MTI which is no larger than the set returned by either the boundary included optimal MTI algorithm or the boundary excluded optimal MTI algorithm.

**Proof:** In the boundary included optimal MTI algorithm, for vertex  $v_i$ , we need to update the distance from  $v_1$  to every unvisited vertex (e.g.  $v_j$ ) with distance 1 to  $v_i$ .

The distance from  $v_1$  to  $v_j$  is denoted as  $d_{v_j}^{(1)}$ , and

$$d_{v_j}^{(1)} = \min\{d_{v_j}^{(1)}, d_{v_i}^{(1)} + 1\}. \quad (4.16)$$

In contrast, in the boundary excluded optimal MTI algorithm, the distance from  $v_1$  to  $v_j$  is denoted as  $d_{v_j}^{(2)}$ , and

$$d_{v_j}^{(2)} = \min\{d_{v_j}^{(2)}, d_{v_{i+1}}^{(2)} + 1\}. \quad (4.17)$$

However, in the boundary mixed optimal MTI algorithm, the distance from  $v_1$  to  $v_j$  is denoted as  $d_{v_j}^{(3)}$ , and

$$d_{v_j}^{(3)} = \min\{d_{v_j}^{(3)}, d_{v_i}^{(3)} + 1, d_{v_{i+1}}^{(3)} + 1\}. \quad (4.18)$$

Obviously, every time when updating the distance from  $v_1$  to  $v_j$ , we can obtain that

$$d_{v_j}^{(3)} \leq d_{v_j}^{(1)} \quad \text{and} \quad d_{v_j}^{(3)} \leq d_{v_j}^{(2)}. \quad (4.19)$$

Then, we have

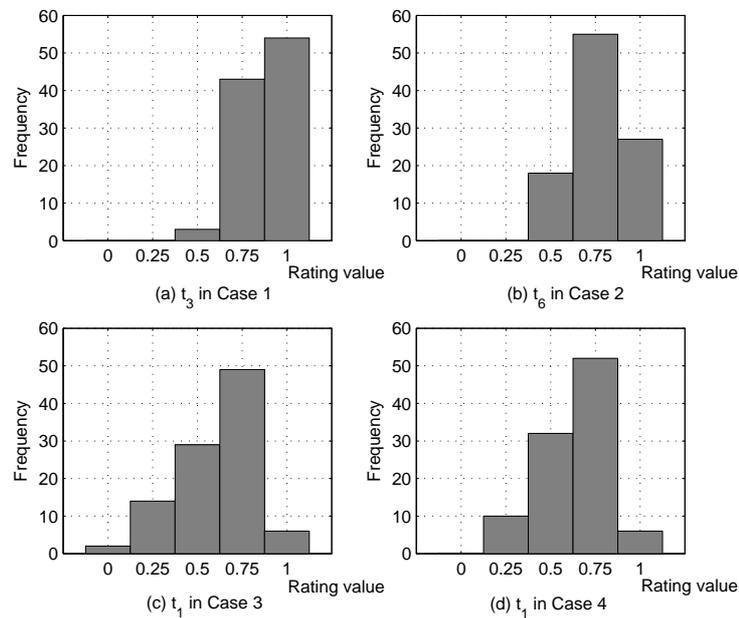
$$d_{v_n}^{(3)} \leq d_{v_n}^{(1)} \quad \text{and} \quad d_{v_n}^{(3)} \leq d_{v_n}^{(2)}. \quad (4.20)$$

The boundary mixed optimal MTI algorithm returns a set of time intervals that is no larger than the one returned by any other two optimal MTI algorithms.  $\square$

Theorem 4 is also confirmed empirically by experiments introduced in Sections 4.2.7.2 and 4.2.7.3.

### 4.2.7 Experiments on MTI Analysis

In this section, we introduce the results of our experiments conducted on both a real-world data set and synthetic data sets. The aim of our experiments is to study the effectiveness and efficiency of our proposed MTI algorithms.



**Figure 4.8:** Rating frequency examples in Experiment 1 of Section 4.2.7.1

#### 4.2.7.1 Experiment 1 – Both Vertical and Horizontal Aggregations

In this experiment, using up to 6 years of teaching evaluations and unit evaluations collected at Macquarie University<sup>1</sup> as data, Sydney, Australia, we study both the vertical aggregation and horizontal aggregation approaches.

At the end of each semester, the Center for Professional Development<sup>2</sup> at Macquarie University asks students to provide feedback on a teacher's teaching quality<sup>3</sup> and a unit's (a subject's) quality<sup>4</sup> using questionnaires.

In this experiment, we use two rating data sets of teaching quality (Cases 1 and 2 in Figs. 4.9 and 4.10) and two rating data sets of unit quality (Cases 3 and 4 in Figs. 4.9 and 4.10). Each data set of teaching quality consists of the ratings given to the same question over 6 years, while each data set of unit quality is for 5 years.

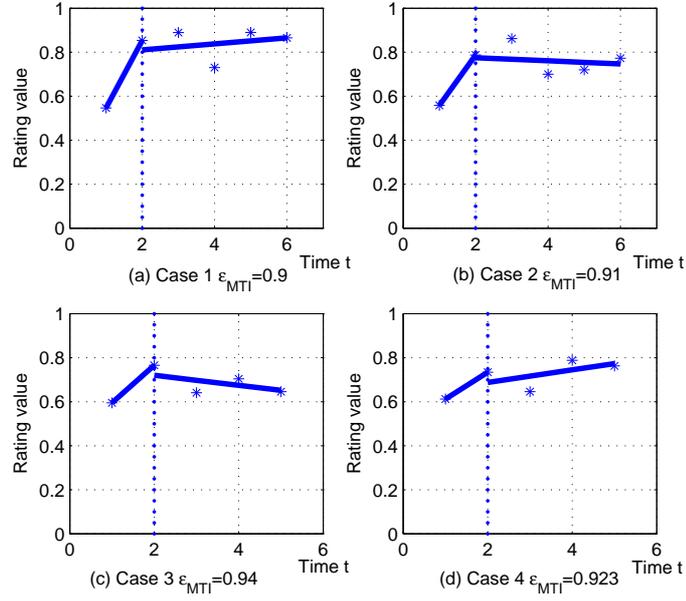
We first study the vertical aggregation of the ratings given to a question in the same

<sup>1</sup><http://www.mq.edu.au/>

<sup>2</sup><http://www.cpd.mq.edu.au/>

<sup>3</sup>[http://www.mq.edu.au/ltc/eval\\_teaching/teds.htm](http://www.mq.edu.au/ltc/eval_teaching/teds.htm)

<sup>4</sup>[http://www.mq.edu.au/ltc/eval\\_teaching/leu.htm](http://www.mq.edu.au/ltc/eval_teaching/leu.htm)

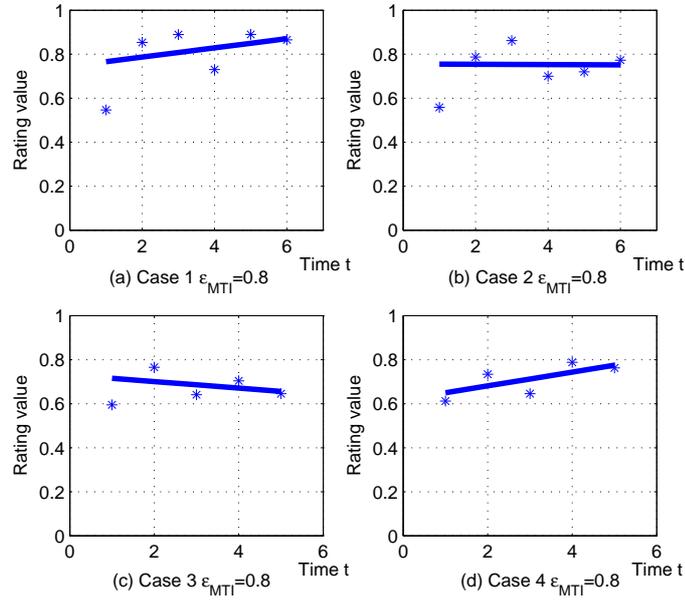


**Figure 4.9:** Single trust vector in Experiment 1 of Section 4.2.7.1

year. The data sets of four cases are plotted in Fig. 4.8(a), Fig. 4.8(b), Fig. 4.8(c) and Fig. 4.8(d) respectively. After vertical aggregation, the centered cluster is  $[0.75, 1]$  in Case 1 (see Fig. 4.8 (a)),  $[0.5, 1]$  in Case 2 (see Fig. 4.8 (b)),  $[0.25, 0.75]$  in Case 3 (see Fig. 4.8 (c)) and  $[0.25, 0.75]$  in Case 4 (see Fig. 4.8 (d)) respectively.

In Case 1 (see Fig. 4.8 (a)), the vertically aggregated rating is 0.8892, which is larger than 0.8775 - the average of ratings, because the rating 0.5 is taken as marginal and it is smaller than the ratings in the centered cluster. In Case 2 (see Fig. 4.8 (b)), the vertically aggregated rating is 0.7725, which is the same as the average of ratings, because there is no marginal rating identified. In contrast, the vertically aggregated rating is 0.5951 in Case 3 (see Fig. 4.8 (c)), which is smaller than the rating average 0.6075. This is because there are more marginal ratings 1 than marginal ratings 0. In Case 4 (see Fig. 4.8 (d)), the vertically aggregated rating is 0.6117. It is smaller than the rating average 0.635 because 1 is the the marginal rating and it is larger than the ratings in the centered cluster.

For horizontal aggregation, if we set the threshold  $\epsilon_{MTI} = 0.8$  for  $T_{SPCL}$ , as plotted



**Figure 4.10:** MTI in Experiment 1 of Section 4.2.7.1

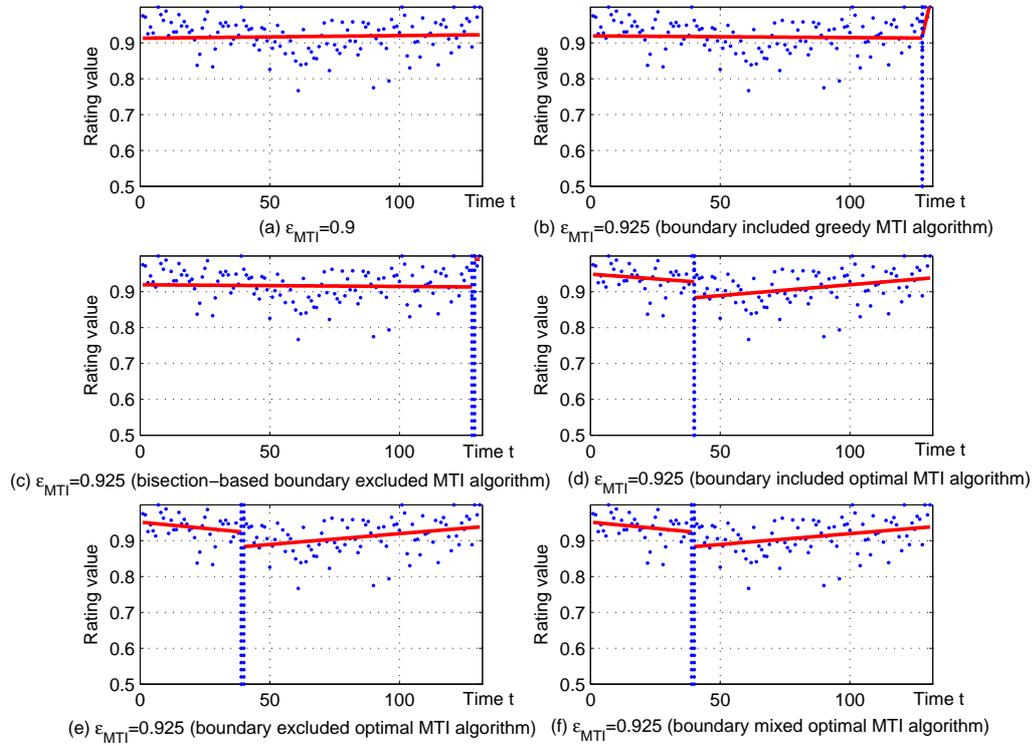
in Fig. 4.9, one trust vector is obtained in each case. In contrast, a higher threshold  $\epsilon_{MTI}$  may lead to more trust vectors (i.e. more time intervals). As plotted in Fig. 4.10, two trust vectors are obtained if  $\epsilon_{MTI} = 0.9$  in Case 1 (see Fig. 4.10(a)),  $\epsilon_{MTI} = 0.91$  in Case 2 (see Fig. 4.10(b)),  $\epsilon_{MTI} = 0.94$  in Case 3 (see Fig. 4.10(c)) and  $\epsilon_{MTI} = 0.923$  in Case 4 (see Fig. 4.10(d)).

Therefore, our trust rating aggregation approach including both vertical aggregation and horizontal aggregation works well for real service-oriented applications.

#### 4.2.7.2 Experiment 2 – Comparison on the Number of Returned MTI

In this experiment, we study our proposed five MTI algorithms (i.e. Algorithms 2 – 6 listed in Table 4.3) over a large set of ratings of one seller obtained from eBay [1].

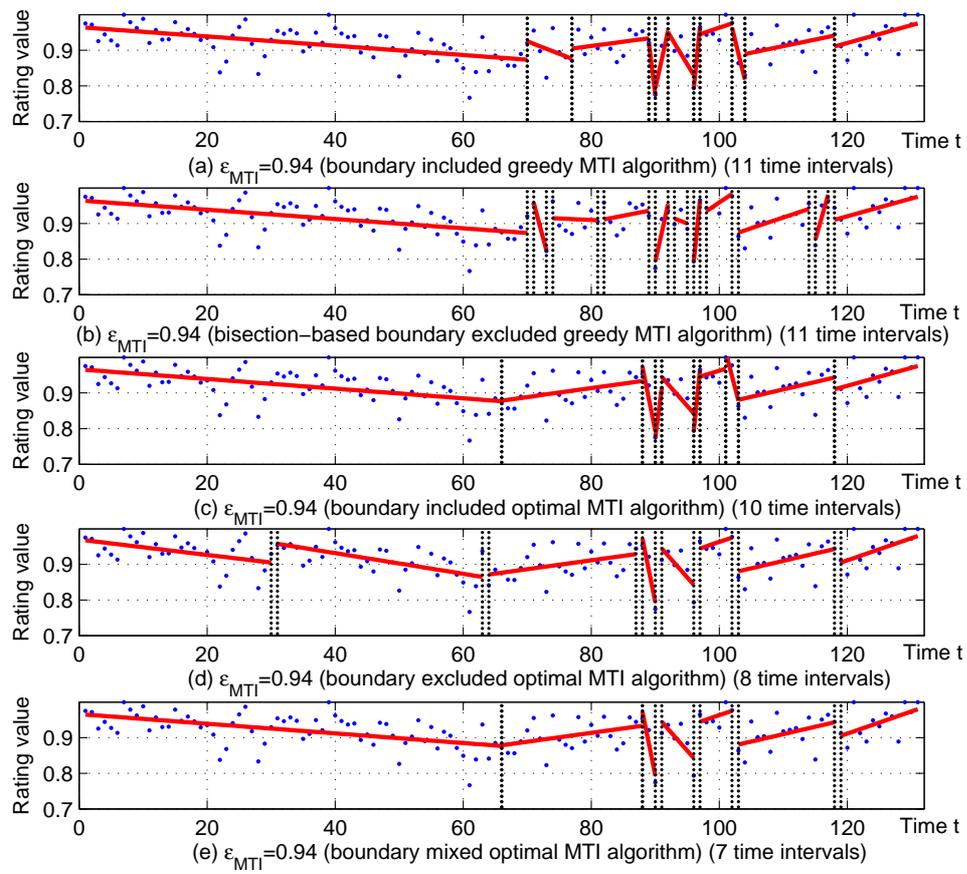
In the rating sample of an eBay seller, there are 11752 ratings in total about the transactions, which happened in 131 days from 13 February 2009 to 23 June 2009. At eBay, a rating can be 1 (“positive”), 0 (“neutral”) or  $-1$  (“negative”). Like the method adopted in [95], the *feedback score percentage* is introduced and calculated as



**Figure 4.11:** MTI in Experiment 2 of Section 4.2.7.2

**Table 4.3:** MTI algorithms compared in Experiments of Section 4.2.7

Algorithm 2	The boundary included greedy MTI algorithm
Algorithm 3	The bisection-based boundary excluded greedy MTI algorithm
Algorithm 4	The boundary included optimal MTI algorithm
Algorithm 5	The boundary excluded optimal MTI algorithm
Algorithm 6	The boundary mixed optimal MTI algorithm



**Figure 4.12:** Comparison of five MTI algorithms in Experiment 2 of Section 4.2.7.2

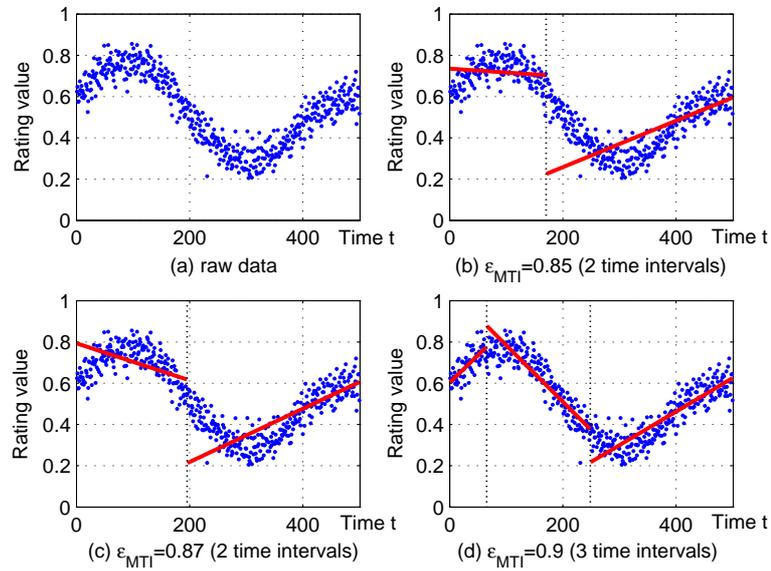
$S_p = \frac{P-N}{P+Ne+N}$ , where  $P$ ,  $Ne$  and  $N$  are the numbers of positive, neutral and negative ratings respectively. We use each day's ratings to compute the feedback score rate  $S_p$ , which is taken as the rating  $R^{(t_i)}$  for time period  $t_i$  (i.e.  $t_i$  is a day in this case and  $i \in [1, 131]$ ). All rating  $\{(t_i, R^{(t_i)}) | 1 \leq i \leq 131\}$  are plotted in Fig. 4.11(a). From Figs 4.11 & 4.12, we can observe that

1. when the threshold of  $T_{SPCL}$  is  $\epsilon_{MTI} = 0.9$ , as plotted in Fig. 4.11(a), with any of the five MTI algorithms listed in Table 4.3, only **1** trust vector is obtained for the whole time interval  $[t_1, t_{131}]$ .
2. with a higher threshold  $\epsilon_{MTI} = 0.925$ , **2** time intervals are obtained by using any of the five MTI algorithms listed in Table 4.3. The results of Algorithms 2 – 6 are plotted in Fig. 4.11(b)(c)(d)(e)(f) respectively.
3. with a further higher threshold  $\epsilon_{MTI} = 0.94$ , we can also observe from Fig. 4.12 that both the boundary included greedy MTI algorithm and the bisection-based boundary excluded greedy MTI algorithm return **11** time intervals (see Fig. 4.12(a)(b)); the boundary included optimal MTI algorithm returns **10** time intervals (see Fig. 4.12(c)); the boundary excluded optimal MTI algorithm returns **8** time intervals (see Fig. 4.12(d)); and the boundary mixed optimal MTI algorithm returns **7** time intervals (see Fig. 4.12(e)).

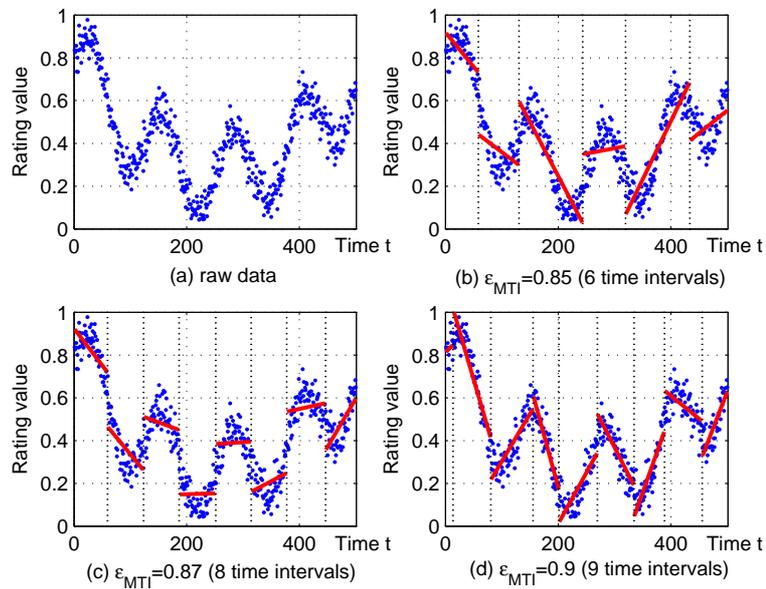
Based on the above results, among all five MTI algorithms listed in Table 4.3, we can observe that the boundary mixed optimal MTI algorithm can return a set of MTI which is no larger than any set returned by other algorithms. This confirms Theorem 4 empirically.

#### 4.2.7.3 Experiment 3 – Comparison on Efficiency

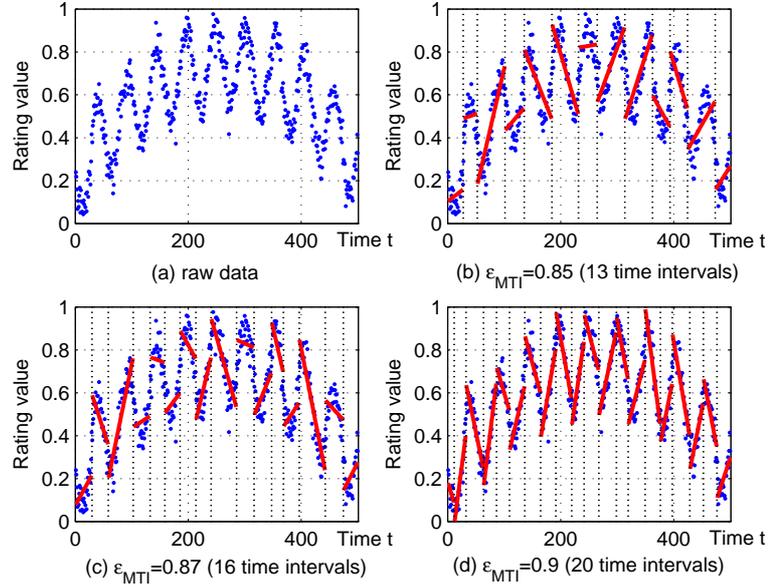
In this experiment, we use large-scale synthetic rating data sets to compare the efficiency of our proposed five MTI algorithms (i.e. Algorithms 2 – 6 listed in Table 4.3).



**Figure 4.13:** Case 1 in Experiment 3 of Section 4.2.7.3 with the boundary mixed optimal MTI algorithm



**Figure 4.14:** Case 2 in Experiment 3 of Section 4.2.7.3 with the boundary mixed optimal MTI algorithm



**Figure 4.15:** Case 3 in Experiment 3 of Section 4.2.7.3 with the boundary mixed optimal MTI algorithm

We conducted our experiments on top of Matlab 7.6.0.324 (R2008a) running on a Dell Vostro V1310 laptop with an Intel Core 2 Duo T5870 2.00GHz CPU and 3GB RAM. Each result of the consumed CPU time is the average of three independent executions with very minor differences in time.

In this experiment, three different sets of ratings have been used, which are plotted in Fig. 4.13(a) (Case 1), Fig. 4.14(a) (Case 2) and Fig. 4.15(a) (Case 3) respectively. Each data set consists of 500 trust ratings distributed in 500 time periods (i.e.  $t_i \in [t_1, t_{500}]$ ). Applying the boundary mixed optimal MTI algorithm to each case produces the following results.

**Case 1:** In this case (see Fig. 4.13(a)), with the threshold of  $T_{SPCL}$  set as  $\epsilon_{MTI} = 0.85$ , 0.87 or 0.9 respectively, we can obtain 2 or 3 time intervals as plotted in Fig. 4.13(b)(c)(d).

**Case 2:** As the trust trend changes a little more frequently in this case (see Fig. 4.14(a)), when the threshold is set as  $\epsilon_{MTI} = 0.85$ , 0.87 or 0.9 respectively, 6, 8

or 9 time intervals are obtained (see Fig. 4.14(b)(c)(d)).

**Case 3:** In contrast, in Case 3 (see Fig. 4.15(a)), the trust trend changes the most frequently in all three cases. With the same thresholds  $\epsilon_{MTI} = 0.85, 0.87$  or  $0.9$ , there are 13 (see Fig. 4.15(b)), 16 (see Fig. 4.15(c)) or 20 time intervals (see Fig. 4.15(d)) obtained respectively.

Thus, in all three cases, with threshold  $\epsilon_{MTI} = 0.85$ , we can use 2, 6 or 13 trust vectors respectively to approximately represent 500 trust ratings. With a high threshold  $\epsilon_{MTI} = 0.9$ , we can use 3, 9 or 20 trust vectors respectively to approximately represent 500 trust ratings. Thus, with our proposed algorithms, a small set of values can represent a large set of trust ratings with well preserved trust features.

In addition, in all three cases using different thresholds, we can compare the consumed CPU time of the five MTI algorithms listed in Table 4.3. From the results listed in Table 4.4, we can derive the following conclusions.

#### **Comparison of different algorithms on efficiency and the number of returned MTI:**

1. The consumed CPU time of the bisection-based boundary excluded greedy MTI algorithm is only 0.6%-8.1% of that of the boundary included greedy MTI algorithm. In addition, the former algorithm returns a set of MTI which is no larger than that returned by the latter algorithm. Hence, the bisection-based boundary excluded greedy MTI algorithm outperforms the boundary included greedy MTI algorithm in terms of efficiency.
2. The bisection-based boundary excluded greedy MTI algorithm consumes less than 0.6% of the CPU time consumed by any of the three optimal MTI algorithms. Thus, we can conclude that the bisection-based boundary excluded greedy MTI algorithm runs much faster than any of the three optimal MTI algorithms.

**Algorithm 6** Boundary Mixed Optimal MTI algorithm**Input:** trust ratings  $R^{(t_i)}$ , the threshold  $\epsilon_{MTI}$  of  $T_{SPCL}$ , the given time interval  $[t_1, t_n]$ .**Output:** the minimal boundary set  $v_b$  of MTI.

```

1: for all  $i \in [1, v_n]$  do
2:   for all  $j \in [i, v_n]$  do
3:     if  $T_{SPCL}^{[i,j]} \geq \epsilon_{MTI}$  then
4:        $M_{i,j} \leftarrow 1$ ;
5:     else
6:        $M_{i,j} \leftarrow \infty$ ;
7:     end if
8:   end for
9: end for
10: for all  $v_i \in [v_1, v_n]$  do
11:    $dis(v_i) \leftarrow M_{v_1, v_i}$ ;
12: end for
13: initialize vector  $unvisit \leftarrow \{v_1, v_2, \dots, v_n\}$ ;
14: let  $u$  be  $v_1$ ;
15: while  $unvisit \neq \emptyset$  do
16:   let  $u$  be  $v_i \in unvisit$  with smallest  $dis(v_i)$ ;
17:   remove  $u$  from  $unvisit$ ;
18:   for all  $v_j$  with  $M_{u, v_j} = 1$  or  $M_{u+1, v_j} = 1$  do
19:      $temp \leftarrow \min\{dis(u), dis(u) + M_{u, v_j}, dis(u) + M_{u+1, v_j}\}$ ;
20:     if  $(temp == dis(u) + M_{u+1, v_j}) \& (temp < dis(u))$  then
21:        $dis(u) \leftarrow temp$ ;
22:        $previous_{1, v_j} \leftarrow u$ ;
23:        $previous_{2, v_j} \leftarrow u + 1$ ;
24:     else if  $(temp == dis(u) + M_{u, v_j}) \& (temp < dis(u))$  then
25:        $dis(u) \leftarrow temp$ ;
26:        $previous_{1, v_j} \leftarrow u$ ;
27:        $previous_{2, v_j} \leftarrow u$ ;
28:     end if
29:   end for
30: end while
31:  $v_k \leftarrow v_n$ ;
32:  $v_{1,b} \leftarrow v_1$ ;
33:  $v_{2,b} \leftarrow v_n$ ;
34: while  $previous_{1, v_k} \neq v_1$  do
35:    $v_{2,b} \leftarrow previous_{1, v_k} \cup v_{2,b}$ ;
36:    $v_k \leftarrow previous_{1, v_k}$ ;
37:    $v_{1,b} \leftarrow v_{1,b} \cup v_k$ ;
38: end while
39: return  $v_b \leftarrow [v_{1,b}^T \ v_{2,b}^T]^T$ ;

```

**Table 4.4:** Consumed CPU time in seconds and the number of returned MTI in Experiment 3 of Section 4.2.7.3

$\epsilon_{MTI}$	Boundary included greedy MTI algorithm		Bisection-based boundary excluded greedy MTI algorithm		Boundary included optimal MTI algorithm			Boundary excluded optimal MTI algorithm			Boundary mixed optimal MTI algorithm					
	time (s)	number	time (s)	number	Step 1	Steps 2-5	Total	Step 1	Steps 2-5	Total	Step 1	Steps 2-5	Total	number		
Case 1	0.85	3	0.3	2	1296.3	1.4	1297.7	2	1295.1	4	1299.1	2	1296.1	5.5	1301.6	2
	0.87	3	0.2	2	1299.1	1.4	1300.5	2	1297.8	3.9	1301.7	2	1296.8	5.5	1302.3	2
	0.9	4	0.3	3	1296.2	1.4	1297.6	3	1298.4	3.9	1302.3	3	1300.9	5.5	1306.4	3
	1	499	8.0	250	1301	1.6	1302.6	499	1289	4	1293	250	1289.6	5.5	1295.1	250
Case 2	0.85	7	0.4	6	1297.3	1.5	1298.8	6	1297.6	4	1301.6	6	1294.7	5.5	1300.2	6
	0.87	10	0.5	8	1298.4	1.4	1299.8	9	1296.9	3.9	1300.8	8	1296.8	5.5	1302.3	8
	0.9	10	0.5	9	1296.6	1.4	1298	9	1299.6	3.9	1303.5	9	1305.3	5.6	1310.9	9
	1	499	8.0	250	1299.4	1.6	1301	499	1286.7	4	1290.7	250	1288.8	5.5	1294.3	250
Case 3	0.85	15	0.6	13	1296.4	1.5	1297.9	14	1297.8	3.9	1301.7	13	1294.6	5.5	1300.1	13
	0.87	20	0.7	18	1299.1	1.5	1300.6	19	1296.1	3.9	1300	16	1301.8	5.5	1307.3	16
	0.9	21	0.8	20	1298.2	1.5	1299.7	20	1301.1	3.9	1305	20	1302.6	5.5	1308.1	20
	1	499	8.0	250	1298.4	1.6	1300	499	1286.6	4	1290.6	250	1290	5.5	1295.5	250

3. For each of the three optimal MTI algorithms, we can divide the whole algorithm into two parts, including the generation of the adjacency matrix (Step 1 introduced in Section 4.2.5 or Section 4.2.6), and the Dijkstra's algorithm based MTI analysis (Steps 2-5). As listed in Table 4.4, the former part takes above 99.6% of the total consumed CPU time since it has to check if  $T_{SPCL} \geq \epsilon_{MTI}$  for all  $\frac{n^2}{2} - n$  possible edges, while the latter part takes less than 0.4% of the total consumed CPU time.
4. With the three optimal MTI algorithms, the shortest consumed CPU time is only 1.5% less than the longest one. Hence, no matter which of the three cases and with what threshold  $\epsilon_{MTI}$  is used (e.g. 0.85, 0.87, 0.9 or 1), all three optimal MTI algorithms consume almost the same CPU time.
5. The boundary mixed optimal MTI algorithm returns a set of MTI which is no larger than the set returned by any of the other four MTI algorithms. This again confirms Theorem 4 empirically.

**Comparison on efficiency and the number of returned MTI in different cases:**

1. From Case 1 to Case 3 (see Fig. 4.13(a), Fig. 4.14(a) and Fig. 4.15(a)), the trust trend changes more and more frequently. With the same MTI algorithm and the same threshold, this change leads to a larger set of MTI. For example, from Case 1 to Case 3, with threshold  $\epsilon_{MTI} = 0.9$ , the boundary mixed optimal MTI algorithm returns 3, 9 and 20 time intervals respectively.
2. From Case 1 to Case 3, when the trust trend changes more and more frequently, with the same threshold the bisection-based boundary excluded greedy MTI algorithm needs more and more CPU time to determine the set of MTI. For example, from Case 1 to Case 3, with threshold  $\epsilon_{MTI} = 0.9$ , this algorithm consumes 0.3, 0.5, and 0.8 seconds of CPU time respectively.

---

**Comparison on efficiency and the number of returned MTI with different thresholds:**

1. When threshold  $\epsilon_{MTI}$  becomes higher, with the same algorithm, a larger set of MTI is returned.
2. When  $\epsilon_{MTI}$  becomes higher, the bisection-based boundary excluded greedy MTI algorithm consumes more CPU time. However, even with the highest threshold  $\epsilon_{MTI} = 1$ , its consumed CPU time is still around 0.6% of that consumed by any of the three optimal MTI algorithms.

Therefore, incorporating the results in Experiment 2 of Section 4.2.7.2, we can see that the bisection-based boundary excluded greedy MTI algorithm is useful when processing large-scale rating data, because it consumes much less CPU time than any of the other four MTI algorithms. However, it may not return the minimal set of MTI. In contrast, the boundary mixed optimal MTI algorithm can return the smallest set of MTI among all five MTI algorithms, but it consumes much more CPU time than the greedy algorithms.

**4.2.7.4 Experiment 4 – Comparison on MTI Goodness-of-Fit**

With our proposed MTI algorithms, a small set of MTI can represent a large set of trust ratings well. However, how well have the trust features been preserved? Now we compare the final trust value aggregated from a set of MTI with the final trust value aggregated from trust ratings directly to see how well the trust features are preserved.

In this section, prior to presenting the detailed analysis, we must first define the *final trust value aggregated from a set of MTI*.

**Definition 21:** With a set of MTI covering the ratings  $\{(t_i, R^{(t_i)})\}$  in time interval  $[t_1, t_n]$

$$\{[t_{lb_i}, t_{rb_i}] | 1 \leq i \leq h, t_{lb_1} = t_1, t_{rb_h} = t_n\}, \quad (4.21)$$

the final trust value aggregated from the set of MTI is

$$\tilde{T}_{FTL}(\{[t_{lb_i}, t_{rb_i}]\}) = \frac{\sum_{i=1}^h T_{FTL}^{[t_{lb_i}, t_{rb_i}]} \sum_{k=lb_i}^{rb_i} w_{t_k}}{\sum_{k=1}^n w_{t_k}}, \quad (4.22)$$

where  $w_{t_k}$  is defined in Eq. (3.2) of Definition 1.

Now we introduce the definition of *MTI goodness-of-fit*, which is measured from the relative difference between the final trust value aggregated from a set of MTI according to Definition 21, and the final trust value aggregated from trust ratings directly according to Definition 1. This measurement indicates how well the trust features are preserved by the set of MTI. A high goodness-of-fit of the set of MTI indicates the high effectiveness of the corresponding MTI algorithm.

**Definition 22:** With a set of MTI covering the ratings  $\{(t_i, R^{(t_i)})\}$  in time interval  $[t_1, t_n]$

$$\{[t_{lb_i}, t_{rb_i}] | 1 \leq i \leq h, t_{lb_1} = t_1, t_{rb_h} = t_n\}, \quad (4.23)$$

the *MTI goodness-of-fit* is

$$G_{MTI}(\{[t_{lb_i}, t_{rb_i}]\}) = 1 - \frac{|T_{FTL}^{[t_1, t_n]} - \tilde{T}_{FTL}(\{[t_{lb_i}, t_{rb_i}]\})|}{T_{FTL}^{[t_1, t_n]}}, \quad (4.24)$$

where  $T_{FTL}^{[t_1, t_n]}$  is defined in Definition 1 and  $\tilde{T}_{FTL}(\{[t_{lb_i}, t_{rb_i}]\})$  is defined in Definition 21.

With Definition 22, we can study the goodness-of-fit of the set of MTI returned by each of two boundary excluded MTI algorithms, including both the bisection-based boundary excluded greedy MTI algorithm and the boundary excluded optimal MTI algorithm.

**Theorem 5:** The goodness-of-fit of the set of MTI returned by either the bisection-based boundary excluded greedy MTI algorithm or the boundary excluded optimal MTI algorithm is 100%.

**Proof:** Let

$$\{[t_{lb_i}, t_{rb_i}] | 1 \leq i \leq h, t_{lb_1} = t_1, t_{rb_h} = t_n\} \quad (4.25)$$

denote a set of MTI returned by either the bisection-based boundary excluded greedy MTI algorithm or the boundary excluded optimal MTI algorithm. As the set of MTI is boundary excluded, we have

$$t_{rb_{j-1}+1} = t_{lb_j} \quad \text{for } j \in [2, h]. \quad (4.26)$$

By substituting Eq. (4.26) into Eq. (4.25), we then have

$$\{[t_{rb_{i-1}+1}, t_{rb_i}] | 1 \leq i \leq h, t_{rb_0+1} = t_1, t_{rb_h} = t_n\}. \quad (4.27)$$

For time interval  $[t_{rb_{i-1}+1}, t_{rb_i}]$ , according to Definition 1, we have

$$T_{FTL}^{[t_{rb_{i-1}+1}, t_{rb_i}]} = \frac{\sum_{k=rb_{i-1}+1}^{rb_i} w_{t_k} R^{(t_k)}}{\sum_{k=rb_{i-1}+1}^{rb_i} w_{t_k}}, \quad (4.28)$$

i.e.

$$T_{FTL}^{[t_{rb_{i-1}+1}, t_{rb_i}]} \sum_{k=rb_{i-1}+1}^{rb_i} w_{t_k} = \sum_{k=rb_{i-1}+1}^{rb_i} w_{t_k} R^{(t_k)}. \quad (4.29)$$

For all time intervals  $\{[t_{rb_{i-1}+1}, t_{rb_i}] | 1 \leq i \leq h\}$  in  $[t_1, t_n]$ , we have

$$\begin{aligned} \sum_{i=1}^h T_{FTL}^{[t_{rb_{i-1}+1}, t_{rb_i}]} \sum_{k=rb_{i-1}+1}^{rb_i} w_{t_k} &= \sum_{i=1}^h \sum_{k=rb_{i-1}+1}^{rb_i} w_{t_k} R^{(t_k)} \\ &= \sum_{k=1}^n w_{t_k} R^{(t_k)}. \end{aligned} \quad (4.30)$$

According to the definition of  $T_{FTL}^{[t_1, t_n]}$  in Definition 1, we have

$$T_{FTL}^{[t_1, t_n]} = \frac{\sum_{k=1}^n w_{t_k} R^{(t_k)}}{\sum_{k=1}^n w_{t_k}}. \quad (4.31)$$

If we now substitute Eq. (4.30) into Eq. (4.31), we then have

$$T_{FTL}^{[t_1, t_n]} = \frac{\sum_{i=1}^h T_{FTL}^{[t_{rb_{i-1}+1}, t_{rb_i}]} \sum_{k=rb_{i-1}+1}^{rb_i} w_{t_k}}{\sum_{k=1}^n w_{t_k}}. \quad (4.32)$$

By substituting Eq. (4.32) into Eq. (4.22) in Definition 21, we have

$$\tilde{T}_{FTL}(\{[t_{rb_{i-1}+1}, t_{rb_i}]\}) = T_{FTL}^{[t_1, t_n]}. \quad (4.33)$$

Therefore, according to Eq. (4.24) in Definition 22, we can obtain that the goodness-of-fit of the set of boundary excluded MTI  $\{[t_{rb_{i-1}+1}, t_{rb_i}]\}$  is

$$G_{MTI}(\{[t_{rb_{i-1}+1}, t_{rb_i}]\}) = 1 - \frac{|T_{FTL}^{[t_1, t_n]} - \tilde{T}_{FTL}(\{[t_{rb_{i-1}+1}, t_{rb_i}]\})|}{T_{FTL}^{[t_1, t_n]}} = 100\%. \quad (4.34)$$

□

Next, with Definition 22, we can also study the goodness-of-fit of the set of MTI returned by each of two boundary included MTI algorithms, including both the boundary included greedy MTI algorithm and the boundary included optimal MTI algorithm.

**Theorem 6:** The goodness-of-fit of the set of MTI returned by either the boundary included greedy MTI algorithm or the boundary included optimal MTI algorithm is less than 100%.

**Proof:** Let

$$\{[t_{lb_i}, t_{rb_i}]\} | 1 \leq i \leq h, t_{lb_1} = t_1, t_{rb_h} = t_n \quad (4.35)$$

denote a set of MTI returned by either the boundary included greedy MTI algorithm or the boundary included optimal MTI algorithm. As the set of MTI is boundary included, we have

$$t_{rb_{j-1}} = t_{lb_j} \text{ for } j \in [2, h]. \quad (4.36)$$

By substituting Eq. (4.36) into Eq. (4.35), we then have

$$\{[t_{rb_{i-1}}, t_{rb_i}] | 1 \leq i \leq h, t_{rb_0} = t_1, t_{rb_h} = t_n\}. \quad (4.37)$$

For time interval  $[t_{rb_{i-1}}, t_{rb_i}]$ , according to Definition 1, we have

$$T_{FTL}^{[t_{rb_{i-1}}, t_{rb_i}]} = \frac{\sum_{k=rb_{i-1}}^{rb_i} w_{t_k} R^{(t_k)}}{\sum_{k=rb_{i-1}}^{rb_i} w_{t_k}}, \quad (4.38)$$

i.e.

$$T_{FTL}^{[t_{rb_{i-1}}, t_{rb_i}]} \sum_{k=rb_{i-1}}^{rb_i} w_{t_k} = \sum_{k=rb_{i-1}}^{rb_i} w_{t_k} R^{(t_k)}. \quad (4.39)$$

For all time intervals  $\{[t_{rb_{i-1}}, t_{rb_i}] | 1 \leq i \leq h\}$  in  $[t_1, t_n]$ , we have

$$\sum_{i=1}^h T_{FTL}^{[t_{rb_{i-1}}, t_{rb_i}]} \sum_{k=rb_{i-1}}^{rb_i} w_{t_k} = \sum_{i=1}^h \sum_{k=rb_{i-1}}^{rb_i} w_{t_k} R^{(t_k)}. \quad (4.40)$$

As in Eq. (4.30)

$$\sum_{i=1}^h \sum_{k=rb_{i-1}+1}^{rb_i} w_{t_k} R^{(t_k)} = \sum_{k=1}^n w_{t_k} R^{(t_k)}, \quad (4.41)$$

we have

$$\sum_{i=1}^h \sum_{k=rb_{i-1}}^{rb_i} w_{t_k} R^{(t_k)} > \sum_{k=1}^n w_{t_k} R^{(t_k)}. \quad (4.42)$$

Then, by substituting Eq. (4.40) into Eq. (4.42), we have

$$\sum_{i=1}^h T_{FTL}^{[t_{rb_{i-1}}, t_{rb_i}]} \sum_{k=rb_{i-1}}^{rb_i} w_{t_k} > \sum_{k=1}^n w_{t_k} R^{(t_k)}, \quad (4.43)$$

i.e.

$$\frac{\sum_{i=1}^h T_{FTL}^{[t_{rb_{i-1}}, t_{rb_i}]} \sum_{k=rb_{i-1}}^{rb_i} w_{t_k}}{\sum_{k=1}^n w_{t_k}} > \frac{\sum_{k=1}^n w_{t_k} R^{(t_k)}}{\sum_{k=1}^n w_{t_k}}. \quad (4.44)$$

According to the definitions of  $T_{FTL}^{[t_1, t_n]}$  in Definition 1 and  $\tilde{T}_{FTL}(\{[t_{rb_{i-1}}, t_{rb_i}]\})$  in Eq. (4.22), from Eq. (4.44) we have

$$\tilde{T}_{FTL}(\{[t_{rb_{i-1}}, t_{rb_i}]\}) > T_{FTL}^{[t_1, t_n]}, \quad (4.45)$$

i.e.

$$|T_{FTL}^{[t_1, t_n]} - \tilde{T}_{FTL}(\{[t_{rb_{i-1}}, t_{rb_i}]\})| > 0. \quad (4.46)$$

Therefore, according to Eq. (4.24) in Definition 22, we can obtain that the goodness-of-fit of the set of boundary included MTI  $\{[t_{rb_{i-1}}, t_{rb_i}]\}$  is

$$G_{MTI}(\{[t_{rb_{i-1}}, t_{rb_i}]\}) = 1 - \frac{|T_{FTL}^{[t_1, t_n]} - \tilde{T}_{FTL}(\{[t_{rb_{i-1}}, t_{rb_i}]\})|}{T_{FTL}^{[t_1, t_n]}} < 100\%. \quad (4.47)$$

□

By applying Definition 22 to the data sets used in Experiments 2 & 3 of Section 4.2.7.2 & 4.2.7.3, the corresponding MTI goodness-of-fit can be evaluated and listed in Table 4.5. From the results listed in Table 4.5, we can obtain the following conclusions.

1. The goodness-of-fit values of the sets of MTI returned by both the bisection-based boundary excluded greedy MTI algorithm (e.g. Fig. 4.11(c) and Fig. 4.12(b)) and the boundary excluded optimal MTI algorithm (e.g. Fig. 4.11(e) and Fig. 4.12(d)) are **100%**.

This confirms Theorem 5 empirically.

2. In this experiment, the goodness-of-fit values of the sets of MTI returned by the boundary included greedy MTI algorithm in Fig. 4.11(b) and Fig. 4.12(a) are **99.97%** and **99.79%** respectively. The goodness-of-fit of the sets of MTI

**Table 4.5:** MTI goodness-of-fit for the data sets used in Experiments 2 & 3 of Section 4.2.7.2 & 4.2.7.3

	(a)	(b)	(c)	(d)	(e)	(f)
Fig. 4.11	N/A	99.97% (boundary included)	100% (boundary excluded)	99.97% (boundary included)	100% (boundary excluded)	100% (boundary mixed)
Fig. 4.12	99.79% (boundary included)	100% (boundary excluded)	99.85% (boundary included)	100% (boundary excluded)	99.99% (boundary mixed)	N/A
Fig. 4.13	N/A	100% (boundary mixed)	100% (boundary mixed)	100% (boundary mixed)	N/A	N/A
Fig. 4.14	N/A	100% (boundary mixed)	100% (boundary mixed)	100% (boundary mixed)	N/A	N/A
Fig. 4.15	N/A	99.99% (boundary mixed)	99.99% (boundary mixed)	99.91% (boundary mixed)	N/A	N/A

returned by the boundary included optimal MTI algorithm in Fig. 4.11(d) and Fig. 4.12(c) are **99.97%** and **99.85%**.

This confirms Theorem 6 empirically.

3. In this experiment, the goodness-of-fit values of the sets of MTI returned by the the boundary mixed optimal MTI algorithm in Fig. 4.12(e) and Fig. 4.15(b)(c)(d) are **99.99%**, **99.99%**, **99.99%** and **99.91%** respectively. In contrast, the MTI goodness-of-fit of the ones in Fig. 4.13(b)(c)(d) and Fig. 4.14(b)(c)(d) are **100%**.

Therefore, the goodness-of-fit of the set of MTI returned by the boundary mixed optimal MTI algorithm can be less than or equal to 100%.

As we pointed out in Chapter 3, a single trust value cannot preserve the trust features well (e.g. depending on whether and how the trust trend changes). Hence, it is necessary to generate a small set of data that represents a large set of trust ratings over

---

a long service history. The research presented in this section confirms that with any of our proposed five MTI algorithms, a small set of values can represent a large set of trust ratings while the trust features can be well preserved. It illustrates the high effectiveness of our proposed algorithms.

### 4.3 Conclusions

In this chapter, we have proposed a novel two dimensional aggregation approach that can aggregate a large set of trust ratings both vertically and horizontally. In the vertical aggregation of trust ratings, we adopt the Gaussian distribution based analysis method and the clustering based analysis method. For the horizontal aggregation of trust ratings, we propose using the service trust vector approach and the multiple time intervals (MTI) analysis approach including a boundary included greedy MTI algorithm, a bisection-based boundary excluded greedy MTI algorithm, a boundary included optimal MTI algorithm, a boundary excluded optimal MTI algorithm and a boundary mixed optimal MTI algorithm. The proposed bisection-based boundary excluded greedy MTI algorithm has a lower time complexity, and it is much faster than any of the other four MTI algorithms. The proposed boundary mixed optimal MTI analysis algorithm can guarantee the representation of a large set of trust ratings with a minimal set of values while highly preserving the trust features. Therefore, our work is significant for large-scale trust data management, transmission and evaluation.

In the boundary mixed optimal MTI algorithm, given a set of ratings and the same threshold, several minimal sets of boundary mixed MTI may exist. Thus, in our future work, the boundary mixed optimal MTI algorithm can be further extended to find the best set of MTI with the largest MTI goodness-of-fit or the largest summation of *SPCL* values.



# Trust-Oriented Composite Service Selection

---

In SOC environments, to satisfy the specified functionality requirement, a service may have to invoke other services forming composite Web services with complex invocations and trust dependencies among services and service providers [69]. Meanwhile, given a set of various services, different compositions may lead to different service structures. Although these certainly enrich the service provision, they greatly increase the computation complexity and thus make trustworthy service selection and discovery a very challenging task.

In the literature, there are some existing studies for service composition and quality driven service selection [29, 69, 94, 101, 103]. However, for trust-oriented composite service selection and discovery, some research problems remain open.

1. The definition of a proper graphic representation of composite services including both probabilistic invocations and parallel invocations is still lacking. The corresponding data structure is also essential. It is fundamental and important to define these representations to support the global trust evaluation of composite services.
2. From the definitions in [41, 45], trust can be taken as the *subjective probability*, i.e. *the degree of belief an individual has in the truth of a proposition* [30, 33], rather than the *objective probability* or *classical probability*, which is *the occur-*

rence frequency of an event [33]. A subjective probability is derived from an individual's personal judgment about a specific outcome (e.g. the evaluation of teaching quality or service quality). It differs from person to person. Hence, classical probability theory is not a good fit for trust evaluation. Instead, *subjective probability theory* [30, 33] should be adopted for trust evaluation.

3. Although there are a variety of trust evaluation methods in a number of different areas [85, 90, 95], no proper mechanism exists for evaluating the global trust of a composite service with a complex structure over service components with different trust values.
4. Taking trust evaluation and the complex structure of composite services into account, effective algorithms are needed for trust-oriented composite service selection and discovery, and are expected to be more efficient than the existing approaches [69, 101].

In this chapter, we first present the service invocation graph and service invocation matrix for composite service representation. In addition, we propose a trust evaluation method for composite services based on Bayesian inference, which is an important component in subjective probability theory. Furthermore, based on the Monte Carlo method, we propose a service selection and discovery algorithm and a QoS constrained service selection algorithm. Experiments have been conducted on composite services of various sizes to compare the proposed algorithms with the existing exhaustive search method [69]. The results illustrate that our proposed algorithms are effective and more efficient.

This chapter is organized as follows. Section 5.1 reviews existing studies in service composition, service selection and trust. Section 5.2 presents our proposed composite services oriented service invocation graph and service invocation matrix. Section 5.3 presents a novel trust evaluation method for composite services. In Section 5.4, a Monte Carlo method based algorithm is proposed for trust-oriented composite service selection and discovery, and experiments are presented for further illustrating that our

proposed algorithm is effective and efficient. In Section 5.5, a QoS constrained Monte Carlo method based algorithm is proposed for trust-oriented composite service selection with QoS constraints, and experiments are presented for further illustrating that our proposed algorithm is effective and efficient. Finally Section 5.6 concludes our work in this chapter.

## **5.1 Related Work on Trust-Oriented Composite Service Selection**

In SOC environments, the composition of services offered by different providers enriches service provision and offers flexibility to service applications. In [67, 68], Medjahed et al. present some frameworks and algorithms for automatically generating composite services from specifications and rules.

In real applications, the criteria of searching services should take into account not only functionalities but also other properties, such as QoS (quality of service) and trust. In the literature, a number of QoS-aware Web service selection mechanisms have been developed, aiming at QoS improvement in composite services [29, 94, 103]. In [103], Zeng et al. present a general and extensible model to evaluate the QoS of composite services. Based on their model, a service selection approach has been introduced using linear programming techniques to compute optimal execution plans for composite services. The work in [29] addresses the selection and composition of Web services based on functional requirements, transactional properties and QoS characteristics. In this model, services are selected in a way that satisfies user preferences, expressed as weights over QoS and transactional requirements. In [94], Xiao et al. present an autonomic service provision framework for establishing QoS-assured end-to-end communication paths across domains. Their algorithms can provide QoS guarantees over domains. The above works have various and different merits. However, none of them has taken parallel invocation into account, which is fundamental and is one of the most

common existing invocations in composite services [69, 101].

Menascé [69] adopts an exhaustive search method to measure service execution time and cost involving probabilistic, parallel, sequential and fastest-predecessor-triggered invocations. However, the algorithm complexity is exponential. Yu et al. [101] study the service selection problem with multiple QoS constraints in composite services, and propose two optimal heuristic algorithms: the combinatorial algorithm and the graph-based algorithm. The former algorithm models service selection as a multidimension multichoice 0-1 knapsack problem. The latter algorithm can be taken as a multiconstraint optimal path problem. Nevertheless, none of these works address any aspect of trust.

The issue of trust has been widely studied in many applications. In e-commerce environments, the trust management system can provide valuable information to buyers and prevent some typical attacks [88, 102]. In Peer-to-Peer information-sharing networks, binary ratings work pretty well as a file is either the definitively correct version or not [98]. In SOC environments, an effective trust management system is critical to identify potential risks, provide objective trust results to clients and prevent malicious service providers from easily deceiving clients and leading to huge monetary loss [85].

In general, the trust from a service client towards a service or a service provider can be taken as being the extent to which the service client *believes* that the service provider can satisfy the client's requirement with desirable performance and quality. Thus, as we have pointed out in Chapter 1, trust is a *subjective belief* and it is therefore best to adopt *subjective probability theory* [33] to deal with trust.

Some works do deal with subjective ratings [40, 91]. Jøsang [40] describes a framework for combining and assessing subjective ratings from different sources based on Dempster-Shafer belief theory. Wang and Singh [91] set up a bijection from subjective ratings to trust values with a mathematical understanding of trust in a variety of multiagent systems. However, their models use either a binary rating (positive or negative) system or a triple rating (positive, negative or uncertain) system, which is

more suitable for security-oriented or P2P file-sharing trust management systems.

As pointed out in [98], in richer service environments such as SOC or e-commerce, a rating in  $[0, 1]$  is more suitable. In [96], Xu et al. propose a reputation-enhanced QoS-based Web service discovery algorithm for service matching, ranking and selection based on existing Web service technologies. Malik et al. [61] propose a set of decentralized techniques aimed at evaluating reputation-based trust using the ratings from peers to facilitate trust-based selection and service composition. However, in these works, neither service invocation nor composite service structure are taken into account. Taking the complex structure of composite services into account, effective algorithms are needed for trust-oriented composite service selection and discovery.

## 5.2 Service Invocation Model

In this section, we present the definitions of our proposed service invocation graph and service invocation matrix for representing the complex structures of composite services. They are essential for our trust-oriented composite service selection and discovery algorithm, which will be introduced in Section 5.4.

### 5.2.1 Composite Services and Invocation Relation

A *composite service* is a conglomeration of services with invocation relations between them. Six atomic invocations [52, 54, 57, 69, 101] are depicted as follows and in Fig. 5.1.

- *Sequential Invocation*: A service  $S$  invokes its unique succeeding service  $A$ . It is denoted as  $Se(S : A)$  (see Fig. 5.1(a)).
- *Parallel Invocation*: A service  $S$  invokes its succeeding services in parallel. E.g. if  $S$  has successors  $A$  and  $B$ , it is denoted as  $Pa(S : A, B)$  (see Fig. 5.1(b)).
- *Probabilistic Invocation*: A service  $S$  invokes its succeeding service with a probability. E.g. if  $S$  invokes successors  $A$  with the probability  $p$  and  $B$  with the

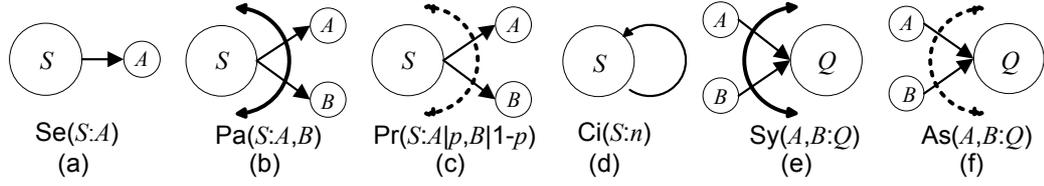


Figure 5.1: Atomic invocations

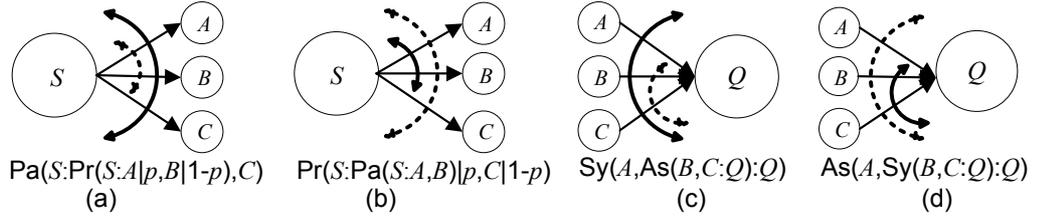


Figure 5.2: Complex invocations examples

probability  $1 - p$ , it is denoted as  $\text{Pr}(S : A|p, B|1 - p)$  (see Fig. 5.1(c)).

- *Circular Invocation*: A service  $S$  invokes itself  $n$  times. It is denoted as  $\text{Ci}(S|n)$  (see Fig. 5.1(d)). A circular invocation can be unfolded by cloning itself  $n$  times [57, 101]. Hence, it can be replaced by  $\text{Se}$  in advance.
- *Synchronous Activation*: A service  $Q$  is activated only when all of its preceding services have been completed. E.g. if  $Q$  has synchronous predecessors  $A$  and  $B$ , it is denoted as  $\text{Sy}(A, B : Q)$  (see Fig. 5.1(e)).
- *Asynchronous Activation*: A service  $Q$  is activated as the result of the completion of one of its preceding services. E.g. if  $Q$  has asynchronous predecessors  $A$  and  $B$ , it is denoted as  $\text{As}(A, B : Q)$  (see Fig. 5.1(f)).

With atomic invocations, some complex invocations can be depicted as Fig. 5.2, which are not clearly introduced in the existing work in this research area.

- *Probabilistic inlaid parallel invocation*, denoted as  $\text{Pa}(S : \text{Pr}(S : A|p, B|1 - p), C)$ .

- *Parallel inlaid probabilistic invocation*,  
denoted as  $\text{Pr}(S : \text{Pa}(S : A, B) | p, C | 1 - p)$ .
- *Asynchronous inlaid synchronous activation*,  
denoted as  $\text{Sy}(A, \text{As}(B, C : Q) : Q)$ .
- *Synchronous inlaid asynchronous activation*,  
denoted as  $\text{As}(A, \text{Sy}(B, C : Q) : Q)$ .

## 5.2.2 An Example: Travel Plan

Here we introduce an example of composite services.

**Example 1:** Smith in Sydney, Australia is making a travel plan to attend an international conference in Stockholm, Sweden. His plan includes conference registration, an airline ticket from Sydney to Stockholm, accommodation and local transportation.

Regarding conference registration *Reg*, Smith could pay *Online* or by *Fax* with a credit card *Ccard*. Regarding accommodation reservation *Acc*, Smith could make a reservation at Hotels *Ha*, *Hb* or *Hc* with credit card *Ccard*. According to the hotel choice, Smith could arrange the local transportation, e.g. take a *Taxi* to *Ha*, or take a *Taxi* or a *Bus* to either *Hb* or *Hc*. Regarding airplane booking *Air*, Smith could choose from Airlines *Aa*, *Ab* and *Ac*, using the credit card *Ccard* for the payment. Smith chooses the services according to their trust values. He will have a higher probability of choosing the service with a better trust value.

In this example, with a starting service *START* and an ending service *END*, the composite services consisting of all possibilities of the travel plan can be depicted by a service invocation graph (*SIG*) (Fig. 5.3). One of all feasible travel plans is a service execution flow as depicted in Fig. 5.4.

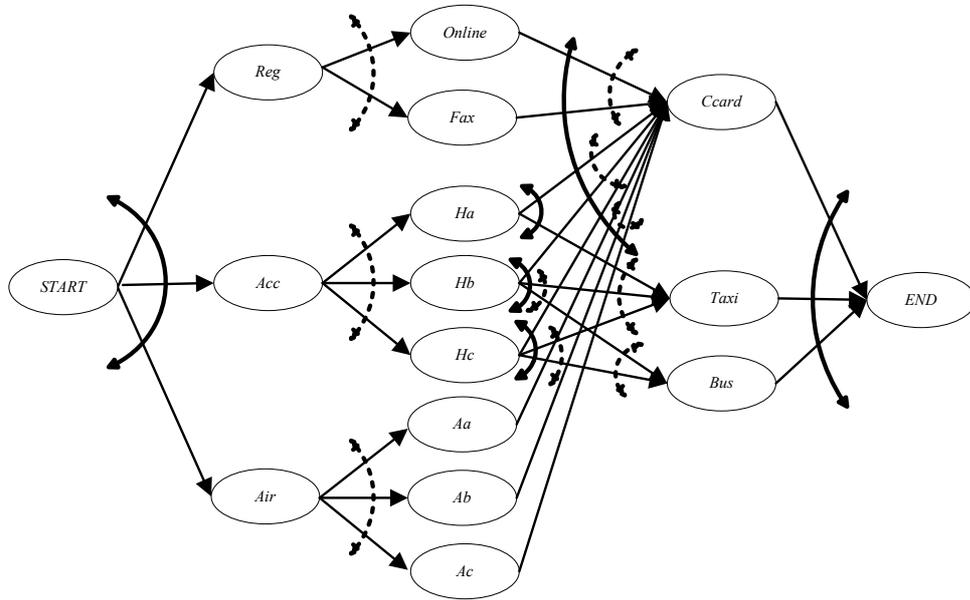


Figure 5.3: The SIG for the travel plan of Smith

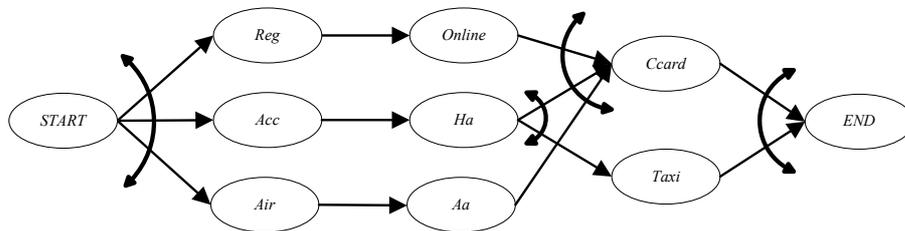


Figure 5.4: A service execution flow

### 5.2.3 Service Invocation Graph

The structure of a composite service can be represented by a service invocation graph (*SIG*), with the initial definition as follows.

**Definition 23:** The *service invocation graph* (*SIG*) is a directed graph  $G = (V, E, R)$ , where  $V$  is a finite set of vertices,  $E$  is a finite set of directed edges and  $R$  is the set of atomic invocations *Se*, *Pa*, *Pr*, *Ci*, *Sy* and *As*. In  $G$ , each vertex  $v \in V$  represents a service.  $\forall e = (v_1, v_2) \in E$  ( $v_1, v_2 \in V$ ) is a directed edge, where  $v_1$  is the *invoking vertex* and  $v_2$  is the *invoked vertex*. Here  $v_1$  is the *direct predecessor* of  $v_2$  and  $v_2$  is the *direct successor* of  $v_1$ . It is denoted as  $v_1 \succeq v_2$ .

**Definition 24:** Given a service invocation graph  $G = (V, E, R)$ , vertex  $v_2 \in V$  is *invocational* from vertex  $v_1 \in V$  if  $(v_1, v_2) \in E$  or there is a directed path  $P$  in  $G$  where  $v_1$  is the starting vertex and  $v_2$  is the ending vertex. If  $v_2$  is invocational from  $v_1$ , it is denoted as  $v_1 \succ v_2$ .

In addition, if  $v_1 \succ v_2$ ,  $v_1$  is the predecessor of  $v_2$  and  $v_2$  is the successor of  $v_1$ . Obviously, the *invocational* relation is transitive, i.e. if  $v_1 \succ v_2$ ,  $v_2 \succ v_3$ , then  $v_1 \succ v_3$ .

**Definition 25:** In a service invocation graph, the *service invocation root* is the entry vertex without any predecessors, and the *service invocation terminal* is the exit vertex without any successors.

Based on the above definitions, *SIG* is well-defined as follows.

**Definition 26:** A composite service can be represented by a *service invocation graph*

$$SIG = (V, I_p, R_p, I_s, R_s), \quad (5.1)$$

where

- In a *SIG*, there are only one service invocation root *START* and only one service invocation terminal *END*;
- $V = \{v_i | v_i \text{ is a vertex, } v_i = \text{START or } \text{START} \succ v_i\}$ ;

- $I_p = \{I_{p_i} | v_i \in V\}$  and  $I_{p_i}$  is a set of direct predecessors invoking  $v_i$ , i.e.  $I_{p_i} = \{p_{ij} | p_{ij} \succeq v_i\}$ ;
- $R_p$  represents a set of activation relations between  $I_p$  and  $V$ , which includes atomic activations Sy and As;
- $I_s = \{I_{s_i} | s_{ij} \in V\}$  and  $I_{s_i}$  is a set of direct successors invoked by  $v_i$ , i.e.  $I_{s_i} = \{s_{ij} | v_i \succeq s_{ij}\}$ ;
- $R_s$  represents a set of invocation relations between  $V$  and  $I_s$ , which includes atomic invocations Se, Pa, Pr and Ci.

Let  $\emptyset$  denote the empty invocation relation set. In a *SIG*, if  $I_{p_i} = \emptyset$ , then  $v_i = START$ . Similarly, if  $I_{s_i} = \emptyset$ , then  $v_i = END$ .

**Definition 27:** A *service execution flow (SEF)* of a *SIG*  $G = (V, E, R)$  is a graph  $G' = (V', E', R')$ , where  $R'$  contains Se, Pa, Sy and Ci,  $V' \subseteq V$  and  $E' \subseteq E$ . In addition,  $\forall v' \in V'$ ,  $v'$  is invocational from the service invocation root *START* of  $G$ , and the service invocation terminal *END* of  $G$  is invocational from  $v'$ .

## 5.2.4 Service Invocation Matrix

In Section 5.2.3, *SIG* provides a clear picture of service invocation relations in composite services. However, an underlying data structure is essential to represent and store vertices and invocation relations. Here we propose a *service invocation matrix* - an algebraic representation of composite services.

**Definition 28:** A composite service can be represented by a *service invocation matrix*

$$SIM = \langle M_{ij} \rangle_{1 \leq i \leq n, 1 \leq j \leq n}, \quad (5.2)$$

where

- $n$  is the number of vertices in the composite services;

- $M_{ij} = 0$  iff there is no invocation from vertex  $i$  to vertex  $j$ ;
- $M_{ij} = \langle M_{ij}^{(1)}, M_{ij}^{(2)}, \dots, M_{ij}^{(k)} \rangle$  ( $i \neq j$ ) represents the invocations from vertex  $i$  to vertex  $j$ , and  $k$  is the number of all invocations from  $i$  to  $j$ ;
- $M_{ij}^{(h)}$  ( $1 \leq h \leq k$ ) is an integer which represents an invocation type from vertex  $i$  to vertex  $j$ ;
  - If it is a parallel invocation,  $M_{ij}^{(h)} = 2m_1$  ( $m_1 = 1, 2 \dots$ ), where  $m_1$  increases from 1 continuously and different  $m_1$  values indicate different parallel invocations Pas;
  - If it is a probabilistic invocation,  $M_{ij}^{(h)} = 2m_2 - 1$  ( $m_2 = 1, 2 \dots$ ), where  $m_2$  increases from 1 continuously and different  $m_2$  values indicate different probabilistic invocations Prs;
- $M_{ii}$  is an integer representing the number of circular times Ci occurs in vertex  $i$ .

According to Definition 28, we have the following property.

**Property 11:**  $\langle M_{ij}^{(1)}, M_{ij}^{(2)} \rangle = \langle M_{ij}^{(2)}, M_{ij}^{(1)} \rangle$

Taking the Travel Plan (Fig. 5.3) in Section 5.2.2 as an example, non-zero entities of the *SIM* are listed in Table 5.1. Our proposed *SIM* can cover all atomic invocation structures and the complex invocation structures derived from them.

## 5.3 Trust Evaluation in Composite Services

In this section, we introduce our trust evaluation models for composite services. In Section 5.3.1, a trust estimation model is proposed to estimate the trust value of each service component from a series of ratings according to Bayesian inference [30, 33], which is an important component in subjective probability theory. These ratings are provided by service clients and stored by a service trust management authority. In Section 5.3.2, a global trust computation model is proposed to compute the global trust value of a composite service based on the trust values of all service components.

**Table 5.1:** Non-zeros of *SIM* in Travel Plan

$i$	$j$	$M_{ij}$	$i$	$j$	$M_{ij}$	$i$	$j$	$M_{ij}$
<i>START</i>	<i>Reg</i>	< 2 >	<i>Reg</i>	<i>Online</i>	< 1 >	<i>Ab</i>	<i>Ccard</i>	< 1 >
<i>START</i>	<i>Acc</i>	< 2 >	<i>Reg</i>	<i>Fax</i>	< 1 >	<i>Ac</i>	<i>Ccard</i>	< 1 >
<i>START</i>	<i>Air</i>	< 2 >	<i>Hb</i>	<i>Ccard</i>	< 2 >	<i>Online</i>	<i>Ccard</i>	< 1 >
<i>Acc</i>	<i>Ha</i>	< 1 >	<i>Hb</i>	<i>Taxi</i>	< 2, 1 >	<i>Fax</i>	<i>Ccard</i>	< 1 >
<i>Acc</i>	<i>Hb</i>	< 1 >	<i>Hb</i>	<i>Bus</i>	< 2, 1 >	<i>Ha</i>	<i>Ccard</i>	< 1 >
<i>Acc</i>	<i>Hc</i>	< 1 >	<i>Hc</i>	<i>Ccard</i>	< 2 >	<i>Ha</i>	<i>Taxi</i>	< 1 >
<i>Air</i>	<i>Aa</i>	< 1 >	<i>Hc</i>	<i>Taxi</i>	< 2, 1 >	<i>Ccard</i>	<i>END</i>	< 1 >
<i>Air</i>	<i>Ab</i>	< 1 >	<i>Hc</i>	<i>Bus</i>	< 2, 1 >	<i>Taxi</i>	<i>END</i>	< 1 >
<i>Air</i>	<i>Ac</i>	< 1 >	<i>Aa</i>	<i>Ccard</i>	< 1 >	<i>Bus</i>	<i>END</i>	< 1 >

### 5.3.1 Trust Estimation Model

Since subjective probability is a person's degree of belief concerning a certain event [30, 33], the trust rating in  $[0, 1]$  of a service given by a service client can be taken as the *subjective possibility* with which the service provider can perform the service satisfactorily. Hence, *subjective probability theory* is the right tool for dealing with trust ratings. In this chapter, we adopt *Bayesian inference*, which is an important component in *subjective probability theory*, to estimate the trust value of a provided service from a set of ratings. Each rating is a value in  $[0, 1]$  evaluated from the subjective judgements of a service client.

The primary goal of *Bayesian inference* [30, 33] is to summarize the available information that defines the distribution of trust ratings through the specification of probability density functions, such as prior distribution and posterior distribution. The *prior distribution* summarizes the subjective information about the trust prior to obtaining the ratings sample  $x_1, x_2, \dots, x_n$ . Once the sample is obtained, the prior distribution can be updated. The updated probability distribution on trust ratings is called the *posterior distribution*, because it reflects probability beliefs posterior to analyzing ratings.

According to [35], if all service clients give ratings for the same service, the pro-

vided ratings conform to a normal distribution. The complete set of ratings can be collected based on honest-feedback-incentive mechanisms [42, 43]. Let  $\mu$  and  $\sigma$  denote the mean and the variance of ratings in the normal distribution. Thus, a sample of ratings  $x_1, x_2, \dots, x_n$  ( $x_i \in [0, 1]$ ) has the normal density with mean  $\mu$  and variance  $\sigma$ . In statistics, when a ratings sample with size  $n$  is drawn from a normal distribution with mean  $\mu$  and variance  $\sigma$ , the mean of the ratings sample also conforms to a normal distribution which has mean  $\mu$  and variance  $\sigma/\sqrt{n}$  [30]. Let  $\delta \in [0, 1]$  denote the prior subjective belief about the trust of a service that a client is requesting. We can assume that the prior normal distribution of  $\mu$  has mean  $\delta$  and variance  $\sigma/\sqrt{n}$ , i.e.

$$f(\mu) = \begin{cases} \frac{\sqrt{n}}{\sigma\sqrt{2\pi}} e^{-\frac{n(\mu-\delta)^2}{2\sigma^2}}, & 0 < \mu < 1; \\ 0, & \text{otherwise.} \end{cases} \quad (5.3)$$

Given  $\mu$ , the joint conditional density of the ratings sample is

$$f(x_1, x_2, \dots, x_n | \mu) = \frac{1}{\sigma^n (2\pi)^{\frac{n}{2}}} e^{-\frac{\sum (x_i - \mu)^2}{2\sigma^2}} = \frac{1}{\sigma^n (2\pi)^{\frac{n}{2}}} e^{-\frac{\sum x_i^2 - 2\mu \sum x_i + n\mu^2}{2\sigma^2}}. \quad (5.4)$$

Hence, the joint density of the ratings sample and  $\mu$  is

$$f(x_1, \dots, x_n; \mu) = \frac{\sqrt{n}}{\sigma^{n+1} (2\pi)^{\frac{n+1}{2}}} e^{-\frac{\sum x_i^2 - 2\mu n\bar{x} + n\mu^2 + n(\mu-\delta)^2}{2\sigma^2}}. \quad (5.5)$$

Based on Eq. (5.5), the marginal density of the ratings sample is

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= \frac{\sqrt{n}}{\sigma^{n+1} (2\pi)^{\frac{n+1}{2}}} e^{-\frac{\sum x_i^2 + n\delta^2}{2\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{n\mu^2 - (n\bar{x} + n\delta)\mu}{\sigma^2}} d\mu \\ &= \frac{\sqrt{n}}{\sigma^{n+1} (2\pi)^{\frac{n+1}{2}}} e^{-\frac{\sum x_i^2 + n\delta^2 - \frac{n(\bar{x} + \delta)^2}{2}}{2\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{n(\mu - \frac{\bar{x} + \delta}{2})^2}{\sigma^2}} d\mu \\ &= \frac{1}{\sqrt{2}\sigma^n (2\pi)^{\frac{n}{2}}} e^{-\frac{\sum x_i^2 + n\delta^2 - \frac{n(\bar{x} + \delta)^2}{2}}{2\sigma^2}}, \end{aligned} \quad (5.6)$$

since a normal density has to integrate to 1.

Thus, the posterior density for  $\mu$  is

$$f(\mu|x_1, x_2, \dots, x_n) = \frac{f(x_1, x_2, \dots, x_n; \mu)}{f(x_1, x_2, \dots, x_n)} = \frac{\sqrt{n}}{\sigma\sqrt{\pi}} e^{\frac{n(\mu - \frac{\bar{x} + \delta}{2})^2}{-\sigma^2}}. \quad (5.7)$$

Therefore, the posterior distribution of  $\mu$  is normal with mean  $\frac{\bar{x} + \delta}{2}$  and variance  $\sigma/\sqrt{2n}$ . If the loss function is estimated by squared error [30, 33], the mean of the posterior normal distribution can be used as the estimation of the trust value from ratings. Hence,

**Theorem 7:** The Bayesian estimation of the trust value of a service with  $n$  ratings  $x_1, x_2, \dots, x_n$  ( $x_i \in [0, 1]$ ) is

$$T(x_1, x_2, \dots, x_n, \delta) = \frac{\bar{x} + \delta}{2} = \frac{\sum_{i=1}^n x_i + n\delta}{2n}, \quad (5.8)$$

where  $\delta \in [0, 1]$  denotes the requesting client's prior subjective belief about the trust.

If the requesting client has no prior subjective information about the trust of the requested service, by default, let  $\delta = \frac{1}{2}$  since  $\frac{1}{2}$  is the middle point of  $[0, 1]$  representing the neutral belief between distrust and trust. After the Bayesian inference, the Bayesian estimation of trust can be taken as the requesting client's prior subjective belief about the trust for the Bayesian inference next time.

Now we can estimate the trust of a requested service by combining the requesting client's prior subjective belief about the trust and ratings. Since trust is subjective, it is more reasonable to include the requesting client's prior subjective belief about the trust in trust evaluation.

### 5.3.2 Global Trust Computation in Composite Services

Our goal is to select the optimal service execution flow (*SEF*) from multiple *SEFs* in a *SIG* aiming at maximizing the global trust value of *SEF*, which is determined by the trust values of the vertices and invocation relations between vertices in the *SEF*.

According to Definition 27, in *SEF* we only need consider Se (Fig. 5.1 (a)), Pa

(Fig. 5.1 (b)) and  $S_y$  (Fig. 5.1 (e)). With  $S_e$  and  $P_a$ ,  $S_y$  in a *SEF* can be determined. As a *SEF* is an end-to-end graph, if in the *SEF* there is a  $P_a$ , with which a service invokes its succeeding services in parallel, there must be a  $S_y$ , with which a service is activated by its preceding services in parallel.

Hence, there are two kinds of atomic structures to determine the trust value of a *SEF*:  $S_e$  and  $P_a$ .  $S_e$  in the *SEF* can be selected from the service invocation relation  $S_e$  (Fig. 5.1(a)) or  $P_r$  (Fig. 5.1(c)) in the *SIG*.  $P_a$  in the *SEF* can be selected from the service invocation relation  $P_a$  (Fig. 5.1 (b)) in the *SIG*.

**Definition 29:** The global trust value  $T_g$  of an  $S_e$  structure where service  $S$  uniquely invokes service  $A$  (see Fig. 5.1 (a)) can be computed by

$$T_g = T_S \cdot T_A, \quad (5.9)$$

where  $T_S$  and  $T_A$  are the trust values of  $S$  and  $A$  respectively, which are evaluated from Theorem 7. As  $S$  and  $A$  are independent, the probability that  $S$  and  $A$  both occur is equal to the product of the probability that  $S$  occurs and the probability that  $A$  occurs.

**Definition 30:** The global trust value  $T_g$  of a  $P_a$  structure where service  $S$  invokes services  $A$  and  $B$  in parallel (see Fig. 5.1 (b)) can be computed from  $T_S$  and the combined trust value  $T_{AB}$  by Definition 29, and

$$T_{AB} = \frac{\omega_{CCS_A}}{\omega_{CCS_A} + \omega_{CCS_B}} \cdot T_A + \frac{\omega_{CCS_B}}{\omega_{CCS_A} + \omega_{CCS_B}} \cdot T_B, \quad (5.10)$$

where  $T_S$ ,  $T_A$  and  $T_B$  are the trust values of  $S$ ,  $A$  and  $B$  respectively, which are evaluated from Theorem 7.  $\omega_{CCS_A}$  and  $\omega_{CCS_B}$  are weights for  $A$  and  $B$  respectively, which are specified in a requesting client's preference or specified as the default value by the service trust management authority.

According to Definitions 29 & 30, each atomic structure  $S_e$  or  $P_a$  can be converted to a single vertex. Hence, in the process of trust computation, a *SEF* consisting of

Se and Pa structures can be incrementally converted to a single vertex with its trust value computed as the global trust. Therefore, the global trust evaluation of the *SEF* algorithm has the following steps (For details, please refer to [52]):

**Step 1** Firstly the trust value of each atomic Se structure in the *SEF* can be computed by using Definition 29. Each computed atomic Se structure is then taken as a vertex in the *SEF*.

**Step 2** After that, the trust value of each atomic Pa structure is computed by using Definition 30. Similarly, each computed atomic Pa structure is then taken as a vertex in the *SEF*.

**Step 3** The computations in Steps 1 & 2 repeat until the final *SEF* is simplified as a vertex, and the global trust value is obtained.

The details of global trust evaluation of *SEF* are illustrated in Algorithm 7.

## 5.4 Trust-Oriented Composite Service Selection

Here we assume that a service trust management authority stores a large volume of services with their ratings. In response to a client's request, the service trust management authority first generates a *SIG* containing all relevant services and invocation relations. Then, the trust-oriented service selection and discovery algorithm is applied to find the optimal *SEF* with the maximized global trust value.

### 5.4.1 Longest *SEF* Algorithm

If there are only Pr (probabilistic invocation) structures in a *SIG* (i.e. there are only Se (sequential invocation) structures in the *SEF*), the *SEF* is a path in the *SIG*. In this case, the longest *SEF* algorithm is applied when searching for the optimal *SEF* with the largest trust value.

**Algorithm 7** Global Trust Evaluation Algorithm of *SEF*


---

**Input:** a *SEF*, trust value for each vertex.  
**Output:** the global trust value of *SEF*  $T_{global}$ .

- 1: let the starting service of *SEF* be *root*, and the ending service of *SEF* be *terminal*;
- 2: **while** there is more than one vertices in *SEF* **do**
- 3:   initialize vector *Container* to contain *root*;
- 4:   **while** *Container*  $\neq \emptyset$  **do**
- 5:     select a vertex *v* in *Container*;
- 6:     remove *v* from *Container*;
- 7:     let vectors *Se* and *Pa* be the *Se* and *Pa* structures from *v*;
- 8:     **if** vector *Se*  $\neq \emptyset$  **then**
- 9:       **if** only *v* invokes *Se* **then**
- 10:          // global trust evaluation of *Se* (lines 11-17)
- 11:          let *vSe* be the vertex which is merged from *v* and *Se*;
- 12:          let the predecessors of *v* be those of *vSe*;
- 13:          let the successors of *Se* be those of *vSe*;
- 14:          remove all the edges to *v* in *SEF*;
- 15:          remove all the edges from *Se* in *SEF*;
- 16:          let the weight of *v* be that of *vSe*;
- 17:          let  $T_{vSe}$  be the trust value of *vSe* based on Definition 29;
- 18:           $T_{global} \leftarrow T_{vSe}$
- 19:          add *vSe* into *Container*;
- 20:       **else**
- 21:          **if** *Se* is not *terminal* and *Se* is not in *Container* **then**
- 22:            add *Se* into *Container*;
- 23:          **end if**
- 24:       **end if**
- 25:     **end if**
- 26:     **if** vector *Pa*  $\neq \emptyset$  **then**
- 27:       **for all**  $Pa(i)$  in *Pa* **do**
- 28:          **if**  $Pa(i)$  is not *terminal* and  $Pa(i)$  is not in *Container* **then**
- 29:            add  $Pa(i)$  into *Container*;
- 30:          **end if**
- 31:       **end for**
- 32:       **for all**  $Pa(i)$  in *Pa* **do**
- 33:          let  $Se_i$  and  $Pa_i$  be the *Se* and *Pa* structures from  $Pa(i)$ ;
- 34:          **for all**  $Pa(j)$  in *Pa* and  $j > i$  **do**
- 35:            let  $Se_j$  and  $Pa_j$  be the *Se* and *Pa* structures from  $Pa(j)$ ;
- 36:            **if**  $Se_i = Se_j$  and  $Pa_i = \emptyset$  and  $Pa_j = \emptyset$  **then**
- 37:              // global trust evaluation of *Pa* (lines 38-44)
- 38:              let *vPa* be the vertex merged from  $Pa(i)$  and  $Pa(j)$ ;
- 39:              let the successors of  $Se_i$  be those of *vPa*;
- 40:              let the predecessors of  $Pa(i)$  and  $Pa(j)$  be those of *vPa*;
- 41:              remove all the edges from *v* to  $Pa(i)$  and  $Pa(j)$ ;
- 42:              remove all the edges from  $Pa(i)$  and  $Pa(j)$  to  $Se_i$ ;
- 43:              let the sum of weights of  $Pa(i)$  and  $Pa(j)$  be that of *vPa*;
- 44:              let  $T_{vPa}$  be the trust value of *vPa* based on Definition 30;
- 45:               $T_{global} \leftarrow T_{vPa}$
- 46:              **if**  $Pa_i$  or  $Pa_j$  is in *Container* **then**
- 47:                remove  $Pa_i$  or  $Pa_j$  from *Container*;
- 48:              **end if**
- 49:            **end if**
- 50:          **end for**
- 51:       **end for**
- 52:     **end if**
- 53:   **end while**
- 54: **end while**
- 55: **return**  $T_{global}$

---

By extending Dijkstra's shortest path algorithm [21], the longest *SEF* algorithm is used to find an execution flow (path) from *START* to *END* so that the multiplication of

trust values of all vertices in the path is maximal according to Definition 29.

Formally, given a weighted graph consisting of set  $V$  of vertices and set  $E$  of edges, find a flow (path)  $P$  from the service invocation root  $START \in V$  to the service invocation terminal  $END \in V$  so that

$$\prod_{v_j \in P, v_j \in V} (T(x_1(v_j), x_2(v_j), \dots, x_n(v_j), \delta_j)) \quad (5.11)$$

is the maximal among all flows (paths) from  $START$  to  $END$ , where  $x_i(v_j)$  denotes a rating for vertex  $v_j$  and  $\delta_j$  denotes the requesting client's prior subjective belief about the trust of vertex  $v_j$ .

#### 5.4.2 Monte Carlo Method Based Algorithm (MCBA)

If there are only Pa structures in a *SIG*, the unique *SEF* is the same as the *SIG*.

If a *SIG* consists of both Prs and Pas, since there is no existing method to consider the kind of structure we have analyzed in Section 5.1, we propose using a *Monte Carlo method based algorithm (MCBA)* to find the optimal *SEF*.

The Monte Carlo method [24] is a computational algorithm which relies on repeated random sampling to compute results. It tends to be adopted when it is infeasible to compute an exact result using a deterministic algorithm. The Monte Carlo method is useful for modeling phenomena with significant uncertainty in inputs, such as the calculation of risk in business [24]. The specific areas of application of the Monte Carlo method include computational physics, physical chemistry, global illumination computations, finance and business, and computational mathematics (e.g. numerical integration and numerical optimization) [24, 71]. It is also one of the techniques for solving NP-complete problems [24, 71]. Generally, the Monte Carlo method consists of four steps: (1) defining a domain of inputs, (2) generating inputs randomly, (3) performing a computation on each input, and (4) aggregating the results into the final result.

The main strategy in *MCBA* is as follows. In a *SIG*, the direct successors of a service need to be selected according to their trust values. Usually, the direct successor with a larger trust value is preferred, which indicates higher probability to be invoked, and vice versa. Then, according to this, a uniform distributed random number is generated to decide which succeeding service is selected.

When determining the optimal *SEF* from a *SIG*, we only need *MCBA* for Pr structures. Let's take Pr in Fig. 5.1(c) as an example to explain the details of our *MCBA*. If successor *A* has a trust value  $T_A$  from Theorem 7 and successor *B* has a trust value  $T_B$  from Theorem 7, the probability for vertex *S* to choose successor *A* is

$$P_A = \frac{T_A}{T_A + T_B}. \quad (5.12)$$

Similarly, the probability to choose successor *B* is

$$P_B = \frac{T_B}{T_A + T_B}. \quad (5.13)$$

Obviously,  $0 < P_A, P_B < 1$ . A uniform distributed random number  $r_0$  in  $(0, 1)$  is then generated to decide which successor is chosen. In detail, if  $r_0 < P_A$ , successor *A* is chosen; if  $P_A < r_0 < P_A + P_B = 1$ , successor *B* is chosen.

Therefore, given a *SIG*, a *SEF* could be obtained by repeating the *MCBA* from the service invocation root *START* until the service invocation terminal *END* is reached. Once a *SEF* is generated, its global trust value can be calculated by using the global trust computation algorithm proposed in Section 5.3.2. By repeating this process for  $l$  simulation times, a set of *SEFs* can be generated, from which the locally optimal *SEF* with the maximal global trust value can be obtained. A high value of  $l$  is necessary to obtain the optimal solution. The *MCBA* for composite service selection and discovery is illustrated in Algorithm 8.

In Theorem 7, the trust estimation algorithm has a complexity of  $O(n)$  with  $n$  ratings. Hence, in global trust computation algorithm in Section 5.3.2, the complexity

**Algorithm 8** MCBA for Composite Service Selection and Discovery**Input:** Simulation times  $l$ ;  $SIM$ , and service ratings  $Reputation$ .**Output:** The optimal  $SEF$  with maximum global trust value  $Trust_{global}$ .

```

1: let  $Trust$  be the trust value for each service evaluated from  $Reputation$  by Theorem 7;
2: for all  $i$  such that  $1 \leq i \leq l$  do
3:   initialize  $active = [root]$ ,  $SEF = [root]$ ;
4:   while  $active \neq \emptyset$  do
5:     select a vertex  $vertex$  from  $active$ , and remove  $vertex$  from  $active$ ;
6:     let vectors  $Pr$  and  $Pa$  be the  $Pr$  and  $Pa$  structures from  $vertex$ ;
7:     if vector  $Pa \neq \emptyset$  then
8:       if  $vertex$  is in  $SEF$  then
9:         for all  $Pa(j)$  in  $Pa$  do
10:          if  $Pa(j)$  is not in  $SEF$  then
11:            add  $Pa(j)$  into  $SEF$ 
12:          end if
13:        end for
14:      end if
15:      for all  $Pa(j')$  in  $Pa(j)$  do
16:        if  $Pa(j')$  is not terminal and  $Pa(j')$  is not in  $active$  then
17:          add  $Pa(j')$  into  $active$ 
18:        end if
19:      end for
20:    end if
21:    if vector  $Pr \neq \emptyset$  then
22:      if  $vertex$  is in  $SEF$  then
23:        if none of  $Pr$  is in  $SEF$  then
24:          for all  $Pr(k)$  in  $Pr$  do
25:            generate a uniform distributed random number  $rand$  in  $[0, 1]$ ;
26:            select the smallest  $k'$  such that  $rand < Trust(k')/sum(Trust(k))$ 
27:          end for
28:          add  $Pr(k')$  in  $SEF$ 
29:        end if
30:      end if
31:      if  $Pr(k')$  is not terminal and  $Pr(k')$  is not in  $active$  then
32:        add  $Pr(k')$  into  $active$ 
33:      end if
34:    end if
35:  end while
36:  let  $Trust_{SEF}$  be the trust value of  $SEF$  according to Global Trust Computation Algorithm
37:   $Trust_{global} = \max Trust_{SEF}$ ;
38: end for
39: return optimal  $SEF$  and  $Trust_{global}$ .

```

of trust evaluation for a composite service with  $N$  services is  $O(nN)$ . Therefore, MCBA with  $l$  simulations incurs a complexity of  $O(nlN)$ .

**Table 5.2:** Ratings and subjective belief of each service component in the travel plan

	<i>Reg</i>	<i>Acc</i>	<i>Air</i>	<i>Online</i>	<i>Fax</i>	<i>Ha</i>	<i>Hb</i>
$x_1$	0.88	0.83	0.78	0.92	0.51	0.17	0.35
$x_2$	0.84	0.82	0.87	0.92	0.38	0.18	0.32
$x_3$	0.97	0.85	0.77	0.94	0.25	0.22	0.46
$x_4$	0.87	0.82	0.83	0.96	0.40	0.12	0.34
$x_5$	0.91	0.74	0.79	0.95	0.41	0.16	0.28
$\delta$	0.92	0.85	0.91	0.95	0.32	0.20	0.50
	<i>Hc</i>	<i>Aa</i>	<i>Ab</i>	<i>Ac</i>	<i>Ccard</i>	<i>Taxi</i>	<i>Bus</i>
$x_1$	0.89	0.30	0.95	0.25	0.95	0.94	0.32
$x_2$	0.86	0.36	0.98	0.30	0.95	0.86	0.37
$x_3$	0.82	0.34	0.91	0.24	0.96	0.86	0.34
$x_4$	0.87	0.29	0.91	0.31	0.96	0.89	0.18
$x_5$	0.88	0.41	0.97	0.29	0.96	0.90	0.35
$\delta$	0.91	0.32	0.92	0.51	0.98	0.89	0.33

**Table 5.3:** Weights of service components in Pa

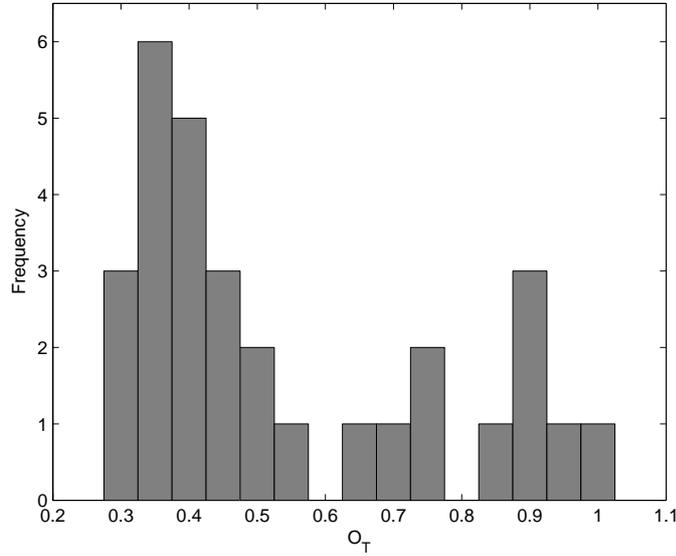
<i>Reg</i>	<i>Acc</i>	<i>Air</i>	<i>Ccard</i>	<i>Taxi</i>	<i>Ccard</i>	<i>Bus</i>
0.1	0.3	0.6	0.6	0.4	0.6	0.4

### 5.4.3 Experiments on Trust-Oriented Composite Service Selection

In this section, we will illustrate the results of conducted experiments for studying our proposed *MCBA*.

#### 5.4.3.1 Comparison on Travel Plan Composite Services

In this experiment, we compare our proposed *MCBA* with the exhaustive search method by applying them to the travel plan composite services (with 16 vertices and 30 *SEFs*). The corresponding ratings and Smith's prior subjective belief regarding each service component are listed in Table 5.2. The weights of service components in all Pa structures of the composite services are listed in Table 5.3.



**Figure 5.5:** Histogram of  $O_T$  for each  $SEF$

The exhaustive search method is inefficient as it aims to enumerate all solutions. In Menascé's work [69], the exhaustive search method is adopted to calculate the execution time and cost of all  $SEFs$  in a composite service.

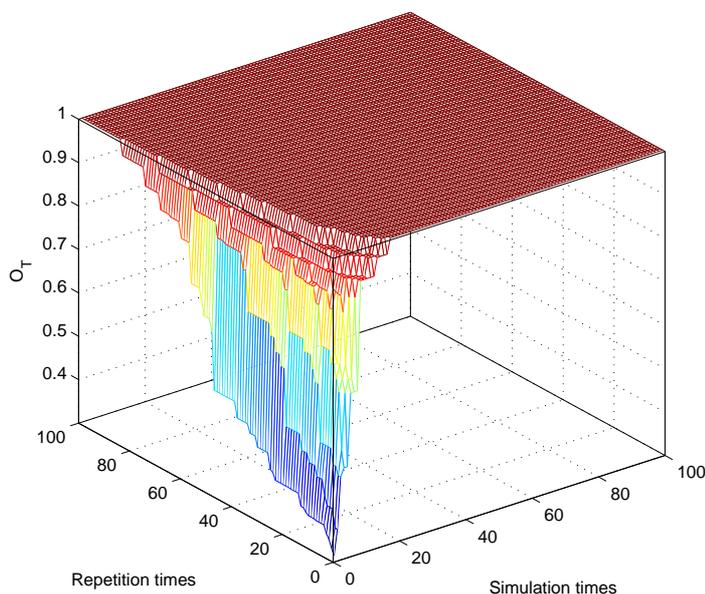
According to the global trust computation algorithm in Section 5.3.2, the global trust value  $T_i$  of  $SEF$   $i$  ( $i = 1, 2, \dots, 30$ ) can be calculated.

**Definition 31:** With the global trust value  $T_i$  of  $SEF$   $i$ , let *trust-based SEF optimality* be

$$O_T(T_i) = \frac{T_i}{\max(T_i)}. \quad (5.14)$$

In the corresponding histogram, the  $O_T(T_i)$  values of 30  $SEFs$  are plotted in Fig. 5.5. From this, we can observe that 80% of  $O_T(T_i)$  values are less than 0.8, implying that if we choose a  $SEF$  randomly, it is very likely that we will obtain a  $SEF$  with a low trust value .

In the  $MCBA$ , there are multiple simulations. In each of these, a  $SEF$  is generated and its global trust value is calculated. After  $l$  simulations, a locally optimal  $SEF$  can

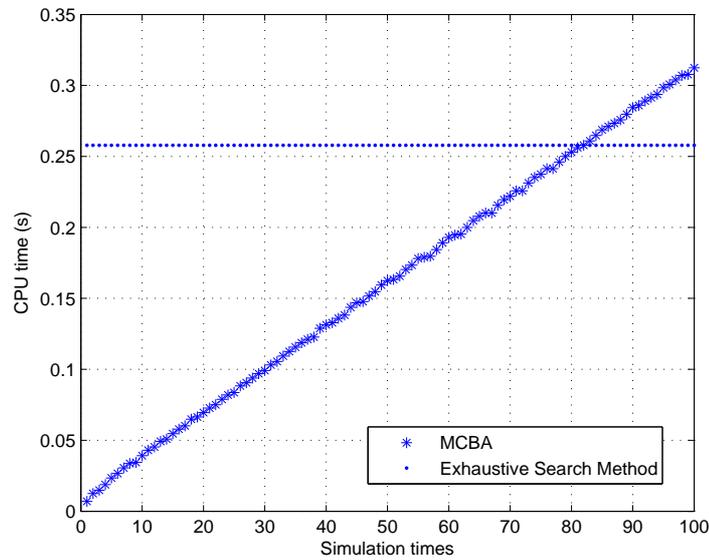


**Figure 5.6:**  $O_T$  in the travel plan example

be obtained from  $l$  generated *SEFs*. In order to study the distribution of the global trust of locally optimal *SEFs*, we take  $l$  simulations as a repetition and repeat for  $m$  times.

Our experiments use Matlab 7.6.0.324 (R2008a) running on a Dell Vostro V1310 laptop with an Intel Core 2 Duo T5870 2.00GHz CPU and a 3GB RAM.  $l$ , the number of simulation times, is set from 1 to 100.  $m$ , the number of repetition times, is set from 1 to 100. The experiment results are plotted in Fig. 5.6. We could observe that with a fixed number of repetitions, the more simulations, the closer to 1  $O_T$  becomes. More simulations thus lead to a higher probability that the optimal *SEF* will be obtained.

Furthermore, we compare the execution time of the *MCBA* with that of the exhaustive search method. Each CPU time in this chapter is the average of ten independent executions. In Fig. 5.7, we can observe that when the number of simulation times is  $l \leq 82$ , our *MCBA* is faster than the exhaustive search method. From Figs 5.7 and 5.6, we can see that the probability of obtaining the optimal *SEF* is 97% when there are 20 simulations, whereas the execution time of our *MCBA* is 27% of that of the exhaustive



**Figure 5.7:** CPU time of simulation times

search method. According to Table 5.2, theoretically the probability of obtaining the optimal *SEF* for each simulation in the *MCBA* is 17.8%, due to *SIG* and the strategy in the *MCBA* proposed in Section 5.4.2. Hence, after 20 simulations, theoretically the *MCBA* has the probability of 98.04% of obtaining the optimal *SEF*. Hence, the experiment result regarding the probability of obtaining the optimal *SEF* confirms the theoretical conclusion.

With this simple travel plan example, the *MCBA* outperforms the exhaustive search method. More significant performance differences can be observed with some complex composite services, which will be introduced in the next section.

#### 5.4.3.2 Comparison of Complex Composite Services

In this experiment, we further compare our proposed Monte Carlo method based algorithm (*MCBA*) with the exhaustive search method on three more complex composite services. The number of vertices of these composite services is 35, 52 and 100 respectively. The numbers of *Ses*, *Pas*, *Prs*, *Sys*, *Ass* and *SEFs* in the corresponding

**Table 5.4:** Structure of complex composite services

Number of vertices	Ses	Pas	Prs	Sys	Ass	<i>SEFs</i>
35	17	8	11	4	11	$1.8 \times 10^3$
52	24	13	16	7	16	$5.4 \times 10^4$
100	51	24	32	12	32	$2.92 \times 10^9$

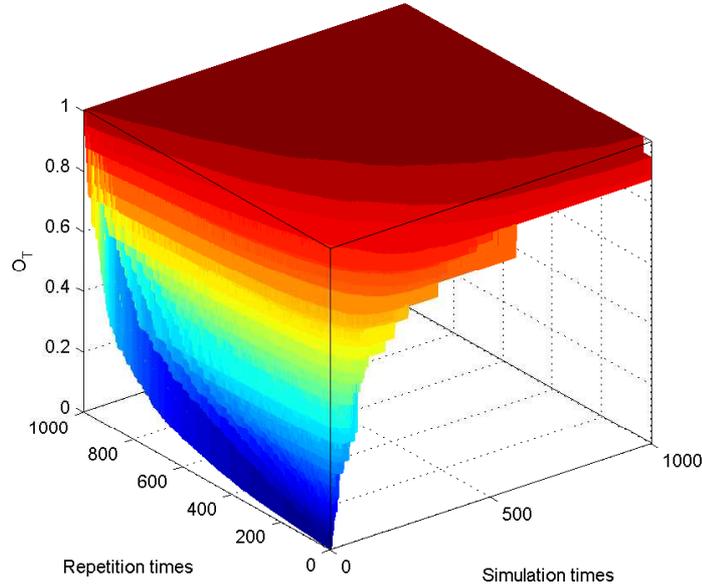
**Table 5.5:** CPU time in seconds of different examples

Number of vertices	16	35	52	100
Probability to obtain the optimal <i>SEF</i> for each simulation	17.84%	14.31%	5.71%	0.33%
Number of simulation times in <i>MCBA</i>	20	20	52	925
Probability to obtain the optimal <i>SEF</i> for <i>MCBA</i>	98.04%	95.45%	95.29%	95.12%
CPU time (seconds) of <i>MCBA</i>	0.0695	0.3219	0.8625	34.51
CPU time (seconds) of exhaustive search method	0.2578	17.09	–	–

composite services are listed in Table 5.4.

In this experiment, we use the same platform as the experiment in Section 5.4.3.1. This produces the following results, which are also listed in Table 5.5.

1. In the case of composite service with 35 vertices, the *MCBA* takes 0.3219 of a second to finish 20 simulations with the probability of 95.45% of obtaining the optimal *SEF*, while the exhaustive search method takes 17.09 seconds.
2. When the number of vertices becomes 52, our *MCBA* takes 0.8625 of a second to finish 52 simulations, with which the probability of obtaining the optimal *SEF* is 95.29%. However, over the same time period, the exhaustive search method can only search 0.42% of  $5.4 \times 10^4$  *SEFs*.
3. When taking 1000 times of the *MCBA* CPU time, this time can only search approximately 1% of all *SEFs*. We further apply our *MCBA* to a composite service with 100 vertices. It takes 34.51 seconds to finish 925 simulations with a probability of 95.12% of obtaining the optimal *SEF*. In contrast, when taking



**Figure 5.8:**  $O_T$  in composite service of 100 vertices

the same time, the exhaustive search method can only search  $(9.56 \times 10^{-6})\%$  of  $2.92 \times 10^9$  *SEFs*. When taking 100 times of the *MCBA* CPU time, it can only search  $(1.01 \times 10^{-5})\%$  of all *SEFs*.

In the case of a composite service with 100 vertices, the results of the *MCBA* are plotted in Fig. 5.8. When there are  $l = 925$  simulation times, the *MCBA* can reach the optimal solution with a probability of 95.2%. Also it has a great chance to obtain the near-optimal solution, even when  $l$  is as small as 200. For example, in Fig. 5.8, when  $l$  is 200 the probability for the trust-based *SEF* optimality to be  $O_T \geq 0.82$  is about 95.7%.

In summary, our proposed *MCBA* can obtain a near-optimal *SEF* after some simulations. As the CPU time for a single simulation in *MCBA* is extremely short, our experiments have illustrated that the overall performance of the *MCBA* is good even with complex composite services. In addition, *MCBA* is suitable for parallel computing since each simulation in *MCBA* is independent. This can greatly speed up computa-

tions and shorten the overall CPU time. Thus, our proposed *MCBA* is effective and efficient.

## 5.5 Trust-Oriented Composite Service Selection with QoS Constraints

In SOC environments, Quality of Service (QoS) is essential when a set of quality metrics have to be achieved during service provision. These non-functional metrics should be measurable and constitute a description of what a service can offer. The QoS of IT services is often expressed in terms of capacity, latency, bandwidth, number of service requests, number of incidents, etc. When a client looks for a service from a large set of services offered by different providers, in addition to functionality and trust, QoS is a key factor for service selection.

In SOC environments, to satisfy the specified functionality and QoS requirements, a service may invoke other services forming a composite service with complex invocations and trust dependencies among its component services [69]. Meanwhile, given a set of various services, different compositions may lead to different service structures. Although these certainly enrich service provision, they greatly increase computation complexity and thus make trustworthy service selection with QoS constraints a very challenging task.

Taking trust evaluation and the complex structure of composite services into account, effective algorithms are needed for trust-oriented composite service selection with QoS constraints, and are expected to be more efficient than the existing approaches [69, 101].

Here we assume that a service trust management authority stores a large volume of services with their ratings. In response to a client's request, the service trust management authority first generates a *SIG* containing all relevant services and invocation relations. Then, the trust-oriented QoS constrained service selection algorithm is ap-

plied to find the most trustworthy *SEF* satisfying QoS constraints.

### 5.5.1 Related Work on Composite Service Selection with QoS Constraints

With different kinds of invocations, including parallel invocation, composite service selection with QoS constraints can be modeled as the Multi-Constrained Optimal Path (MCOP) problem, and several algorithms have been proposed to process the MCOP selection. In [69], an exhaustive search method is adopted to measure service execution time and cost involving probabilistic, parallel, sequential and fastest-predecessor-triggered invocations. However, its algorithm complexity is exponential. In [46], the H-MCOP algorithm is proposed to select the multi-constrained optimal path with the utility function

$$g_\lambda(p) = \sum_{i=1}^m \left( \frac{q_i(p)}{Q_i} \right)^\lambda, \quad (5.15)$$

where  $\lambda \geq 1$ ;  $q_i(p)$  is the aggregated value of the  $i^{th}$  QoS attribute of path  $p$ ;  $Q_i$  is the  $i^{th}$  QoS constraint of path  $p$ . This algorithm adopts both the backward and the forward Dijkstra's algorithm [21] in optimal path selection. In [101], the MCSP\_K algorithm is proposed to process the QoS-driven composite service selection. By taking the utility function

$$\xi(p) = \max\left\{ \left( \frac{q_i(p)}{Q_i} \right) \right\}, \quad (5.16)$$

this algorithm keeps the paths with up to  $K$  minimum  $\xi$  values at each intermediate service component, i.e. it keeps only  $K$  paths from the service invocation root to each intermediate service component. This  $K$ -path selection strategy aims to reduce the searching space and thus avoid excessive overhead in obtaining the near-optimal solution. Nevertheless, none of these works address any aspect of trust.

## 5.5.2 QoS Constrained Monte Carlo Method Based Algorithm

The trust-oriented composite service selection with QoS constraints can be modeled as the Multi-Constrained Optimal Path (MCOP) problem, which is an NP-complete problem [46, 101].

In composite services, each service component can be associated with multiple QoS attributes, which can be roughly classified as additive or non-additive [46].

- The aggregated value of a *SEF* with respect to an additive QoS attribute, such as delay, cost, execution time, etc, is given by the sum of the QoS values of service components along that *SEF* [69]. In addition, multiplicative constraints, such as reliability, can be transformed into additive constraints [46].
- In contrast, for non-additive QoS attributes (e.g. bandwidth), the aggregated value of a *SEF* is determined by the value of that QoS attribute at the bottleneck.

It is known that constraints associated with non-additive QoS attributes can be easily dealt with by using a preprocessing step, pruning away all service components that do not satisfy these constraints to simplify the structure of the composite services [46].

Therefore, in this chapter we will mainly focus on additive QoS attributes and assume that composite service selection with QoS constraints is only based on additive QoS attributes.

Selecting the optimal *SEF* with QoS constraints is a NP-complete problem [46, 101]. For this problem, we propose a *QoS constrained Monte Carlo method based algorithm (QC-MCBA)* to find the most trustworthy *SEF* satisfying QoS constraints.

The main strategy in *QC-MCBA* is as follows. In a *SIG*, the direct successor of a service needs to be selected according to the values of the utility function defined by

$$U_{QoS}(X) = \begin{cases} \omega_T \cdot T(X) + \sum_{i=1}^{m_{QoS}} \left( \frac{Q_i - q_i(X)}{Q_i} \right)^{\omega_{QoS}}, & \forall Q_i \geq q_i(X) \\ 0, & \forall Q_i < q_i(X) \end{cases}, \quad (5.17)$$

where  $\omega_T$  and  $\omega_{QoS}$  ( $\omega_{QoS} \geq 1$ ) are the weights for trust and all QoS attributes respectively specified in a requesting client's preference or specified as default values by the service trust management authority;  $T(X)$  is the trust value of direct successor  $X$  computed following Theorem 7;  $q_i(X)$  is the aggregated value of the  $i^{th}$  QoS attribute about  $SEF'$ , which is part of the  $SEF$  from the service invocation root to service component  $X$ ;  $Q_i$  is the  $i^{th}$  QoS constraint and  $m_{QoS}$  is the total number of QoS constraints.

In  $QC\_MCBA$ , the direct successor with a larger utility value is preferred, which indicates that a higher probability will be invoked. Then, according to this, a uniform distributed random number is generated to decide which succeeding service is selected. When determining the optimal  $SEF$  with QoS constraints from a  $SIG$ , we only need to use  $QC\_MCBA$  for Pr structures. Let's take the Pr structure in Fig. 5.1(c) as an example to explain the details of  $QC\_MCBA$ . If successor  $A$  has the utility value  $U_{QoS}(A)$  and successor  $B$  has the utility value  $U_{QoS}(B)$ , the probability for vertex  $S$  to select successor  $A$  is

$$P_A = \frac{U_{QoS}(A)}{U_{QoS}(A) + U_{QoS}(B)}. \quad (5.18)$$

Similarly, the probability to select successor  $B$  is

$$P_B = \frac{U_{QoS}(B)}{U_{QoS}(A) + U_{QoS}(B)}. \quad (5.19)$$

Obviously,  $0 < P_A, P_B < 1$ . Then, a uniform distributed random number  $r_0$  in  $(0, 1)$  is generated to decide which successor is selected. In detail, if  $r_0 < P_A$ , successor  $A$  is selected; if  $P_A < r_0 < P_A + P_B = 1$ , successor  $B$  is selected.

Therefore, given a  $SIG$ , a feasible  $SEF$  satisfying QoS constraints could be obtained by repeating  $QC\_MCBA$  from the  $START$  until  $END$  is reached. Once a feasible  $SEF$  is generated, its global trust value can be calculated by the global trust computation algorithm in Section 5.3.2. By repeating this process for  $l$  simulation times, a set of feasible  $SEFs$  can be generated from which the locally optimal QoS constrained  $SEF$  with the maximal global trust value can be obtained. The value of  $l$  determines

the performance and overhead of *QC\_MCBA*. If  $l$  is large enough, this algorithm can obtain the optimal solution, but its computational cost will be very high.

Our proposed *MCBA* & *QC\_MCBA* are not designed to consider all *SEF*'s in composite services. If we know the information of service components (e.g. trust values and QoS values), after  $l$  simulation times, a set of feasible *SEF*'s with better trust values are generated, from which the locally optimal *SEF* can be obtained. Therefore, the selection process in *MCBA* & *QC\_MCBA* is performed at run time rather than design time, making our proposed method practical in applications.

### 5.5.3 Experiment on Trust-Oriented Composite Service Selection with QoS Constraints

In this experiment, we compare our proposed *QC\_MCBA* with the exhaustive search method by applying it to the composite services listed in Section 5.4.3. Meanwhile, we adopt the same platform as the one used in Section 5.4.3 as well.

Firstly, we focus on the travel plan composite services. In this experiment, only two kinds of QoS attributes of each service component are taken into account: cost and execution time. In order to adopt *QC\_MCBA*, it is necessary to compute  $q_i(X)$  used in Eq. (5.17), i.e. the aggregated value of the  $i^{th}$  QoS attribute about *SEF'*, which is the part of *SEF* from the service invocation root to service component  $X$ . The aggregated value of cost is just the summation of the cost of each service component in *SEF'*, i.e.

$$q_{cost}(X) = \sum_{Y \in SEF'} c_Y, \quad (5.20)$$

where  $c_Y$  is the cost of service component  $Y$ . If there are only Se (sequential invocation) structures in the *SEF'*, there is no difference between cost aggregation and execution time aggregation. However, if Pa structures are involved in the *SEF'*, we need to pay extra attention to the aggregation of execution time. We use service component *Ccard* in the *SEF* of Fig. 5.4 as an example to illustrate the aggregation of

**Table 5.6:** QoS attribute values of each service component in the travel plan example

	<i>START</i>	<i>Reg</i>	<i>Acc</i>	<i>Air</i>	<i>Online</i>	<i>Fax</i>	<i>Ha</i>	<i>Hb</i>
cost	0	50	20	50	800	800	1100	1200
execution time	100	80	160	100	30	300	150	160
	<i>Hc</i>	<i>Aa</i>	<i>Ab</i>	<i>Ac</i>	<i>Ccard</i>	<i>Taxi</i>	<i>Bus</i>	<i>END</i>
cost	1000	2100	2000	2200	50	120	80	0
execution time	150	220	200	210	100	180	80	10

execution time.

$$q_{time}(Ccard) = t_{START} + \max\{t_{Reg} + t_{Online}, t_{Acc} + t_{Ha}, t_{Air} + t_{Aa}\} + t_{Ccard}, \quad (5.21)$$

where  $t_X$  is the execution time of service component  $X$ . Hence, we can extend Dijkstra's shortest path algorithm [21] to find the aggregated execution time, which is the longest path in the  $SEF'$ .

Corresponding QoS attribute values of each service component are listed in Table 5.6. We set  $Q_{cost} = 4400$ ,  $Q_{time} = 605$ ,  $\omega_T = 1$  and  $\omega_{QoS} = 2$ .

**Definition 32:** With the global trust value  $T_i$  of  $SEF$   $i$  ( $i = 1, 2, \dots, 30$ ), let us define the *trust-based QoS constrained SEF optimality*

$$O_{T_{QoS}}(T_i) = \begin{cases} \frac{T_i}{\max(T_i)}, & \text{if it satisfies all QoS constraints,} \\ 0, & \text{otherwise,} \end{cases} \quad (5.22)$$

The corresponding histogram of  $O_{T_{QoS}}(T_i)$  values of 30  $SEF$ s is plotted in Fig. 5.9. From it, we can observe that 86.7% of  $O_{T_{QoS}}(T_i)$  values are less than 0.8, implying that if we select a  $SEF$  randomly, we will very likely obtain a  $SEF$  with a low trust value or a  $SEF$  which does not satisfy QoS constraints. With simulation times  $1 \leq l \leq 100$  and repetition times  $1 \leq m \leq 100$ , the experimental results of  $QC$   $MCBA$  are plotted in Fig. 5.11. As for the CPU time, in Fig. 5.10 we can observe that with the number of

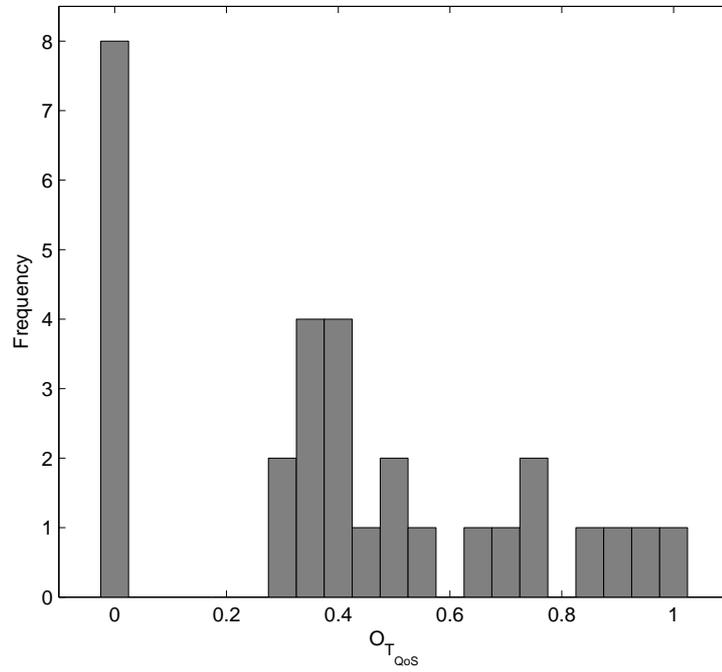


Figure 5.9: Histogram of  $O_{T_{QoS}}$  for each *SEF* in the travel plan example

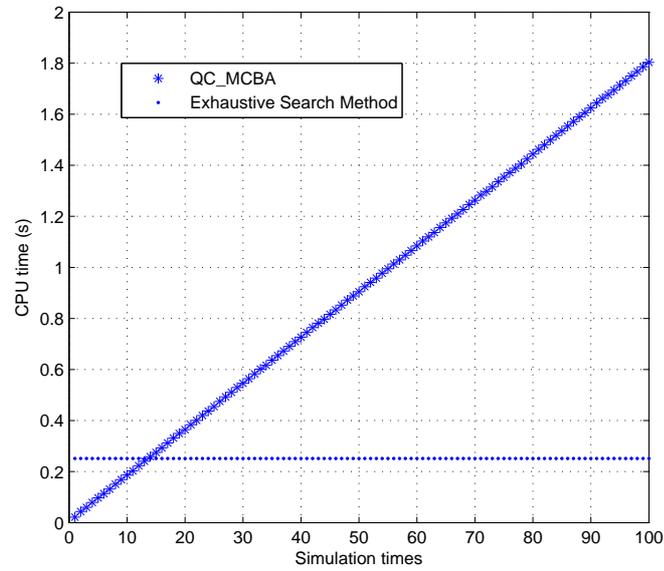
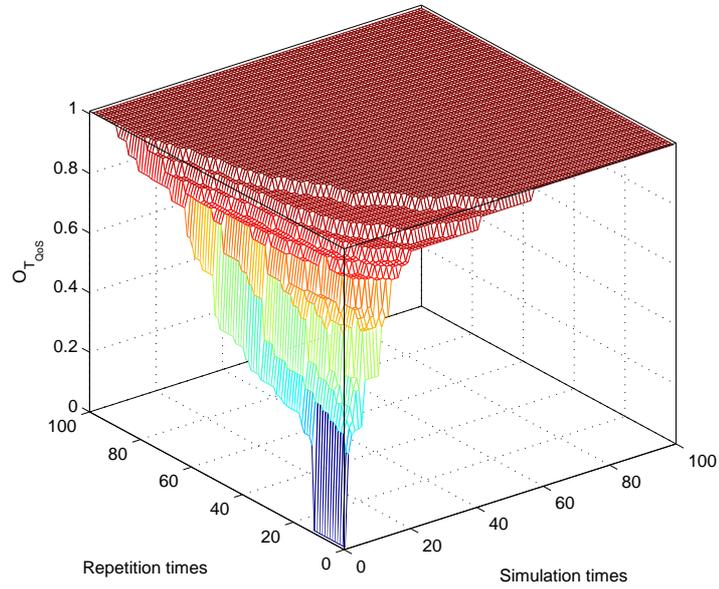
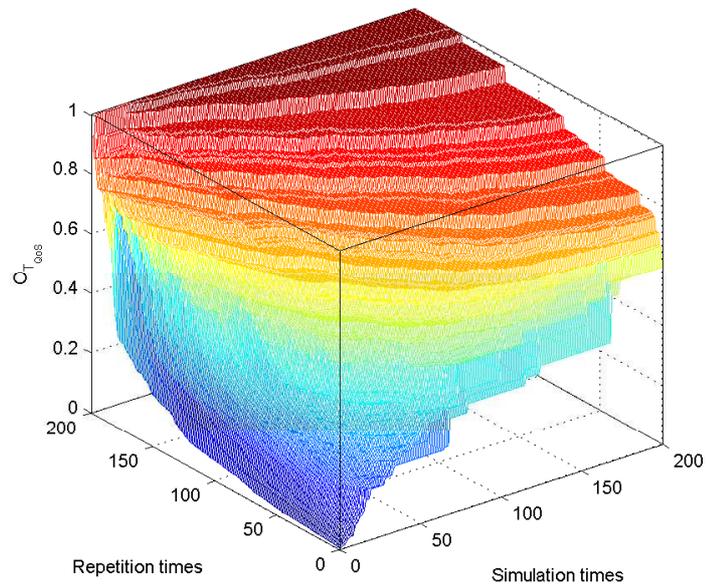


Figure 5.10: CPU time of *QC\_MCBA* in the travel plan example



**Figure 5.11:**  $O_{T_{QoS}}$  in the travel plan composite service



**Figure 5.12:**  $O_{T_{QoS}}$  in the composite service of 52 vertices

**Table 5.7:** CPU time in seconds of different examples with QoS constraints

Number of vertices	35			52		
CPU time (seconds) of exhaustive search method	25.04			–		
Number of simulation times in <i>QC_MCBA</i>	40	60	100	52	100	200
Probability to obtain the optimal <i>SEF</i> for <i>QC_MCBA</i>	13%	20%	32%	2%	8%	14%
Probability for $O_{T_{QoS}} \geq 0.8$	78%	87%	92%	42.5%	68.5%	90%
CPU time (seconds) of <i>QC_MCBA</i>	7.000	10.39	17.36	25.95	47.64	99.49

simulation times  $l \leq 13$ , our *QC\_MCBA* is faster than the exhaustive search method.

More significant performance differences can be observed with some of the complex composite services listed in Table 5.7. As the exhaustive search method in trust-oriented composite service selection without QoS constraints or with QoS constraints shares the same process before enumerating all solutions, it has the same CPU time before enumerating all solutions. Hence, as for the details of CPU time in the exhaustive search method, please refer to Section 5.4.3.2. We take the case of a composite service with 52 vertices as an example and depict the experimental results of *QC\_MCBA* in Fig. 5.12 and Table 5.7. In Fig. 5.12, we can observe that *QC\_MCBA* has a great chance of obtaining the near-optimal *SEF*, e.g. when  $l$  is 7, the probability for the trust-based QoS constrained *SEF* optimality being  $O_{T_{QoS}} \geq 0.85$  is about 90%. Meanwhile, the execution time of our *QC\_MCBA* is only 52% of that of the exhaustive search method. From these results, we can conclude that our proposed *QC\_MCBA* can obtain a near-optimal *SEF* after a certain number of simulations.

## 5.6 Conclusions

In this chapter, we first propose our service invocation graph and service invocation matrix for composite service representation. In addition, a novel trust evaluation approach based on Bayesian inference has been proposed that can aggregate the ratings from other clients and the requesting client's prior subjective belief about the trust. Based on them, a Monte Carlo method based trust-oriented service selection and dis-

covery algorithm (*MCBA*) and a QoS constrained Monte Carlo method based trust-oriented composite service selection algorithm (*QC\_MCBA*) have been proposed. Experiments have illustrated that our proposed approaches can discover the near-optimal composite services efficiently.

In our future work, strategies for optimizing the Monte Carlo method based algorithm will be studied to further improve its efficiency. We will also study some heuristic approaches for trust-oriented optimal service selection and discovery without or with QoS constraints.

# Subjective Trust Evaluation in Composite Services

---

In SOC environments, the trustworthiness of each service provider is critical for a service client when selecting one service provider from a large pool of service providers. The trust value of a service provider is usually in the range of  $[0,1]$  and is evaluated from the ratings given by service clients, which represent the subjective belief of service clients on the satisfaction of delivered services. A trust value can therefore be taken as a *subjective probability*, by which one party believes that another party can perform an action in a certain situation. Hence, subjective probability theory should be adopted in trust evaluation. In addition, in SOC environments a service provider can usually invoke the services of other service providers forming a composite service. Thus, the global trust of a composite service should be evaluated based on both the subjective probability property of trust and complex invocation structures.

Considering service invocation structures in composite services, in [52] we propose a global trust evaluation method. However, this method has not taken the subjective probability property of trust into account. In Chapter 5, we propose a Bayesian inference based subjective trust evaluation approach which aggregates the subjective ratings from other clients. Nevertheless, this approach still has some drawbacks. Firstly, it assumes that trust ratings conform to a normal distribution, which is a continuous distribution. However, trust ratings adopted in most existing rating systems [1, 2, 5] are discrete numbers. Thus, they cannot conform to a continuous distribution. Secondly,

the subjective probability method (Bayesian inference) in Chapter 5 is used to evaluate the trust values of service components rather than the global trust value of composite services. Finally, although the global trust evaluation of composite services in Chapter 5 has considered service invocation structures, it has not taken the subjective probability property of trust into account. In summary, in the literature, although there are a number of studies on the global trust evaluation of composite services [52, 57], some problems remain.

- Trust is context dependent, i.e. for different transaction contexts (e.g. transaction cost, product/service category, clients), different factors influence the trust result [88, 89].
- In our previous work in Section 5.3.1, a Bayesian inference based subjective trust evaluation approach has been proposed for aggregating the trust ratings of service components. It assumes that the trust ratings of each service component conform to a normal distribution, which is a continuous distribution. However, in most existing rating systems [1, 2, 5], trust ratings are discrete numbers, making them nearly impossible to conform to a continuous distribution. Therefore, the trust ratings of each service component should conform to a discrete distribution, based on which subjective probability theory can be adopted properly in trust evaluation.
- In composite services, all the dependency between service components results from direct invocations. When subjective probability theory is adopted in trust evaluation, the trust dependency should be interpreted properly using subjective probability theory.
- Although a variety of trust evaluation methods exist in different areas [45, 85, 95, 102], they either ignore the subjective probability property of trust ratings, or neglect complex invocation structures. As a result, no proper mechanism exists yet for the subjective global trust evaluation of composite services.

In this chapter, we propose a method for building up a projection from the trust ratings in the transaction history of a service provider to an upcoming transaction depending on the similarity between previous transactions and the upcoming one, and the familiarity between each rater and the service client of the upcoming transaction. This process is termed *context based trust normalization*. After trust normalization, we first propose a Bayesian inference based subjective trust estimation method for service components. In addition, we interpret the trust dependency caused by service invocations as conditional probability, which can be evaluated based on the trust values of service components. Furthermore, we propose a joint subjective probability approach and a subjective probability based deductive approach to evaluate the subjective global trust of a composite service on the basis of trust dependency.

This chapter is organized as follows. Section 6.1 presents our context based trust normalization method to evaluate the trust value that would be closely bound to the upcoming transaction, and some corresponding experiments for illustrating that our proposed method can detect some typical risks. Section 6.2 presents a joint subjective probability approach and a novel Subjective probability based deductive (SELECTIVE) approach in composite services, and some corresponding experiments for further illustrating that when compared with existing approaches our proposed SELECTIVE approach can yield more reasonable results than existing approaches. Finally Section 6.3 concludes our work in this chapter.

## 6.1 Trust Normalization in Service-Oriented Environments

Trust is context dependent, i.e. for service-oriented transactions, different factors (e.g. transaction cost, service category) influence the trust result [88, 89]. These factors should be considered comprehensively to adjust strategies in trust evaluation. Otherwise, malicious service providers and fraudulent transactions may widely exist with

the following types of risks.

**Type 1 risk** In EC or SOC environments, a typical risk is to accumulate a good reputation by offering cheap and attractive products/services, then cheating clients by offering expensive products/services [88, 89].

**Type 2 risk** This risk is accumulating a good reputation by providing products/services in a similar category (e.g. cameras), and then cheating clients by offering products/services in a different category (e.g. watches) with which the service provider has insufficient experience [88, 89].

**Type 3 risk** This risk is to collude within a small group to earn a good reputation, then cheating victims out side of the group [99, 100].

In order to evaluate the trust value that would be closely bound to an upcoming transaction, we need ratings to reflect the quality of previous transactions by the service provider. However, the trust ratings of previous transactions with different contexts should not be aggregated without considering contextual difference when obtaining the trust value [88, 89]. In our proposed method, a projection is built up from the trust ratings in the service provider's transaction history to the upcoming transaction depending on the similarity between previous transactions and the upcoming one, and the familiarity between each rater and the service client of the upcoming transaction. This process is named *context based trust normalization*. After trust normalization, normalized trust ratings are used for trust evaluation, the results of which would be closely bound to the upcoming transaction.

In fact, trust normalization is preprocessing before trust evaluation. This makes trust normalization totally different from any trust evaluation methods. In addition, this also makes it easily transferable to any trust evaluation method without many modifications.

Trust is a very complicated issue, including many uncertain factors [14]. Hence, with the fuzziness, fuzzy comprehensive evaluation provided by fuzzy set theory [8]

can deal with context based trust normalization in a reasonable manner. The fuzzy comprehensive evaluation based method analyzes complicated questions in terms of factors, by decomposing questions into several factors, stipulating every score of each factor and weighing the evaluated results using a certain scale. In particular, the fuzzy comprehensive evaluation based method is superior to other evaluation methods in dealing with subjective factors [8, 23].

### **6.1.1 Comprehensive Evaluation Index System**

In order to provide an effective fuzzy comprehensive evaluation based method in context based trust normalization, it is necessary to firstly establish a systematic and comprehensive index system. This index system includes all influencing factors, analyzes the relationships between factors and finds out which main factors influence context based trust normalization the most. The criteria for developing the comprehensive evaluation index system are as follows [23]:

- The index system must be capable of reflecting every aspect influencing context based trust normalization, i.e. every aspect determining the similarity between previous transactions and the upcoming one, and the familiarity between each rater and the service client of the upcoming transaction.
- The data for the factors in the index system must be consistent, and must be capable of being collected from reliable sources.
- The index system must be capable of accommodating the relationship between factors and the evaluation criteria, especially of generating corresponding main factors at the request of evaluators.

According to the aforementioned definition of trust, difference in context based trust in SOC environments is determined by both the difference of transactions and the difference of involved parties. Therefore, following the above criteria, in context based trust normalization, in order to determine the similarity between previous transactions

and the upcoming one, and the familiarity between each rater and the service client of the upcoming transaction, we set up a comprehensive evaluation index system, which consists of

**Transaction cost relativity:** In economics and related disciplines, the transaction cost is the cost incurred in making an economic exchange [6]. The larger the cost of the previous transaction is than the cost of the upcoming transaction is, the higher the transaction cost relativity is, and vice versa.

**Transaction category similarity:** The more similar to the category of product/service in the upcoming transaction the category of product/service in the previous transaction, the higher the transaction category similarity, and vice versa.

**Social relationship influence:** The social relationship influence is determined by social relationships, such as the social intimacy degree and the role impact factor [60]. The higher the social intimacy degree, the larger the social relationship influence. The larger the role impact factor value, the larger the social relationship influence.

## 6.1.2 Fuzzy Comprehensive Evaluation Model

The fuzzy comprehensive evaluation model is a synthetical application of analytical hierarchy process and fuzzy mathematics by inspecting many influencing factors.

### 6.1.2.1 Single-level and Multi-level Fuzzy Comprehensive Evaluations

Fuzzy comprehensive evaluation can be divided into single-level and multi-level. Generally, single-level fuzzy comprehensive evaluation is adopted to evaluate the case that there are few factors involved in the evaluation process. The steps of single-level fuzzy comprehensive evaluation are as follows.

- First, determine the *affiliation score of each factor* in the comprehensive evaluation index system, then a *fuzzy affiliation matrix* is obtained.

- Second, an *affiliation vector* which evaluates the similarity between previous transactions and the upcoming one and the familiarity between each rater and the service client of the upcoming transaction can be obtained by the composition operation of the affiliation matrix and the *weight vector of factors*.
- Last, evaluation results can be obtained using different principles.

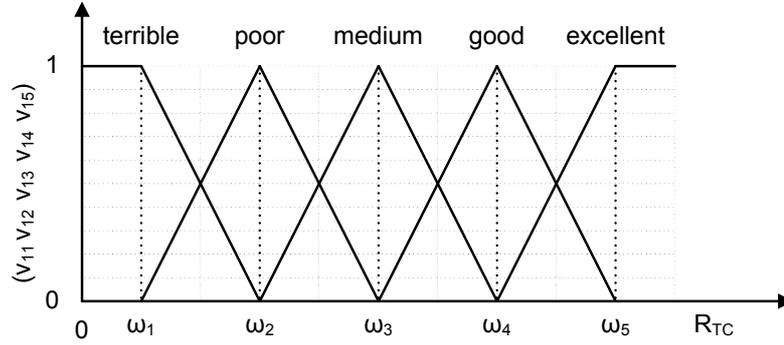
The steps of multi-level fuzzy comprehensive evaluation are as follows.

- First, the *factor set* is divided into several sub-factor sets.
- Second, as for the sub-factor sets, the single-level evaluation method is adopted to obtain some affiliation vectors.
- Third, combine the vectors to obtain a matrix, then perform a composition operation on this matrix and its immediate higher level weight vector. The evaluation vector can be obtained until the aforementioned three steps are adopted to achieve the highest level.

In this section, we set up {Transaction Cost Relativity, Transaction Category Similarity, Social Relationship Influence} as our comprehensive evaluation index system. We take this factor set as an example to illustrate our proposed single-level fuzzy comprehensive evaluation based method for context based trust normalization. If there is any necessity to process the more detailed analysis involving more factors with multiple levels, we can follow the aforementioned three steps for multi-level fuzzy comprehensive evaluation, which are similar to our single-level fuzzy comprehensive evaluation discussed in this section.

### 6.1.2.2 Establishing Affiliation Score Set

In real systems, the trust rating scores of a service provider given by raters are represented by a series of fixed numbers [54]. For example, the rating scores at eBay [1] are in the set of  $\{-1, 0, 1\}$ . At Epinions [2], each rating score is an integer in  $\{1, 2, 3, 4, 5\}$ .



**Figure 6.1:** Triangular membership functions with five levels

At YouTube [5], each rating score is in  $\{-10, -9, \dots, 10\}$ . In this section, we take the five-level scale at Epinions [2] as an example. The affiliation score set is  $\{1, 2, 3, 4, 5\}$ . With linguistic interpretation for these scores, the rater's corresponding comment set can be labeled as  $\{\text{terrible, poor, medium, good, excellent}\}$ .

### 6.1.2.3 Establishing the Fuzzy Affiliation Matrix

Here we establish the affiliation score of each factor in the comprehensive evaluation index system.

Considering transaction cost relativity, firstly let  $C_{previous}$  denote a previous transaction cost of the service provider and  $C_{upcoming}$  denote the upcoming transaction cost, then the transaction cost relativity value can be evaluated by

$$R_{TC} = \frac{C_{previous}}{C_{upcoming}}. \quad (6.1)$$

To determine the corresponding frequency  $\{(v_{11} \ v_{12} \ v_{13} \ v_{14} \ v_{15}) \mid \sum_{i=1}^5 v_{1i} = 1\}$  about the comment set  $\{\text{terrible, poor, medium, good, excellent}\}$  of the transaction cost relativity, we introduce triangular membership functions with five levels [8] as follows,

which is also illustrated in Fig. 6.1.

$$(v_{11} \ v_{12} \ v_{13} \ v_{14} \ v_{15}) = \begin{cases} v_{11} = 1; v_{12} = v_{13} = v_{14} = v_{15} = 0; & \text{if } R_{TC} \leq \omega_{N_1}; \\ v_{11} = \frac{R_{TC} - \omega_{N_2}}{\omega_{N_1} - \omega_{N_2}}; v_{12} = \frac{R_{TC} - \omega_{N_1}}{\omega_{N_2} - \omega_{N_1}}; v_{13} = v_{14} = v_{15} = 0; & \text{if } \omega_{N_1} < R_{TC} \leq \omega_{N_2}; \\ v_{12} = \frac{R_{TC} - \omega_{N_3}}{\omega_{N_2} - \omega_{N_3}}; v_{13} = \frac{R_{TC} - \omega_{N_2}}{\omega_{N_3} - \omega_{N_2}}; v_{11} = v_{14} = v_{15} = 0; & \text{if } \omega_{N_2} < R_{TC} \leq \omega_{N_3}; \\ v_{13} = \frac{R_{TC} - \omega_{N_4}}{\omega_{N_3} - \omega_{N_4}}; v_{14} = \frac{R_{TC} - \omega_{N_3}}{\omega_{N_4} - \omega_{N_3}}; v_{11} = v_{12} = v_{15} = 0; & \text{if } \omega_{N_3} < R_{TC} \leq \omega_{N_4}; \\ v_{14} = \frac{R_{TC} - \omega_{N_5}}{\omega_{N_4} - \omega_{N_5}}; v_{15} = \frac{R_{TC} - \omega_{N_4}}{\omega_{N_5} - \omega_{N_4}}; v_{11} = v_{12} = v_{13} = 0; & \text{if } \omega_{N_4} < R_{TC} \leq \omega_{N_5}; \\ v_{15} = 1; v_{11} = v_{12} = v_{13} = v_{14} = 0; & \text{if } R_{TC} > \omega_{N_5}; \end{cases} \quad (6.2)$$

where  $\omega_{N_1}, \omega_{N_2}, \omega_{N_3}, \omega_{N_4}$  and  $\omega_{N_5}$  are the parameters for determining the membership function curves. The values of these parameters are assessed by the subjective knowledge of domain experts [8] in advance and stored in the trust management authority. In this section, we adopt the triangular membership functions with five levels in Eq. (6.2) as an example to illustrate our method. It is a similar process to adopt any other membership functions.

For calculating transaction category similarity, firstly it is necessary to build up the category system based on connectivity in the network and lexicosyntactic matching [73, 74]. With this domain independent taxonomy, the similarity between transaction categories can be evaluated [74]. By applying the triangular membership functions with five levels [8], the corresponding frequency

$$\{(v_{21} \ v_{22} \ v_{23} \ v_{24} \ v_{25}) \mid \sum_{i=1}^5 v_{2i} = 1\} \quad (6.3)$$

about the comment set {terrible, poor, medium, good, excellent} of the transaction category similarity can be determined.

Social relationship influence can be determined by both social intimacy degree and role impact factor. Social intimacy degree is used to describe the extent to which two parties have intimate social relationships, because we usually trust the party with

which we have more intimate social relationships [7, 60]. The value of the social intimacy degree can be estimated by collecting information regarding social relationships [81]. The role impact factor is defined to reflect the impact of a party's recommendation role on trust propagation, because we usually put greater reliance on the party with particular social positions where his/her actions weigh heavily on his/her social position [6, 60]. The value of role impact factor can be estimated using the PageRank model [81]. By combining the social intimacy degree and the role impact factor and applying the triangular membership functions with five levels [8], the corresponding frequency

$$\{(v_{31} \ v_{32} \ v_{33} \ v_{34} \ v_{35}) \mid \sum_{i=1}^5 v_{3i} = 1\} \quad (6.4)$$

pertaining to the comment set {terrible, poor, medium, good, excellent} of the social relationship influence can be determined.

Therefore, we can obtain the fuzzy affiliation matrix  $M$  in Eq. (6.5).

$$M = \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} & v_{15} \\ v_{21} & v_{22} & v_{23} & v_{24} & v_{25} \\ v_{31} & v_{32} & v_{33} & v_{34} & v_{35} \end{bmatrix} \quad (6.5)$$

#### 6.1.2.4 Establishing Weight Vector

The weights in the weight vector  $W = (\omega_{N_6} \ \omega_{N_7} \ \omega_{N_8})$  of factors in the index system reflect the importance of each factor, and  $\omega_{N_6} + \omega_{N_7} + \omega_{N_8} = 1$ . These weights are specified in the service client's preference or fulfilled by domain experts [8] based on their own knowledge and experience.

#### 6.1.2.5 Establishing Affiliation Vector

As we have pointed out, the affiliation vector, which evaluates the similarity between previous transactions and the upcoming one, and the familiarity between each rater and the service client of the upcoming transaction, can be obtained by the composition

operation of the affiliation matrix and the weight vector of factors. If the affiliation vector  $A = (A_1 A_2 A_3 A_4 A_5)$ , then we have

$$\begin{aligned}
 A &= W \times M \\
 &= (\omega_{N_6} \ \omega_{N_7} \ \omega_{N_8}) \times \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} & v_{15} \\ v_{21} & v_{22} & v_{23} & v_{24} & v_{25} \\ v_{31} & v_{32} & v_{33} & v_{34} & v_{35} \end{bmatrix} \\
 &= (A_1 A_2 A_3 A_4 A_5)
 \end{aligned} \tag{6.6}$$

i.e.

$$A_i = \omega_{N_6}v_{1i} + \omega_{N_7}v_{2i} + \omega_{N_8}v_{3i}, \quad i = 1, 2, 3, 4, 5. \tag{6.7}$$

Because of the similarity between previous transactions and the upcoming one, and the familiarity between each rater and the service client of the upcoming transaction can be evaluated, the affiliation vector is obtained.

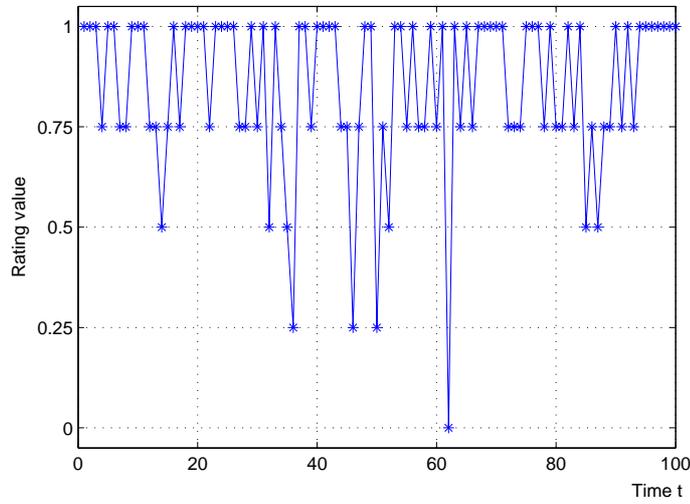
### 6.1.2.6 Discounting Rate

Since in this section the five-level scale at Epinions [2] is adopted as an example, ratings are scaled to be in  $\{0, 0.25, 0.5, 0.75, 1\}$  [54], which corresponds with the comments {terrible, poor, medium, good, excellent} respectively. Then the *discounting rate* can be determined

$$D_r = (A_1 A_2 A_3 A_4 A_5) \times (0 \ 0.25 \ 0.5 \ 0.75 \ 1)^T \tag{6.8}$$

$$= 0.25A_2 + 0.5A_3 + 0.75A_4 + A_5 \tag{6.9}$$

Therefore, the projection from the trust ratings in the transaction history of the service provider to the upcoming transaction is well established, where *the trust ratings of the service provider can be normalized by multiplying the discounting rate*. Normalized trust ratings can then be used for trust evaluation, the results of which would be closely bound to the upcoming transaction. For the details of the context based trust



**Figure 6.2:** Ratings from Epinions in Trust Normalization Experiment

normalization process, please refer to the example of context based trust normalization proposed in Section 6.1.3.1.

### 6.1.3 Experiments on Trust Normalization

In this section, we first illustrate the details of our proposed context based trust normalization method, after which we present the results of our conducted experiments for studying our proposed method and explain why the context based trust normalization method can detect some typical risks.

In these experiments, ratings are taken from Epinions [2], which is a popular on-line reputation system, and each rating is an integer in  $\{1, 2, 3, 4, 5\}$ . Since ratings as numerical values in  $[0, 1]$  are more suitable for trust evaluation [54, 98], all Epinions ratings are scaled to be in  $\{0, 0.25, 0.5, 0.75, 1\}$  in advance, which corresponds with the comments  $\{\text{terrible, poor, medium, good, excellent}\}$  respectively.

In these experiments, we set  $\omega_{N_1} = 0.1$ ,  $\omega_{N_2} = 0.5$ ,  $\omega_{N_3} = 1$ ,  $\omega_{N_4} = 5$  and  $\omega_{N_5} = 10$ , which are the thresholds for determining the triangular membership functions with five levels in Eq. (6.2) (depicted in Fig. 6.1).

### 6.1.3.1 An Example of Context Based Trust Normalization

In this experiment, we take the ratings at Epinions [2] as an example to illustrate the details of our context based trust normalization method, including establishing the affiliation score set, the fuzzy affiliation matrix, the weight vector and the affiliation vector in the fuzzy comprehensive evaluation, and determining the discounting rate.

Let us consider a scenario as follows. Service client  $A$  plans to buy a Nokia 6085 cell phone (denoted by transaction  $TR_{AP_1}$ ) from service provider  $P_{N_1}$ . Hence,  $A$  needs the ratings reflecting the quality of previous transactions about  $P_{N_1}$ , and requests the trust management authority to provide valuable information prior to placing an order and making payment. Assume we randomly select a rating  $r_{BP_1} = 0.75$  of a previous transaction  $TR_{BP_1}$  about service provider  $P_{N_1}$  rated by rater (client)  $B$ . In  $TR_{BP_1}$ ,  $B$  bought an Olympus EVOLT E-300 digital camera from  $P_{N_1}$ .

In order to normalize all relevant ratings, we need build up a projection from the ratings in the transaction history of  $P_{N_1}$  to the upcoming transaction  $TR_{AP_1}$  depending on the fuzzy affiliation matrix, which consists of both the similarity between transaction  $TR_{BP_1}$  and transaction  $TR_{AP_1}$ , and the familiarity between service client  $A$  and rater  $B$ . In order to obtain the fuzzy affiliation matrix  $M$  (defined by Eq. (6.5)), we need establish the affiliate scores of each factor (defined by Eqs (6.2)(6.3)(6.4)).

Considering the affiliate score of the transaction cost relativity, let  $C_{TR_{AP_1}}$  and  $C_{TR_{BP_1}}$  denote the transaction costs of  $TR_{AP_1}$  and  $TR_{BP_1}$  respectively. Since in SOC environments the price of a product is the major part of the transaction cost and the main concern of clients, we take this price as the transaction cost in this section. Hence, we have  $C_{TR_{AP_1}} = \$100$  and  $C_{TR_{BP_1}} = \$900$ . According to Eq. (6.1), the transaction cost relativity is  $R_{TC} = 9$ . With the thresholds for determining triangular membership functions in Eq. (6.2)  $\omega_{N_1} = 0.1, \omega_{N_2} = 0.5, \omega_{N_3} = 1, \omega_{N_4} = 5$  and  $\omega_{N_5} = 10$ , we can have the affiliate score of the transaction cost relativity  $(v_{11} v_{12} v_{13} v_{14} v_{15}) = (0 0 0 0.2 0.8)$ . Considering the affiliate scores of the transaction category similarity and the social relationship influence respectively, since this section does not focus on detailed data

mining techniques, we will just list the results of the affiliate scores of the transaction category similarity and the social relationship influence without any computational details. Hence, we can have the affiliate score of the transaction category similarity  $(v_{21} \ v_{22} \ v_{23} \ v_{24} \ v_{25}) = (0 \ 0 \ 0.4 \ 0.6 \ 0)$  and the affiliate score of the social relationship influence  $(v_{31} \ v_{32} \ v_{33} \ v_{34} \ v_{35}) = (0 \ 0 \ 0.5 \ 0.5 \ 0)$ .

Here we assume the weight vector specified in  $A$ 's preference is  $W = (0.2 \ 0.5 \ 0.3)$ . The affiliation vector  $A$  can be obtained by the composition operation of the weight vector of factors  $W$  and the affiliation matrix  $M$  defined by Eq. (6.5). Following Eq. (6.6), we have

$$\begin{aligned} A &= W \times M \\ &= (0.2 \ 0.5 \ 0.3) \times \begin{bmatrix} 0 & 0 & 0 & 0.2 & 0.8 \\ 0 & 0 & 0.4 & 0.6 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{bmatrix} \\ &= (0 \ 0 \ 0.35 \ 0.49 \ 0.16). \end{aligned} \quad (6.10)$$

In addition, following Eq. (6.9), the discount rate is

$$D_r = (0 \ 0 \ 0.35 \ 0.49 \ 0.16) \times (0 \ 0.25 \ 0.5 \ 0.75 \ 1)^T \quad (6.11)$$

$$= 0.7025. \quad (6.12)$$

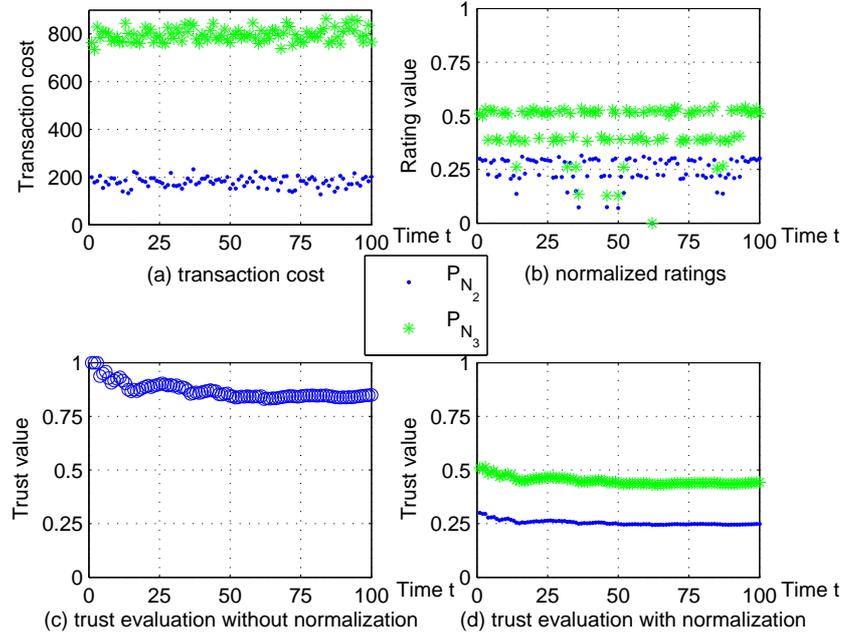
Therefore, the normalized rating  $N_{P_{N_1}}$  from  $r_{BP_1}$  can be obtained

$$N_{P_{N_1}} = r_{BP_1} \times D_r \quad (6.13)$$

$$= 0.75 \times 0.7025 \quad (6.14)$$

$$\approx 0.53.$$

Trust is context dependent, and different upcoming transactions may lead to different trust levels with the same service provider. Therefore, after trust normalization, the rating  $r_{BP_1}$ , which is 0.75 for the transaction with the product of an Olympus EVOLT



**Figure 6.3:** Experimental results about Type 1 risk

E-300 digital camera, becomes 0.53 for the transaction with the product of a Nokia 6085 cell phone.

### 6.1.3.2 Experiment on Type 1 Risk

In this experiment, we aim to illustrate how our proposed method can detect Type 1 risk, with which a malicious seller accumulates a good reputation by offering cheap and attractive services, then may cheat clients by introducing expensive services [88, 89].

In this experiment, after trust normalization we take the average of ratings as the trust evaluation method, which has been widely adopted in many trust evaluation models [41, 59].

Service client  $A$  plans to buy a Canon PowerShot A640 digital camera, which is provided by two service providers  $P_{N_2}$  and  $P_{N_3}$  with the same transaction cost  $C_{TRAP_2} = C_{TRAP_3} = \$900$ . We assume that they both have the same ratings illustrated in Fig. 6.2. Their transaction costs are listed in Fig. 6.3 (a). We can observe that

$C_{TR_{P_{N_2}}}$  is much smaller than  $C_{TR_{P_{N_3}}}$ , which means  $P_{N_2}$  accumulates a good reputation by offering cheap services and starts to provide expensive services now, i.e. Type 1 risk.

For the sake of simplicity, we assume that  $P_{N_2}$  and  $P_{N_3}$  have the same affiliate scores of transaction category similarity and social relationship influence. Then we have the fuzzy affiliation matrices

$$M_{P_{N_2}} = \begin{bmatrix} y_{11} & 1 - y_{11} & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0.6 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{bmatrix}, \quad (6.15)$$

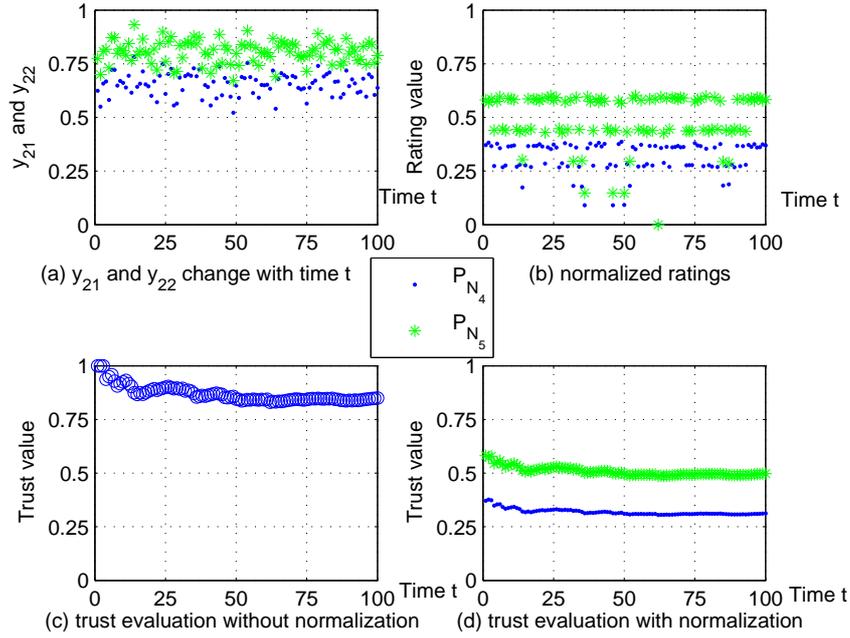
$$M_{P_{N_3}} = \begin{bmatrix} 0 & 1 - y_{12} & y_{12} & 0 & 0 \\ 0 & 0 & 0.4 & 0.6 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{bmatrix}, \quad (6.16)$$

where  $y_{11}$  and  $y_{12}$  is determined by the triangular membership functions in Eq. (6.2) with the values of  $C_{TR_{P_{N_2}}}$  and  $C_{TR_{P_{N_3}}}$ . With the weight vector  $W_1 = (0.6 \ 0.2 \ 0.2)$ , the corresponding normalized ratings of  $P_{N_2}$  and  $P_{N_3}$  are listed in Fig. 6.3 (b). For the details of context based trust normalization, please refer to Section 6.1.3.1.

Without trust normalization,  $P_{N_2}$  and  $P_{N_3}$  have the same trust evaluation results, which are depicted in Fig. 6.3 (c). However, considering the evaluated trust results with trust normalization as plotted in Fig. 6.3 (d), service client  $A$  prefers  $P_{N_3}$  to  $P_{N_2}$  because the final trust value of  $P_{N_3}$  is much larger than that of  $P_{N_2}$ . That preference results from the fact that  $P_{N_2}$  may be a malicious service provider who cheats service clients. Therefore, we can observe that our proposed context based trust normalization method can detect Type 1 risk.

### 6.1.3.3 Experiment on Type 2 Risk

In this experiment, we also take the average of ratings as the trust evaluation method [41, 59] to illustrate that our proposed context based trust normalization method can



**Figure 6.4:** Experimental results about Type 2 risk

detect Type 2 risk, which is to accumulate a good reputation by providing services in a similar category, then cheating clients by offering services in a different category with which the service provider has insufficient experience [88, 89].

Service client  $A$  plans to buy a Canon PowerShot A640 digital camera, which is provided by two service providers  $P_{N_4}$  and  $P_{N_5}$ . We assume that they both have the same ratings as illustrated in Fig. 6.2. However,  $P_{N_4}$  used to sell watches while  $P_{N_5}$  used to sell camcorders. We can observe that the transaction category similarity between camcorders and digital cameras is larger than that between watches and digital cameras, because camcorders and digital cameras belong to the electronics category, which excludes watches. This means that  $P_{N_4}$  accumulates a good reputation by providing services in a similar category, then starts to provide services in a different category, i.e. Type 2 risk.

For the sake of simplicity, we assume that  $P_{N_4}$  and  $P_{N_5}$  have the same affiliate scores of transaction cost relativity and social relationship influence. This then pro-

duces the fuzzy affiliation matrices

$$M_{P_{N_4}} = \begin{bmatrix} 0 & 0 & 0 & 0.2 & 0.8 \\ y_{21} & 1 - y_{21} & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{bmatrix}, \quad (6.17)$$

$$M_{P_{N_5}} = \begin{bmatrix} 0 & 0 & 0 & 0.2 & 0.8 \\ 0 & 1 - y_{22} & y_{22} & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{bmatrix}, \quad (6.18)$$

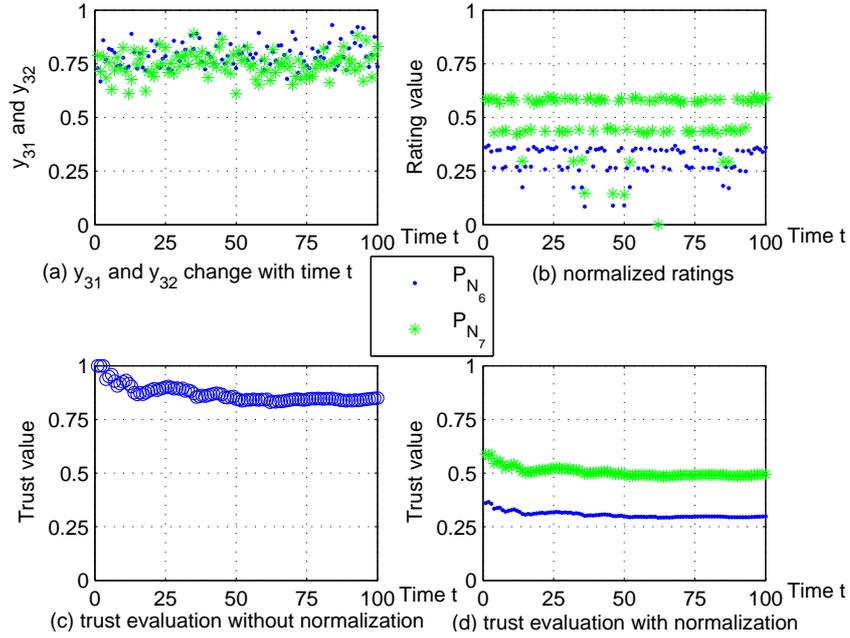
where the values of  $y_{21}$  and  $y_{22}$  are listed in Fig. 6.4 (a). With the weight vector  $W_2 = (0.2 \ 0.6 \ 0.2)$ , the corresponding normalized ratings of  $P_{N_4}$  and  $P_{N_5}$  can be obtained, as represented in Fig. 6.4 (b).

With these normalized ratings, the evaluation results are illustrated in Fig. 6.4 (d). Compared with the same trust values of  $P_{N_4}$  and  $P_{N_5}$  obtained by trust evaluation without normalization depicted in Fig. 6.4 (c), in Fig. 6.4 (d) we can observe that the evaluated trust value of  $P_{N_4}$  is much smaller than that of  $P_{N_5}$ . This results from the fact that  $P_{N_4}$  may be a malicious service provider who cheats service clients. Therefore, our proposed context based trust normalization method can detect Type 2 risk.

#### 6.1.3.4 Experiment on Type 3 Risk

In this experiment, by adopting the average of ratings as the trust evaluation method [41, 59] we aim to illustrate that our context based trust normalization method can detect Type 3 risk, which is colluding within a small group to earn a good reputation, then cheating victims outside of the group [99, 100].

Service client  $A$  plans to buy a Canon PowerShot A640 digital camera, which is provided by two service providers  $P_{N_6}$  and  $P_{N_7}$ . We assume that they both have the same ratings, as illustrated in Fig. 6.2. However,  $P_{N_6}$  used to have frequent transactions with a few clients.  $P_{N_6}$  and these clients comprise a group who only conduct transactions amongst themselves. In contrast,  $P_{N_7}$  used to conduct transactions with a



**Figure 6.5:** Experimental results about Type 3 risk

large variety of clients. Hence, we can observe that the social relationship influence of  $P_{N_6}$  is much smaller than that of  $P_{N_7}$ . That means that  $P_{N_6}$  collaborates with a small group to earn a good reputation, then starts to cheat victims outside of the group, i.e. Type 3 risk.

For the sake of simplicity, we assume that  $P_{N_6}$  and  $P_{N_7}$  have the same affiliate scores of transaction cost relativity and transaction category similarity. We then have the fuzzy affiliation matrices

$$M_{P_{N_6}} = \begin{bmatrix} 0 & 0 & 0 & 0.2 & 0.8 \\ 0 & 0 & 0.4 & 0.6 & 0 \\ y_{31} & 1 - y_{31} & 0 & 0 & 0 \end{bmatrix}, \quad (6.19)$$

$$M_{P_{N_7}} = \begin{bmatrix} 0 & 0 & 0 & 0.2 & 0.8 \\ 0 & 0 & 0.4 & 0.6 & 0 \\ 0 & 1 - y_{32} & y_{32} & 0 & 0 \end{bmatrix}, \quad (6.20)$$

where the values of  $y_{31}$  and  $y_{32}$  are listed in Fig. 6.5 (a). With the weight vector

$W_3 = (0.2 \ 0.2 \ 0.6)$ , the corresponding normalized ratings can be obtained, as depicted in Fig. 6.5 (b).

With these normalized ratings, the evaluation results are illustrated in Fig. 6.5 (d). Compared with the same trust values of  $P_{N_6}$  and  $P_{N_7}$  obtained by trust evaluation without normalization depicted in Fig. 6.5 (c), we can observe from Fig. 6.5 (d) that the evaluated trust value of  $P_{N_6}$  is much smaller than that of  $P_{N_7}$ . This results from the fact that  $P_{N_6}$  may be a malicious service provider who cheats service clients. Therefore, our proposed context based trust normalization method can detect Type 3 risk.

## 6.2 Subjective Global Trust Evaluation in Composite Services

If the trust rating of a service is scaled to the range of  $[0, 1]$ , it can represent the subjective probability with which the service provider is believed to be able to perform the service satisfactorily [41]. Therefore, *subjective probability theory* [33, 38] is the right tool for dealing with trust ratings [57].

In Section 6.2.1, based on *Bayesian inference* [30, 33], which is an important component in subjective probability theory, we propose a novel method that evaluates the subjective trust of service components from a series of ratings given by service clients. In Section 6.2.2, we interpret the trust dependency caused by service invocations as conditional probability, which is evaluated based on the subjective trust values of service components. In Section 6.2.3, we propose a joint subjective probability method that evaluates the subjective global trust value of a *SEF* from the trust values and trust dependency of all service components.

## 6.2.1 Trust Estimation of Service Components

In most existing rating systems [1, 2, 5], trust ratings are discrete numbers, making the number of occurrences of ratings for each service component conform to a multinomial distribution [33]. That is because in statistics if each trial results in exactly one of  $k$  ( $k$  is a fixed positive integer) kinds of possible outcomes with certain probabilities, the number of occurrences of outcome  $i$  ( $1 \leq i \leq k$ ) must follow a multinomial distribution [33].

### 6.2.1.1 Rating Space and Trust Space

In real systems, the trust ratings of a service given by service clients are represented by a series of fixed numbers. For example, the ratings at eBay [1] are in the set  $\{-1, 0, 1\}$ . At Epinions [2], each rating is an integer in  $\{1, 2, 3, 4, 5\}$ . At YouTube [5], each rating is in  $\{-10, -9, \dots, 10\}$ . In order to analyze these ratings, they should be normalized to the range of  $[0, 1]$  in advance. Hence, the interval  $[0, 1]$  is partitioned into  $k$  mutually exclusive ratings, say  $r_1, r_2, \dots$ , and  $r_k$  ( $0 \leq r_{i-1} < r_i \leq 1$ ). For example, at Epinions [2], after normalization, the ratings are in  $\{0, 0.25, 0.5, 0.75, 1\}$ . Hence,  $r_1 = 0$ ,  $r_2 = 0.25$ ,  $r_3 = 0.5$ ,  $r_4 = 0.75$ , and  $r_5 = 1$ . Let  $p_i = P(r_i)$  be the probability for a service to obtain the rating  $r_i$  ( $i = 1, 2, \dots, k$ ), and  $\sum_{i=1}^k p_i = 1$ . Let  $x_i$  be the number of occurrences of rating  $r_i$  in the rating sample, and  $n = \sum_{i=1}^k x_i$ .

Traditionally, some principles [41, 88] have been considered in trust evaluation. One of them is to assign higher weights to the trust values of later services [51, 102], which can be interpreted as discounting former  $x_i$  over time. Because of such discounting,  $x_i$  is taken as a real number. Accordingly, the rating space is modeled as  $R = \mathbb{R}^k$ , a  $k$ -dimensional space of reals.

**Definition 33:** The *rating space* for each service component is

$$R = \{X = (x_1, x_2, \dots, x_k) | x_i \geq 0, x_i \in \mathbb{R}, i = 1, 2, \dots, k\}.$$

Following the definition in [39], the trust space for each service component can be partitioned into *trust* (a good outcome), *distrust* (a bad outcome) and *uncertainty*.

**Definition 34:** The *trust space* for each service component is

$$T = \{(t, d, u) | t \geq 0, d \geq 0, u \geq 0, t + d + u = 1\}.$$

Hence, if  $C$  is a service component in composite services, then let  $t_C$ ,  $d_C$  and  $u_C$  denote the trust, distrust and uncertainty of  $C$ , respectively.

### 6.2.1.2 Bayesian Inference

The primary goal of adopting Bayesian inference [30, 33] is to summarize the available information that defines the distribution of trust ratings through the specification of probability density functions, such as prior distribution and posterior distribution. The *prior distribution* summarizes the subjective information about the trust prior to obtaining the rating sample  $X = (x_1, x_2, \dots, x_k)$ . Once  $X$  is obtained, the prior distribution can be updated to represent the *posterior distribution*.

Let  $V = (p_1, p_2, \dots, p_{k-1})$  and  $p_k = 1 - \sum_{i=1}^{k-1} p_i$ . Due to a lack of additional information, we can first assume that the prior distribution  $f(V)$  is a uniform distribution. Since the rating sample  $X$  conforms to a multinomial distribution [33], i.e.

$$f(X|V) = \frac{n!}{\prod_{i=1}^k (x_i!)} \prod_{i=1}^k p_i^{x_i}, \quad (6.21)$$

the posterior distribution can be estimated [33]

$$\begin{aligned} f(V|X) &= \frac{f(X|V)f(V)}{\int_0^1 \int_0^1 \cdots \int_0^1 f(X|V)f(V) dp_1 dp_2 \cdots dp_{k-1}} \\ &= \frac{(1 - \sum_{i=1}^{k-1} p_i)^{x_k} \prod_{i=1}^{k-1} p_i^{x_i}}{\int_0^1 \int_0^1 \cdots \int_0^1 ((1 - \sum_{i=1}^{k-1} p_i)^{x_k} \prod_{i=1}^{k-1} p_i^{x_i}) dp_1 dp_2 \cdots dp_{k-1}} \end{aligned} \quad (6.22)$$

### 6.2.1.3 Certainty, Expected Positiveness and Expected Negativeness

The certainty of trust captures the confirmation of trust from ratings, i.e. for services with the same trust value, a service client prefers the service with the trust value determined by more ratings [39].

In this section, the certainty of trust is defined based on statistical measure [91]. Since the cumulative probability of the probability distribution of  $V$  within  $\Omega$  must be 1, let the distribution of  $V$  follow the function given below  $g : \Omega = [0, 1] \times [0, 1] \times \dots \times [0, 1] \rightarrow [0, \infty)$  such that  $\int_{\Omega} g(V) dV = 1$ . Hence, the mean value of  $g(V)$  within  $\Omega$  is  $\frac{\int_{\Omega} g(V) dV}{(1-0)^{k-1}} = 1$ . Following the common principle in statistics [33], without additional information, we can take the prior distribution  $g(V)$  as a uniform distribution. The certainty can be evaluated based on the mean absolute deviation from the prior distribution [91]. Since  $g(V)$  has a mean value of 1, both increment and reduction from 1 are counted by  $|g(V) - 1|$ . So  $\frac{1}{2}$  is needed to remove the double counting. Therefore, certainty is defined as follows:

**Definition 35:** The *certainty* based on rating sample  $X$  is

$$c(X) = \frac{1}{2} \int_{\Omega} \left| \frac{(1 - \sum_{i=1}^{k-1} p_i)^{x_k} \prod_{i=1}^{k-1} p_i^{x_i}}{\int_{\Omega} ((1 - \sum_{i=1}^{k-1} p_i)^{x_k} \prod_{i=1}^{k-1} p_i^{x_i}) dV} - 1 \right| dV$$

Since  $\frac{1}{2}$  is the middle point of the range of ratings  $[0, 1]$ , which represents the neutral belief between distrust and trust, the ratings in  $(\frac{1}{2}, 1]$  can be taken as positive ratings and the ratings in  $[0, \frac{1}{2})$  can be taken as negative ratings.

**Definition 36:** *Expected positiveness* is defined as being the expected degree for ratings to be positive

$$\alpha(X) = \frac{\sum_{r_i > \frac{1}{2}} (2r_i - 1)x_i}{\sum_{i=1}^k x_i}, \quad (6.23)$$

and *expected negativeness* is defined by

$$\beta(X) = \frac{\sum_{r_i < \frac{1}{2}} (1 - 2r_i)x_i}{\sum_{i=1}^k x_i}. \quad (6.24)$$

#### 6.2.1.4 From Rating Space to Trust Space

**Definition 37:** Let  $Z(X) = (t, d, u)$  be a transformation function from rating space  $R$  to trust space  $T$  such that  $Z(X) = (t, d, u)$ , where  $t = \alpha(X)c(X)$ ,  $d = \beta(X)c(X)$ , and  $u = 1 - (\alpha(X) + \beta(X))c(X)$ .

According to Definition 37, we have  $t + d + u = 1$  and the following definition.

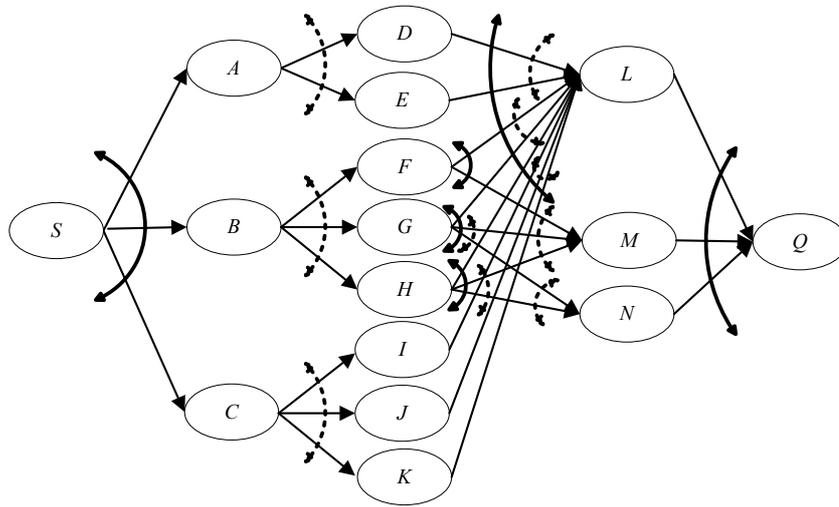
**Definition 38:** For each trust rating  $r_i \in [0, 1]$ , we have

$$r_i \text{ is } \begin{cases} \text{trust,} & \text{if } r_i \geq 1 - t, \\ \text{distrust,} & \text{if } r_i \leq d, \\ \text{uncertain,} & \text{if } d < r_i < 1 - t, \end{cases} \quad (6.25)$$

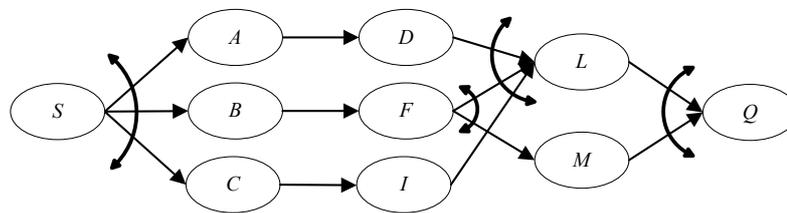
**Example 2:** Let's take the travel plan in Section 5.2.2 as an example. In this example, starting from a *service invocation root*  $S$  and ending at a *service invocation terminal*  $Q$ , the composite service consisting of all combinations of travel plans can be depicted by a *service invocation graph (SIG)* in Fig. 6.6. Each feasible travel plan is termed a *service execution flow (SEF)*, which is a subgraph of a *SIG*. A *SEF* example of the *SIG* in Fig. 6.6 is plotted in Fig. 6.7.

Let's take the service execution flow (*SEF*) in Fig. 6.7 as an example to illustrate the trust estimation of a service component. All ratings of service components in Fig. 6.7 are taken from Epinions [2] and are listed in Table 6.1, where each row corresponds with an execution of the *SEF*.

For service component  $C$  in Fig. 6.7 with all its 20 trust ratings in column  $C$  in Table 6.1, according to Definitions 35, 36 and 37, based on the ratings listed in Table 6.1, we can obtain  $c = 0.88$ ,  $\alpha = 0.48$ ,  $\beta = 0.03$ ,  $t = 0.42$ ,  $u = 0.56$  and  $d = 0.02$ . Similarly, all these parameters for each service component in the *SEF* in Fig. 6.7 can be obtained, as listed in Table 6.2. Hereafter, for each rating  $r_{Ci}$  of  $C$ , according to



**Figure 6.6:** The service invocation graph (SIG) for the travel plan



**Figure 6.7:** A service execution flow (SEF) in the SIG

**Table 6.1:** Ratings for service components in the *SEF* in Fig. 6.7

	<i>S</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>F</i>	<i>G</i>	<i>I</i>	<i>L</i>	<i>M</i>	<i>Q</i>
$e_1$	1	0.5	1	0.75	1	0.75	1	1	1	0.75	1
$e_2$	1	0.75	1	0.75	1	1	0.75	1	0.75	1	1
$e_3$	1	0.75	0.75	0.75	0.25	1	0	1	1	1	1
$e_4$	1	1	1	0.75	1	1	1	1	1	0.75	1
$e_5$	1	1	1	1	1	0.75	0.75	1	1	1	1
$e_6$	1	1	0.75	0.75	0.25	0.75	1	1	1	1	1
$e_7$	1	0.5	1	0.75	0	0.5	1	1	1	0.75	1
$e_8$	1	1	0.75	0.75	1	0.75	0.75	0.75	1	1	1
$e_9$	1	1	0.75	0.75	0.75	1	0.75	0.5	1	1	1
$e_{10}$	1	0.75	0.75	0.75	0.75	0.75	0.75	1	0.75	1	1
$e_{11}$	1	0.75	0.75	0.5	1	1	1	0.75	1	1	1
$e_{12}$	1	0.75	1	1	0.75	1	0	0.5	0.75	0.5	1
$e_{13}$	1	0.75	0.75	0.75	1	1	0.75	0	1	0.5	1
$e_{14}$	1	1	0.5	0.75	0.75	1	0.75	1	1	1	1
$e_{15}$	1	1	1	0.75	1	0.75	0.75	1	1	0.75	1
$e_{16}$	1	1	1	0.75	1	1	1	1	1	0.75	1
$e_{17}$	1	1	1	0.25	0.25	1	0	0.5	0.75	1	1
$e_{18}$	0.75	1	0.75	0.5	1	1	0.75	1	1	1	1
$e_{19}$	1	0.75	0.75	1	1	1	0.75	0.75	0.75	1	1
$e_{20}$	1	1	1	0.5	1	0.75	0.75	1	1	0.75	1

Definition 38, we have

$$r_{C_i} \text{ is } \begin{cases} \text{trust,} & \text{if } r_{C_i} \geq 0.58; \\ \text{distrust,} & \text{if } r_{C_i} \leq 0.02; \\ \text{uncertain,} & \text{if } 0.02 < r_{C_i} < 0.58. \end{cases} \quad (6.26)$$

Let  $t_Y$ ,  $d_Y$  and  $u_Y$  denote the *trust*, *distrust* or *uncertain* of any service component  $Y$  in the *SEF* in Fig. 6.7, which are calculated and listed in Table 6.3.

## 6.2.2 Probability Interpretation of Trust Dependency

Dependency is a state in which one object uses a functionality of another object [30]. In composite services, dependency between service components results from direct



invocations [54], e.g. if there is a direct invocation from service component  $A$  to service component  $B$ , we say that “ $B$  depends on  $A$ ”.

**Definition 39:** In composite services, *trust dependency* represents the fact that the trust of a service component is only dependent on its trust propensity and the trust of its direct predecessor(s).

According to the theorem about the probabilities of conditionals and conditional probabilities [34], we can use conditional probability to formalize trust dependency.

**Definition 40:** In composite services, if  $\{P_i\}$  are the direct predecessors of service component  $S$ , and  $P$  is the rational subjective probability function, we have

$$P(\bigwedge P_i \succeq S) = P(S | \bigwedge P_i).$$

Following Definition 40, the important link between probability theory and invocations in composite services has been well established. Probability theory will then be a source of insight into the invocation structure of composite services.

The graphical representation of composite services, the service invocation graph (*SIG*), pictorially represents the service dependency properties in composite services [57]. In addition, Definition 40 enables a *SIG* to prove arbitrary service dependency conjectures concerning any service component in composite services. For example, in the *SEF* of Fig. 6.7, which is one of the feasible service compositions of the *SIG* in Fig. 6.6, we can immediately read off the graph that

$$Q \perp\!\!\!\perp (S \wedge A \wedge B \wedge C \wedge D \wedge F \wedge I) \text{ and } (L \wedge M) \succeq Q, \quad (6.27)$$

where  $\perp\!\!\!\perp$  denotes “is independent of”.

In order to evaluate the conditional probability for trust dependency, in *subjective probability theory* [38] the following principle has been proposed for bridging from classic probability (i.e. the occurrence frequency of an event) to subjective probability

(i.e. the degree of belief that an individual has in the truth of a proposition).

**Principle 4:** Without any additional knowledge, our knowledge that the chance of hypothesis  $H$  has probability  $p$  guarantees that our subjective probability for  $H$  is  $p$ .

Therefore, according to the definition of conditional probability, the trust dependency, which is the conditional probability of the trustworthiness of a service component given the trustworthiness of its predecessors, can be evaluated based on Principle 4. E.g. if  $\{P_i\}$  are the direct predecessors of service component  $S$ , the evaluation of trust dependency of  $\bigwedge P_i \succeq S$  is to evaluate  $P(t_S|t_{\bigwedge P_i})$ ,  $P(t_S|u_{\bigwedge P_i})$  and  $P(t_S|d_{\bigwedge P_i})$ . In addition, as the service invocation root in a *SIG* has no predecessor, its trust dependency can be evaluated directly from its ratings according to Principle 4.

Here we assume that when a rating of a delivered service is stored by the trust management authority, the invocation relationship (i.e. the predecessor(s) of the delivered service) is also recorded.

**Example 3:** Let us continue the computation in Example 2 to illustrate the evaluation of the conditional probability for the trust dependency in composite services.

Let  $t_Y$ ,  $d_Y$  and  $u_Y$  denote the *trust*, *distrust* or *uncertain* of any service component  $Y$  in the *SEF* in Fig. 6.7, which are calculated and listed in Table 6.3. As service component  $C$  invokes service component  $I$  (denoted as  $C \succeq I$ ) in Fig. 6.7, following the definition of conditional probability,  $P(t_I|t_C)$  is the chance of  $I$  to be *trust* given its direct predecessor  $C$  is *trust*. According to the results in Table 6.3, there are 20 executions in total and 13 of them (i.e.  $e_1-e_8$ ,  $e_{10}$ ,  $e_{14}-e_{16}$  and  $e_{19}$ ) are the case that  $I$  is *trust* given  $C$  is *trust*. Hence, we have  $P(t_I|t_C)=13/20=0.65$ . Likewise, we can compute  $P(t_I|u_C)=3/20=0.15$  and  $P(t_I|d_C)=0/20=0$ . The trust dependency of  $C \succeq I$  has been evaluated. Following the same procedure, all the conditional probabilities corresponding to each trust dependency in a *SEF* can be evaluated and listed in Table 6.4.

**Table 6.4:** Trust dependency in the *SEF* in Fig. 6.7

$P(t_A t_S)$	1	$P(t_B t_S)$	1	$P(t_L t_D \wedge t_F \wedge t_I)$	0.65
$P(t_A \neg t_S)$	0	$P(t_B \neg t_S)$	0	$P(t_L \neg(t_D \wedge t_F \wedge t_I))$	0.35
$P(t_C t_S)$	0.8	$P(t_D t_A)$	0.8	$P(t_Q t_L \wedge t_M)$	1
$P(t_C \neg t_S)$	0	$P(t_D \neg t_A)$	0	$P(t_Q \neg(t_L \wedge t_M))$	0
$P(t_F t_B)$	1	$P(t_I t_C)$	0.65	$P(t_M t_F)$	1
$P(t_F \neg t_B)$	0	$P(t_I \neg t_C)$	0.15	$P(t_M \neg t_F)$	0
$P(d_A d_S)$	0	$P(d_B d_S)$	0	$P(d_L d_D \wedge d_F \wedge d_I)$	0
$P(d_A \neg d_S)$	0	$P(d_B \neg d_S)$	0	$P(d_L \neg(d_D \wedge d_F \wedge d_I))$	0
$P(d_C d_S)$	0	$P(d_D d_A)$	0	$P(d_Q d_L \wedge d_M)$	0
$P(d_C \neg d_S)$	0	$P(d_D \neg d_A)$	0.05	$P(d_Q \neg(d_L \wedge d_M))$	0
$P(d_F d_B)$	0	$P(d_I d_C)$	0	$P(d_M d_F)$	0
$P(d_F \neg d_B)$	0	$P(d_I \neg d_C)$	0.05	$P(d_M \neg d_F)$	0

### 6.2.3 Joint Subjective Probability Method

In this section, the *joint subjective probability method* is proposed to take the subjective global trust value of a *SEF*,  $P(t_{SEF})$ , as a joint probability distribution.

**Definition 41:** The subjective global trust value of a *SEF* can be factorized into a series of trust dependencies in the *SEF*, i.e.

$$P(t_{SEF}) = \prod_{v \in SEF} P(t_v | \bigwedge_{u^{(i)} \in SEF, u^{(i)} \succeq v} t_{u^{(i)}}). \quad (6.28)$$

Let's take the *SEF* in Fig. 6.7 as an example to illustrate our proposed joint subjective probability method. Following Definition 41, we can obtain

$$\begin{aligned}
P(t_{SEF}) &= P(t_S)P(t_A|t_S)P(t_B|t_S)P(t_C|t_S)P(t_D|t_A) \\
&\quad P(t_F|t_B)P(t_I|t_C)P(t_L|t_D \wedge t_F \wedge t_I) \\
&\quad P(t_M|t_F)P(t_Q|t_L \wedge t_M)
\end{aligned} \quad (6.29)$$

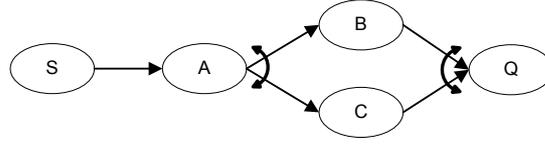
By applying the breadth-first search algorithm, since each trust dependency (e.g.  $P(t_I|t_C)$  or  $P(t_Q|t_L \wedge t_M)$ ) in a *SEF* (e.g. those in Eq. (6.29)) can be evaluated (as illustrated in Example 3), the subjective global trust value of the *SEF* in Fig. 6.7,  $P(t_{SEF})$ , can be computed. Due to space constraints, the details of this algorithm are omitted.

#### 6.2.4 Subjective Probability Based Deductive (SELECTIVE) Approach

In this section, we propose a Subjective probability based deductive (SELECTIVE) approach that evaluates the subjective global trustworthiness of a *SEF* from all the trust dependency in the *SEF*. This process follows subjective probability theory and maintains the subjective probability property of trust in evaluations.

In *subjective probability theory*, there is *causal decision theory* [38]. In this section, we borrow causal decision theory's idea in dealing with subjective decision-making based on the imputations of probabilistic causal influence, but extend it to the evaluation of the subjective global trustworthiness of a *SEF* in composite services.

On the basis of trust dependency caused by direct invocations, an indirect invocation also leads to a certain dependency relationship, which is actually the composition of trust dependency. For example, in Fig. 6.8, service component *A* is dependent on service component *S* and service component *B* is dependent on *A*. Therefore, *B* is dependent on *S* indirectly, compositing trust dependency  $S \succeq A$  and  $A \succeq B$ . As each trust dependency corresponds with a conditional probability, a composition of trust dependency can yield a subjective probability. In the process of composition, we can compute the probability of trustworthiness of a service component based on all its preceding trust dependency. Hence, the composition of trust dependency starting from the service invocation root to any intermediate service component in a *SEF* until the service invocation terminal is reached can deductively compute the subjective probability of trustworthiness of the service invocation terminal based on all the trust dependency in the *SEF*, which is taken as the global trust value of the *SEF*.



**Figure 6.8:** A simple *SEF* example

Now let us illustrate the evaluation process of our SELECTIVE approach. The computational process starts from the service invocation root of a *SEF* until the service invocation terminal is reached. For any intermediate service component  $V$ , when its direct predecessors  $\{W_i\}$  are processed, we have  $P(\bigwedge t_{W_i})$  and  $P(\neg \bigwedge t_{W_i})$ . In addition, trust dependency  $P(t_V | \bigwedge t_{W_i})$  and  $P(t_V | \neg \bigwedge t_{W_i})$  can be evaluated following the approach illustrated in Example 3. Then according to the law of total expectation [38] in causal decision theory, we can compute

$$P(t_V) = P(t_V | \bigwedge t_{W_i})P(\bigwedge t_{W_i}) + P(t_V | \neg \bigwedge t_{W_i})P(\neg \bigwedge t_{W_i}). \quad (6.30)$$

When the service invocation terminal is finally processed, the deducted subjective trust value results from the composition of all trust dependency from the service invocation root to the service invocation terminal in the *SEF*. Hence, it is taken as the subjective global trust value of the *SEF*. Similarly, the subjective global distrust value and the subjective global uncertain value can be obtained. The details of our SELECTIVE approach are presented in Algorithm 9, which extends the *topological sort algorithm* [27] that guarantees any node in a directed graph is always visited after all its predecessors. This algorithm incurs a complexity of  $O(N_v + N_e)$ , where  $N_v$  is the number of service components (vertices) in a *SEF* and  $N_e$  is the number of invocations (edges).

**Example 4:** Let's illustrate the computational details of our proposed SELECTIVE approach with the simple *SEF* in Fig. 6.8. As we have defined in Example 3, let  $t_A$ ,  $d_A$  and  $u_A$  denote *trust*, *distrust* or *uncertain* in regard to service component  $A$  respectively. In Fig. 6.8, we can observe that since service invocation root  $S$  doesn't

**Algorithm 9** Subjective Probability Based Deductive (SELECTIVE) Approach**Input:** a *SEF*, *trust*, *distrust* or *uncertain* of each rating of each service component.**Output:** the subjective global trust value of the *SEF*  $P(t_{SEF})$ ,  
the subjective global distrust value of the *SEF*  $P(d_{SEF})$ ,  
the subjective global uncertain value of the *SEF*  $P(u_{SEF})$ .

- 1: Let the service invocation terminal of the *SEF* be  $Q$ ;
- 2: Create a stack  $s$ ;
- 3: Push service invocation root into stack  $s$ , and mark it as visited;
- 4: **while**  $s$  is not empty **do**
- 5:   Pop the vertex  $v$  on the top of stack  $s$ , and mark it as visited;
- 6:   Mark all the predecessors as  $w_i$ ;
- 7:    $P(t_v) = P(t_v | \bigwedge t_{w_i})P(\bigwedge t_{w_i}) + P(t_v | \neg \bigwedge t_{w_i})P(\neg \bigwedge t_{w_i})$ ;
- 8:    $P(d_v) = P(d_v | \bigwedge d_{w_i})P(\bigwedge d_{w_i}) + P(d_v | \neg \bigwedge d_{w_i})P(\neg \bigwedge d_{w_i})$ ;
- 9:   **for all** each successor  $u$  of  $v$  **do**
- 10:     **if**  $u$  has no unvisited predecessors **then**
- 11:       Push  $u$  into  $s$ ;
- 12:     **end if**
- 13:   **end for**
- 14: **end while**
- 15:  $P(t_{SEF}) = P(t_Q)$ ;
- 16:  $P(d_{SEF}) = P(d_Q)$ ;
- 17:  $P(u_{SEF}) = 1 - P(t_{SEF}) - P(d_{SEF})$ ;
- 18: **return**  $P(t_{SEF}), P(d_{SEF}), P(u_{SEF})$  ;

invoke service invocation terminal  $Q$  directly, there is no trust dependency of  $Q$  on  $S$  directly. However, with service components  $A$ ,  $B$  and  $C$ ,  $Q$  can be indirectly invoked by  $S$ , corresponding with the composition of trust dependency from  $S$  to  $Q$ .

In Fig. 6.8 as  $A$  is only dependent on  $S$ , and  $t_S$ ,  $d_S$  and  $u_S$  are the partition of the trust space for  $S$ , according to the law of total expectation, we have

$$\begin{aligned}
 P(t_A) &= P(t_A | t_S)P(t_S) + P(t_A | \neg t_S)(P(\neg t_S)) \\
 &= P(t_A | t_S)P(t_S) + P(t_A | \neg t_S)(1 - P(t_S))
 \end{aligned} \tag{6.31}$$

where  $\neg$  is the NOT operator in logic  $P(\neg X) = 1 - P(X)$ ,  $P(\neg t_S) = 1 - P(t_S) = P(d_S) + P(u_S)$ . In addition, as  $S$  is the service invocation root,  $P(t_S)$ ,  $P(d_S)$  and  $P(u_S)$  can be computed from the trust ratings of  $S$  (as illustrated in Example 2). Moreover,

trust dependency  $P(t_A|t_S)$  and  $P(t_A|\neg t_S)$  can also be computed (as illustrated in Example 3). Hence, according to Eq. (6.31),  $P(t_A)$  can be obtained. Similarly,  $P(t_B)$  and  $P(t_C)$  can be deducted as follows,

$$P(t_B) = P(t_B|t_A)P(t_A) + P(t_B|\neg t_A)(1 - P(t_B)), \quad (6.32)$$

$$P(t_C) = P(t_C|t_A)P(t_A) + P(t_C|\neg t_A)(1 - P(t_C)). \quad (6.33)$$

As  $P(t_B|t_A)$ ,  $P(t_B|\neg t_A)$ ,  $P(t_C|t_A)$  and  $P(t_C|\neg t_A)$  in Eqs. (6.32) and (6.33) can be computed (as illustrated in Example 3),  $P(t_B)$  and  $P(t_C)$  can be obtained respectively. In Fig. 6.8, as the service invocation terminal  $Q$  is dependent on service components  $B$  and  $C$ , we have

$$\begin{aligned} P(t_Q) &= P(t_Q|t_B \wedge t_C)P(t_B \wedge t_C) \\ &\quad + P(t_Q|\neg(t_B \wedge t_C))P(\neg(t_B \wedge t_C)). \end{aligned} \quad (6.34)$$

In addition, as  $B$  and  $C$  are independent of each other,  $P(t_B \wedge t_C) = P(t_B)P(t_C)$ . Hence, from Eq. (6.34) we have

$$\begin{aligned} P(t_Q) &= P(t_Q|t_B \wedge t_C)P(t_B)P(t_C) \\ &\quad + P(t_Q|\neg(t_B \wedge t_C))(1 - P(t_B)P(t_C)). \end{aligned} \quad (6.35)$$

In Eq. (6.35), as both  $P(t_Q|t_B \wedge t_C)$  and  $P(t_Q|\neg(t_B \wedge t_C))$  can be computed (as illustrated in Example 3),  $P(t_Q)$  can be obtained. Therefore, the global trust of the *SEF*,  $P(t_{SEF}) = P(t_Q)$ , can be evaluated from the above deductive process from Eq. (6.31) to Eq. (6.35).

When applying Algorithm 9 to the *SEF* in Fig. 6.7, we have the following process

in steps.

$$\text{Step 1 : } P(t_A) = P(t_A|t_S)P(t_S) + P(t_A|\neg t_S)(1 - P(t_S)); \quad (6.36)$$

$$\text{Step 2 : } P(t_B) = P(t_B|t_S)P(t_S) + P(t_B|\neg t_S)(1 - P(t_S)); \quad (6.37)$$

$$\text{Step 3 : } P(t_C) = P(t_C|t_S)P(t_S) + P(t_C|\neg t_S)(1 - P(t_S)); \quad (6.38)$$

$$\text{Step 4 : } P(t_D) = P(t_D|t_A)P(t_A) + P(t_D|\neg t_A)(1 - P(t_A)); \quad (6.39)$$

$$\text{Step 5 : } P(t_F) = P(t_F|t_B)P(t_B) + P(t_F|\neg t_B)(1 - P(t_B)); \quad (6.40)$$

$$\text{Step 6 : } P(t_I) = P(t_I|t_C)P(t_C) + P(t_I|\neg t_C)(1 - P(t_C)); \quad (6.41)$$

$$\begin{aligned} \text{Step 7 : } P(t_L) &= P(t_L|t_D \wedge t_F \wedge t_I)P(t_D)P(t_F)P(t_I) \\ &\quad + P(t_L|\neg(t_D \wedge t_F \wedge t_I))(1 - P(t_D)P(t_F)P(t_I)); \end{aligned} \quad (6.42)$$

$$\text{Step 8 : } P(t_M) = P(t_M|t_F)P(t_F) + P(t_M|\neg t_F)(1 - P(t_F)); \quad (6.43)$$

$$\begin{aligned} \text{Step 9 : } P(t_Q) &= P(t_Q|t_L \wedge t_M)P(t_L)P(t_M) \\ &\quad + P(t_Q|\neg(t_L \wedge t_M))(1 - P(t_L)P(t_M)); \end{aligned} \quad (6.44)$$

$$\text{Step 10 : } P(t_{SEF}) = P(t_Q). \quad (6.45)$$

As each trust dependency, represented by a conditional probability in the above equations, can be computed (as illustrated in Example 3), from Eq. (6.36) to Eq. (6.45) the subjective global trust value of the *SEF* in Fig. 6.7,  $P(t_{SEF})$  can be finally obtained. Similarly, the subjective global distrust value  $P(d_{SEF})$  of the *SEF* in Fig. 6.7 can be computed, with which the subjective global uncertain value  $P(u_{SEF}) = 1 - P(t_{SEF}) - P(d_{SEF})$  can be obtained.

Regarding the comparison of multiple *SEFs* with the same functionality, we firstly compare their trust values. For any two *SEFs*, the one with a larger trust value is preferable. If they have approximately the same trust value, their distrust values should be compared. The *SEF* with a lower distrust value is preferable.

$SEF_1$  with  $(P(t_{SEF_1}), P(d_{SEF_1}), P(u_{SEF_1}))$  and  $SEF_2$  with  $(P(t_{SEF_2}), P(d_{SEF_2}), P(u_{SEF_2}))$  are comparable in the following cases:

**Case 1:** If  $|P(t_{SEF_1}) - P(t_{SEF_2})| < \epsilon_{SEF_1}$  and  $|P(d_{SEF_1}) - P(d_{SEF_2})| < \epsilon_{SEF_2}$ ,  $SEF_1$  and  $SEF_2$  are equivalent in trust level, where  $0 < \epsilon_{SEF_1}, \epsilon_{SEF_2} \ll 1$  are thresholds that can be specified by service clients or trust management authorities.

**Case 2:** If  $P(t_{SEF_1}) - P(t_{SEF_2}) > \epsilon_{SEF_1}$ ,  $SEF_1$  is preferable.

**Case 3:** If  $|P(t_{SEF_1}) - P(t_{SEF_2})| < \epsilon_{SEF_1}$  and  $P(d_{SEF_1}) - P(d_{SEF_2}) > \epsilon_{SEF_2}$ ,  $SEF_2$  is preferable.

### 6.2.5 Experiments and Analysis on Subjective Global Trust Evaluation

In this section, we will study the properties of our trust estimation method for service components, after which we will present the experimental results for studying our proposed SELECTIVE approach and explaining the following questions:

- Q1:** why should service invocation structures be taken into account in trust evaluation?
- Q2:** why should trust dependency caused by direct invocations be taken into account in global trust evaluation?
- Q3:** why should the dependency caused by indirect invocations be taken into account in trust evaluation?

In these experiments, ratings are taken from Epinions [2], which is a popular online reputation system, and each rating is an integer in  $\{1, 2, 3, 4, 5\}$ . After normalization, a rating is in  $\{0, 0.25, 0.5, 0.75, 1\}$ . The rating data set adopted in this chapter has 664824 ratings in total, out of which 6.50% are 0, 7.62% are 0.25, 11.36% are 0.5, 29.23% are 0.75 and 45.28% are 1. In general, the ratings at Epinions are observed to be surprisingly positive.

We set  $\epsilon_{SEF_1} = \epsilon_{SEF_2} = 0.001$ , which are the thresholds in the comparison of multiple  $SEFs$ .

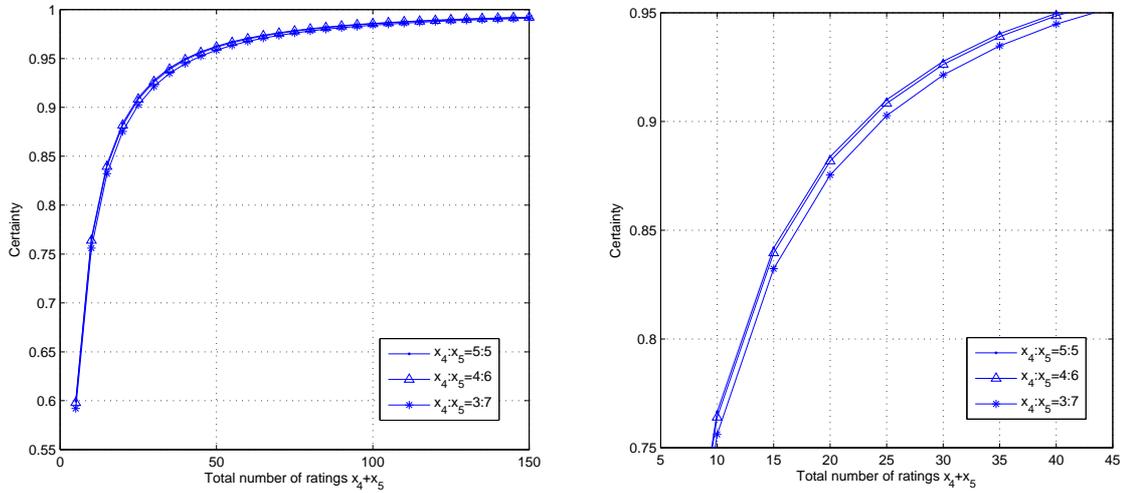


Figure 6.9: Certainty with fixed ratio of  $x_4$  and  $x_5$

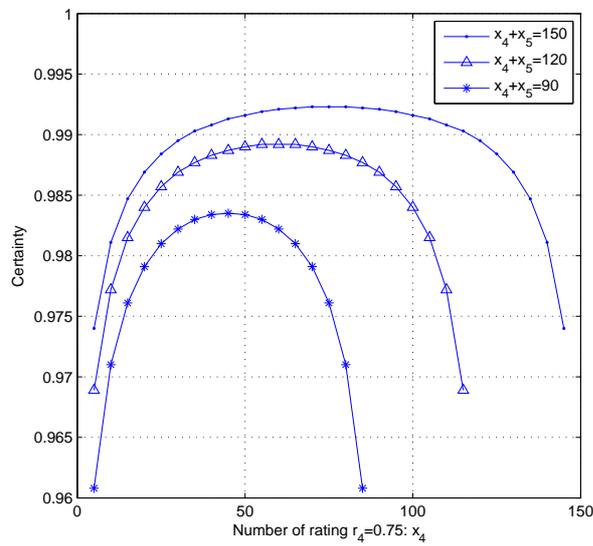


Figure 6.10: Certainty with fixed  $x_4 + x_5$

### 6.2.5.1 Important Properties in Trust Estimation

Since certainty is important for the trust estimation of service components, which is the foundation of our proposed subjective global trust evaluation method, we will illustrate its important properties in this section.

Let  $x_i$  be the number of occurrences of rating  $r_i$  in the rating sample ( $0 \leq i \leq k$ ). In this section, we will mostly focus on the cases when  $x_1 = x_2 = x_3 = 0$ , which corresponds to the scenario that our adopted Epinions rating data set is observed to be surprisingly positive. This scenario also universally exists in the other rating data sets, such as eBay, as reported in [41].

Firstly, let us consider a scenario where the total number of ratings is increasing when  $x_1 = x_2 = x_3 = 0$  and the ratios of  $x_4$  and  $x_5$  are fixed. Let  $x_4 : x_5$  be  $3 : 7$ ,  $4 : 6$  and  $5 : 5$  respectively. We can observe how the function curve of certainty changes in Fig. 6.9, where Fig. 6.9 (right) is partly enlarged in comparison to Fig. 6.9 (left). Below, we introduce a theorem that generalizes the case illustrated in Fig. 6.9.

**Theorem 8:** If  $x_i : x_j$  ( $i \neq j$ ) is fixed, given the fixed  $x_h$  ( $h \neq i, h \neq j$ ), the certainty of ratings increases with respect to the total number of ratings.

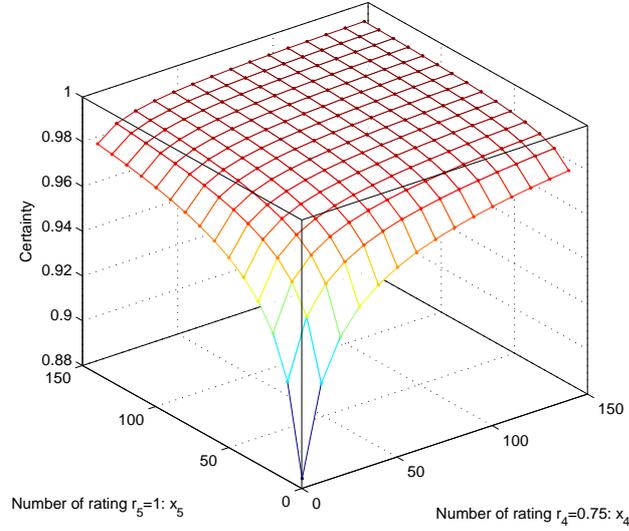
**Proof idea:** Show that  $c'(x_j, x_k, x_l, x_m, x_i) > 0$  for  $x_i : x_j = k$  and fixed  $x_k, x_l$  and  $x_m$ . □

Due to space constraints, the full proofs of all theorems in this chapter are included in Appendix B.

Secondly, let us consider a scenario where  $x_4$  is increasing when  $x_1 = x_2 = x_3 = 0$  and  $x_4 + x_5$  is fixed. We set  $x_4 + x_5 = 150, 120$  or  $90$  respectively, and observe how the function curve of certainty changes in Fig. 6.10. Hence, we can have the following theorem generalizing the observations.

**Theorem 9:** If  $sum = x_i + x_j$  ( $i \neq j$ ) is fixed, given the fixed  $x_h$  ( $h \neq i, h \neq j$ ), the certainty of ratings is increasing with respect to  $x_i$  when  $x_i < sum/2$ ; otherwise, the certainty of ratings is decreasing with respect to  $x_i$  when  $x_i > sum/2$ .

**Proof idea:** Show that the deviation from the prior distribution increases with the



**Figure 6.11:** Certainty with  $x_4$  and  $x_5$  when  $x_1 = x_2 = x_3 = 0$

increment of  $x_i$  when  $x_i < sum/2$ , and the deviation from the prior distribution decreases with the increment of  $x_i$  when  $x_i > sum/2$ . □

In addition, let us consider a scenario where  $x_4$  and  $x_5$  are both increasing when  $x_1 = x_2 = x_3 = 0$ .

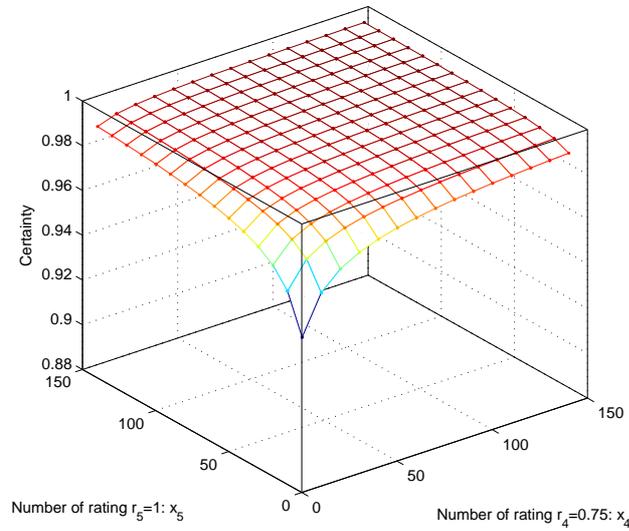
In Fig. 6.11, when  $x_4$  is fixed and  $x_1 = x_2 = x_3 = 0$ , the certainty of ratings increases with the increment of  $x_5$ . Meanwhile, when  $x_5$  is fixed and  $x_1 = x_2 = x_3 = 0$ , the certainty of ratings increases with respect to  $x_4$ . Furthermore, we can observe that the plane of certainty function is symmetric with the plane of  $x_4 = x_5$ . Hence, we can have the following theorem.

**Theorem 10:**  $c(x_i, x_j, x_k, x_l, x_m) = c(x_j, x_k, x_l, x_i, x_m)$  for fixed  $x_k, x_l$  and  $x_m$ .

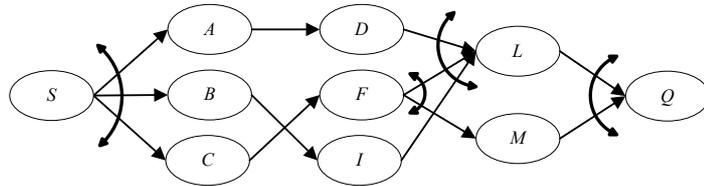
**Proof:** According to Definition 35,  $c(x_i, x_j, x_k, x_l, x_m) = c(x_j, x_k, x_l, x_i, x_m)$  for fixed  $x_k, x_l$  and  $x_m$ . □

Finally, let us consider a scenario where  $x_4$  and  $x_5$  are increasing when  $x_1 = x_2 = 0$  and  $x_3 = 10$ . The properties illustrated in Fig. 6.12 are similar to those in Fig. 6.11. Hence, we can have the following theorem.

**Theorem 11:** The certainty of ratings increases with respect to  $x_i$  (i.e. the number of



**Figure 6.12:** Certainty with  $x_4$  and  $x_5$  when  $x_1 = x_2 = 0$  and  $x_3 = 10$



**Figure 6.13:** Service execution flow  $SEF_2$

occurrences of rating  $r_i$ ), given all the fixed  $x_h$  ( $h \neq i$ ).

**Proof idea:** Show that given all the fixed  $x_h$ , the deviation from the prior distribution increases with the increment of  $x_i$ . □

The above theorems show how certainty, which is important to determine trust according to Definition 37, evolves with respect to the increment of the number of a rating’s occurrences under different conditions. Following these theorems, a service provider who wishes his/her service to achieve a specific level of certainty can ask the trust management authority to find out how many trust ratings would be needed under a certain condition, or a service client can ask the trust management authority to compute certainty to see if a service has reached an acceptable level.

### 6.2.5.2 Experiment 1 on Service Invocation Structure

In this experiment, we take the *SEF* in Fig. 6.7 (denoted as  $SEF_1$ ) as an example to illustrate the computational details of our proposed SELECTIVE approach. By comparing our SELECTIVE approach with the existing global trust evaluation approach discussed in [104], we explain why service invocation structures should be taken into account in trust evaluation (Q1).

As illustrated in Example 2 and listed in Table 6.3, each rating of a service component can be judged as *trust*, *distrust* or *uncertain*. The trust dependency in composite services can then be evaluated, as illustrated in Example 3 and listed in Table 6.4. According to Definition 39, the *trust* of service invocation root  $S$  can be computed based on its ratings directly, and  $P(t_S) = 1$ . Following our proposed SELECTIVE approach, the subjective global trust value of  $SEF_1$  in Fig. 6.7,  $P(t_{SEF_1})$ , can be calculated according to Eqs. (6.36) to (6.45). Hence, we can obtain  $P(t_{SEF_1}) = 0.4820$ . Similarly, we have  $P(d_{SEF_1}) = 0$  and  $P(u_{SEF_1}) = 1 - P(t_{SEF_1}) - P(d_{SEF_1}) = 0.5180$ .

In order to illustrate the necessity of service invocation structures in trust evaluation, let us change the service invocation structure of  $SEF_1$  in Fig. 6.7 to be the one in Fig. 6.13, which is denoted as  $SEF_2$ . The difference between  $SEF_1$  and  $SEF_2$  is that in Fig. 6.7 service component  $B$  invokes  $F$  and  $C$  invokes  $I$ . In contrast, in Fig. 6.13,  $B$  invokes  $I$  and  $C$  invokes  $F$ . The trust dependency in  $SEF_2$  can be evaluated as  $P(t_B|t_S) = 0.8$ ,  $P(t_B|\neg t_S) = 0$ ,  $P(t_C|t_S) = 1$ ,  $P(t_C|\neg t_S) = 0$ ,  $P(t_F|t_B) = 0.8$ ,  $P(t_F|\neg t_B) = 0.2$ ,  $P(t_I|t_C) = 0.8$ ,  $P(t_I|\neg t_C) = 0$ , and the remaining trust dependency in  $SEF_2$  is listed in Table 6.4. Following our SELECTIVE approach (from Eq. (6.36) to Eq. (6.45)), the subjective global trust value of  $SEF_2$  is  $P(t_{SEF_2}) = 0.3268$ . When compared with  $P(t_{SEF_1}) = 0.4820$ , we can observe that following our SELECTIVE approach, we prefer  $SEF_1$  to  $SEF_2$  (refer to Table 6.5).

In most existing global trust evaluation approaches, service invocation structures have not been taken into account (e.g. the approach proposed in [104]). When we change the service invocation structure (e.g. change the invocation structure in Fig.

**Table 6.5:** Trust results in Experiment 1

	$SEF_1$	$SEF_2$
Global trust evaluation approach in [104]	$T_{SEF_1} = 0.8763$	$T_{SEF_2} = 0.8763$
SELECTIVE approach	$P(t_{SEF_1}) = 0.4820$	$P(t_{SEF_2}) = 0.3268$

6.7 to be the one in Fig. 6.13), the global trust value remains unchanged in these approaches (e.g.  $T_{SEF_1} = T_{SEF_2} = 0.8763$  (refer to Table 6.5)). However, considering trust dependency, when we have changed the service invocation structure, the subjective global trust value of the  $SEF$  should change as well (e.g.  $P(t_{SEF_1}) \neq P(t_{SEF_2})$ ). Hence, without considering service invocation structures, the trustworthiness of these two different  $SEFs$  cannot be distinguished. Therefore, service invocation structures should be taken into account in the global trust evaluation of composite services, and our proposed SELECTIVE approach considering service invocation structures can yield reasonable results.

### 6.2.5.3 Experiment 2 on Trust Dependency

In this section, we present the experimental results to compare our proposed SELECTIVE approach with the existing global trust evaluation approach in Section 5.3.2, and explain why trust dependency should be taken into account in global trust evaluation (Q2).

In order to illustrate the necessity of trust dependency in trust evaluation, let us exchange the rating at execution  $e_9$  (i.e.  $r_{I9} = 0.5$ ) and the rating at  $e_{18}$  (i.e.  $r_{I18} = 1$ ) for service component  $I$  in  $SEF_1$ , and let  $SEF_3$  denote the exchanged  $SEF$ . Although the global trust evaluation approach proposed in Section 5.3.2 has considered service invocation structures, the trust dependency in composite services is ignored. In this experiment, without loss of generality, the weights of service components in all Pa structures of composite services are set at 1, and the requesting client's prior subjective belief about the trust of each service component is set at  $\delta = 0.5$ . Then,

**Table 6.6:** Trust results in Experiment 2

	$SEF_1$	$SEF_3$
Global trust evaluation approach in Section 5.3.2	$T_{SEF_1} = 0.1759$	$T_{SEF_3} = 0.1759$
SELECTIVE approach	$P(t_{SEF_1}) = 0.4820$	$P(t_{SEF_3}) = 0.4892$

following Algorithm 7 in Section 5.3.2, the global trust value can be computed as  $T_{SEF_1} = T_{SEF_3} = 0.1759$  (refer to Table 6.6).

However, when compared with the trust dependency in  $SEF_1$ , the trust dependency in  $SEF_3$  changes to  $P(t_I|t_C) = 0.7$  and  $P(t_I|\neg t_C) = 0.1$ . Following our proposed SELECTIVE approach, the subjective global trust value of  $SEF_3$  is  $P(t_{SEF_3}) = 0.4892$ , which is larger than  $P(t_{SEF_1}) = 0.4820$  (also refer to Table 6.6). This means that following our SELECTIVE approach, we prefer  $SEF_3$  to  $SEF_1$ .

In the global trust evaluation approach proposed in Section 5.3.2, firstly the trust value of each service component is computed. Based on service invocation structures, these values are then aggregated to obtain the global trust value. Obviously, this approach has not considered trust dependency. However, following Definition 39, the trust of a service component is different when it is invoked by different direct predecessors. In order to consider this kind of dependency caused by invocations when evaluating the global trust of composite services, it is necessary to take into account the trust dependency between service components instead of a single trust value of every service component only. Therefore, our proposed SELECTIVE approach, which takes trust dependency into account, can yield reasonable results in the global trust evaluation of composite services.

#### 6.2.5.4 Experiment 3 on Trust-Oriented Composite Service Selection

In this experiment, we compare our proposed SELECTIVE approach with the joint subjective probability approach proposed in Section 6.2.3 by applying both approaches to the travel plan composite services in Fig. 6.6. The joint subjective probability

**Table 6.7:** Trust results in Experiment 3

	$SEF_1$	$SEF_4$
Joint subjective probability approach in Section 6.2.3	$P(SEF_1) = 0.2704$	$P(SEF_4) = 0.2774$
SELECTIVE approach	$P(t_{SEF_1}) = 0.4820$	$P(t_{SEF_4}) = 0.3537$

approach is proposed to evaluate the subjective global trust of a composite service. In this approach, we explain why the dependency caused by indirect invocations should be taken into account in trust evaluation (Q3).

Let  $SEF_1$  denote the service execution flow  $\{S, A, B, C, D, F, I, L, M \text{ and } Q\}$  and let  $SEF_4$  denote the service execution flow  $\{S, A, B, C, D, G, I, L, M \text{ and } Q\}$  in Fig. 6.6. The difference between the two  $SEF$ s is that  $F$  is in  $SEF_1$  and  $G$  is in  $SEF_4$ . All the ratings of service components in this section are taken from Epinions [2] and are listed in Table 6.1. For the sake of simplicity, we assume that the ratings of  $L$  and  $M$  when invoked by  $G$  are the same as their ratings when invoked by  $F$ .

Following the joint subjective probability approach, with the trust dependency listed in Table 6.4, the subjective global trust value of  $SEF_1$  can be obtained as  $P(SEF_1) = 0.2704$ . In  $SEF_4$ , the trust dependency related to service component  $G$  can be evaluated (as illustrated in Example 3). Hence, we have  $P(t_G|t_B) = 0.85$ ,  $P(t_G|\neg t_B) = 0$ ,  $P(t_L|t_D \wedge t_G \wedge t_I) = 0.6$ ,  $P(t_L|\neg(t_D \wedge t_G \wedge t_I)) = 0.4$ ,  $P(t_M|t_G) = 0.85$  and  $P(t_M|\neg t_G) = 0.15$ , and the remaining trust dependency in  $SEF_4$  can be evaluated, as listed in Table 6.4. Following the joint subjective probability approach in Section 6.2.3, we have  $P(SEF_4) = 0.2774$ , which is *larger than*  $P(SEF_1) = 0.2704$  (also refer to Table 6.7).

In contrast, following our proposed SELECTIVE approach, as we have illustrated in the first experiment, the subjective global trust value of  $SEF_1$  is  $P(t_{SEF_1}) = 0.4820$ . Similarly, the subjective global trust value of  $SEF_4$  is  $P(t_{SEF_4}) = 0.3537$ , which is *smaller than*  $P(t_{SEF_1}) = 0.4820$  (also refer to Table 6.7).

Both approaches consider both service invocation structures and trust dependency. However, in the above two examples, they yield completely different conclusions. That

is because the joint subjective probability approach only considers trust dependency  $P(t_V | \bigwedge t_{W_i})$ , i.e. the possibility of all service invocations in a composite service are trustworthy, where  $\{W_i\}$  are the direct predecessors of  $V$ . In contrast, according to the law of total expectation in causal decision theory, our SELECTIVE approach takes into account not only trust dependency  $P(t_V | \bigwedge t_{W_i})$ , but also trust dependency  $P(t_V | \neg \bigwedge t_{W_i})$ , which contains both  $P(t_V | \bigvee u_{W_i})$  and  $P(t_V | \bigvee d_{W_i})$ . In our SELECTIVE approach, every service component in a composite service corresponds with a subjective probability value, which is computed based on all its preceding trust dependency. Thus, the service invocation terminal corresponds with a subjective probability value. This value is computed based on all trust dependency in the composite service, caused by both direct invocations and indirect ones, and is taken as the global trust value of the composite service. Therefore, our SELECTIVE approach can compute more reasonable results that can reflect the global trustworthiness of a composite service in comparison to existing approaches.

## 6.3 Conclusions

In this chapter, we firstly propose a fuzzy comprehensive evaluation based method for building up a projection from the trust ratings in the transaction history of a service provider to an upcoming transaction. This process is named context based trust normalization. After trust normalization, normalized trust ratings are used directly for trust evaluation, the result of which would be closely bound to the upcoming transaction, and thus is more accurate and objective. Hence, trust normalization is preprocessing before trust evaluation, which makes it easily transferable to any trust evaluation method without many modifications. From our experimental results, we can observe that our proposed context based trust normalization method can detect some typical risks in transactions that can hardly be identified by existing trust evaluation methods calculating the general and global trust value only.

With the normalized ratings, in composite services our proposed subjective trust

estimation method for service components is based on Bayesian inference, which is a component of subjective probability theory. This novel method can aggregate the non-binary discrete subjective ratings given by service clients and maintain the subjective probability property of trust. In addition, the trust dependency caused by service invocations is interpreted as conditional probability, which is evaluated based on the subjective trust estimation of service components. This novel interpretation makes it feasible to deal with invocation structures using subjective probability theory. Furthermore, on the basis of the above fundamental subjective trust estimation and probability interpretation of trust dependency, a subjective probability based deductive approach has been proposed to evaluate the subjective global trustworthiness of a composite service. Experiments have demonstrated that our approach can deliver reasonable results that are critical for the decision-making of service clients in trustworthy service selection.

To our best knowledge, this is the first work in the literature on subjective trust estimation for service components based on non-binary discrete ratings. This is also the first work in the literature on the subjective global trust evaluation of composite services with complex invocation structures.

In composite services, if there is a dependency between QoS (Quality of Service) of service components, our SELECTIVE approach can be applied in QoS-oriented composite service selection. In our future work, with our subjective global trust evaluation approach, efficient algorithms will be studied for trust-oriented composite service selection.

---

# Conclusions and Future Work

---

In recent years, Service-Oriented Computing (SOC) has emerged to be an increasingly important research area attracting attention from both the research and industry communities. In SOC applications, various services are provided to clients by different providers in a loosely-coupled environment. In such context, a service can refer to a transaction, such as selling a product online (i.e. the traditional online services), or a functional component implemented by Web service technologies. However, when a client looks for a service out of a large pool of services provided by different service providers, in addition to functionality the trust of a service provider is also a key factor for service selection.

In SOC environments, the trust issue is very important. Effective and efficient trust evaluation is highly crucial to provide valuable information to service clients, enabling them to select trustworthy service providers, utilize high quality services and prevent monetary loss.

In this thesis, two main aspects to our research studies on trust evaluation in service-oriented environments can be described.

1. One aspect of the work presented in this thesis is trust vector based approaches to trust rating aggregations in service-oriented environments.
  - (a) In our single trust vector approach, a trust vector is proposed to predict the trust trend and represent a set of ratings distributed within a time interval, which consists of three values: final trust level, service trust trend and ser-

vice performance consistency level. The final trust level is the global trust value represented by a value in  $[0, 1]$ . The service trust trend is computed as a numerical value in  $(-\infty, +\infty)$  representing the trend of service trust changes within a given time interval that can be interpreted as coherent, up-going, dropping or uncertain. The service performance consistency level is represented by a numerical value in  $[0, 1]$  measuring the extent to which the computed service trust trend fits the given set of trust ratings.

With trust vectors, two service providers with the similar final trust values can be compared.

- (b) In our multiple trust vector approach, a two dimensional aggregation is performed, which consists of both vertical and horizontal aggregations of trust ratings. The vertical aggregation calculates the aggregated rating representing the trust level for the services delivered in a small time period. The horizontal aggregation applies our proposed optimal multiple time intervals (MTI) algorithm to determine the minimal number of time intervals, within each of which a trust vector with three values can be calculated to represent all the ratings in that time interval and preserve the trust features well. Hence, a small set of trust vectors can represent a large set of trust ratings. This is significant for large-scale trust rating transmission and trust evaluation.

In the optimal MTI algorithm, given a set of ratings and the same threshold, several minimal sets of MTI may exist. Thus, in our future work, the optimal MTI algorithm can be further extended to find the best set of MTI with the best preserved trust features or the largest summation of *SPCL* values.

2. The other aspect of the work presented in this thesis concerns trust-oriented composite service selection and discovery.

In SOC environments, to satisfy the specified functionality requirement, it is

usually necessary to effectively compose different kinds of services across domains forming a composite service, which requires that the involved service can be trusted by service clients and other collaborating services. For example, if we need a travel plan for attending a conference, this travel plan includes a conference registration service, a hotel reservation service, an air ticket booking service and a local transportation arrangement service (to choose the airport shuttle, train or taxi). Hence, the travel plan is a composite service. Meanwhile, given a set of various services, different compositions may lead to different service structures. Although these certainly enrich the service provision, they greatly increase the computation complexity and thus make trustworthy service selection and discovery a very challenging task.

In the literature, although a variety of trust evaluation methods exist in different areas, no proper mechanism exists for evaluating the global trust of a composite service with both the subjective property of trust and the complex service invocation structure. On the basis of the subjective trust estimation of service components and the probability interpretation of trust dependency between service components, in this thesis an effective subjective trust evaluation approach is proposed to evaluate the subjective global trust value of a composite service.

In addition, taking trust evaluation and the complex structure of composite services into account, an effective and efficient algorithm is proposed for composite service selection and discovery. For the travel plan example, with our proposed algorithm, a service client can send a single request and the most trustworthy or nearly-optimal composite service, which is the best combination of service components with the largest global trust value, can be obtained automatically.

However, some research problems remain open for trust-oriented composite service selection and discovery, which will be solved in our future work.

- (a) Although we have mentioned some mechanisms from cognitive science to deal with trust (e.g. the recency effect presented in Section 3.1.1 and the

mechanism presented in Section 4.1.1), there is still a necessity to systematically present all mechanisms from cognitive science, which can deal with trust subjectively and maintain the subjective property of ratings and trust results.

- (b) With our subjective global trust evaluation approach, efficient algorithms will be studied for trust-oriented composite service selection. Especially for trust-oriented composite service selection with QoS constraints, as it is a NP-hard problem, effective and efficient QoS constrained trust-oriented composite service selection algorithms will be studied.
- (c) In composite services, if there is a dependency between QoS (Quality of Service) of service components our proposed composite service selection algorithm can be applied in QoS-oriented composite service selection.
- (d) During the execution of a service execution flow (*SEF*) in composite services, when some services suddenly become inaccessible, it is necessary to composite services again and select another *SEF*. This *SEF* selection should not only maximize the trust value of final executed *SEF*, but also maximally utilize the assessed services.

# Appendix A

---

## Notations Used in This Thesis

---

**Table A.1:** Notations Used in Chapter 3

Notation	Representation	First occurrence
$\tilde{A}(\alpha, C)$	fuzzy number	Section 3.2.2.1
$d_{t_i}$	the distance from point $(t_i, R^{(t_i)})$ to the regression line	Section 3.1.2
$FTL$	Final Trust Level	Section 3.1
$h$	goodness-of-fit	Definition 7
$H$	threshold of goodness-of-fit	Section 3.2.3
$p_{a_0}$ $p_{a_1}$	regression line $R = p_{a_0} + p_{a_1}t$	Section 3.1.2
$q_{i1}$ $q_{i2}$ $\vdots$ $q_{in}$	the advertised QoS values at time $i$	Section 3.2.3
$R^{(t_i)}$	the trust rating for the service delivered at the small time period $t_i$	Section 3.1.2
$S$	objective function in linear programming	Section 3.2.3
$S_{\tilde{A}}$	the fuzziness of $\tilde{A}$	Definition 8
$S_{dis}$	the sum of squares of the distance	Definition 2
$SPCL$	Service Performance Consistency Level	Section 3.1
$STT$	Service Trust Trend	Section 3.1
$T_{FTL}^{[t_1, t_n]}$	the $FTL$ value for the time interval $[t_1, t_n]$	Definition 1
$T_i$	the rating of the delivered service	Section 3.2.3
$T_i^*$	the center of $\tilde{T}_i^*$	Section 3.2.3
$T_{SPCL}^{[t_1, t_n]}$	the $SPCL$ value for the time interval $[t_1, t_n]$	Definition 5
$T_{STT}^{[t_1, t_n]}$	the $STT$ value for the time interval $[t_1, t_n]$	Definition 6

**Table A.2:** Notations Used in Chapter 3 (continued)

Notation	Representation	First occurrence
$\bar{T}^{[t_1, t_n]}$	The service trust vector for the trust ratings given in time interval $[t_1, t_n]$	Definition 6
$V_{pr}(t_i)$	the predicted value of regression line	Definition 3
$V_{t_{dis}}^{[t_1, t_n]}$	weighted average distance for the time interval $[t_1, t_n]$	Definition 4
$w_{t_i}$	weight at $t_i$	Definition 1
$W_j$	parameters in linear programming	Section 3.2.3
$\epsilon_{SPCL_1}$ $\epsilon_{SPCL_2}$	thresholds used to determine the case of <i>SPCL</i>	Section 3.1.3
$\epsilon_{STT}$	threshold used to determine the case of <i>STT</i>	Section 3.1.3
$\epsilon_{Vector_1}$ $\epsilon_{Vector_2}$ $\epsilon_{Vector_3}$	thresholds used to compare trust vector	Section 3.1.4
$\mu_{\tilde{A}}(x)$	the membership function of fuzzy number $\tilde{A}$	Section 3.2.2.1

**Table A.3:** Notations Used in Chapter 4

Notation	Representation	First occurrence
$d_{v_j}^{(1)}$	in the boundary included optimal MTI algorithm, the distance from $v_1$ to $v_j$	Section 4.2.6
$d_{v_j}^{(2)}$	in the boundary excluded optimal MTI algorithm, the distance from $v_1$ to $v_j$	Section 4.2.6
$d_{v_j}^{(3)}$	in the boundary mixed optimal MTI algorithm, the distance from $v_1$ to $v_j$	Section 4.2.6
$\deg^-(v_i)$	the indegree of $v_i$	Section 4.2.4
$dis(v_i)$	the distance from $v_1$ to $v_i$	Section 4.2.4
$D(v_i, v_j)$	the distance from $v_i$ to $v_j$	Section 4.2.6
$D_r^{(t_i)}$	relative rating density	Definition 15
$fre(r_j^{(t_i)})$	the frequency of $r_j^{(t_i)}$ delivered at $t_i$	Definition 15
<i>FTL</i>	Final Trust Level	Section 3.1
$G_{MTI}(\{[t_{lb_i}, t_{rb_i}]\})$	MTI goodness-of-fit	Definition 22
$m$	the number of ratings for the services delivered at $t_i$	Section 4.1
$m_0$	argument to control the function curve	Definition 18
$m'$	the number of ratings in $[R_l^{(t_i)}, R_u^{(t_i)}]$	Definition 14
$m''$	the number of trust ratings in $[R_l^{(t_i)'}, R_u^{(t_i)'}]$	Definition 20

**Table A.4:** Notations Used in Chapter 4 (continued)

Notation	Representation	First occurrence
$M$	the adjacent matrix	Section 4.2.4
MTI	multiple time intervals	Section 4.2
$n$	the number of data points, i.e. $n =  \{(t_i, R^{(t_i)})\} $	Section 4.2.2
$n_{marginal}^{(t_i)}$	the number of marginal ratings for the services delivered at $t_i$	Definition 16
$n_{total}^{(t_i)}$	the total number of ratings for the services delivered at $t_i$	Definition 16
$P_{marginal}^{(t_i)}$	the marginal rating percentage at $t_i$	Definition 16
$r_j^{(t_i)}$	rating from client $j$ for the services delivered at $t_i$	Definition 13
$r_{dis}^{(t_i)}$	the distance between $r_j^{(t_i)}$ and $R_c^{(t_i)'}$	Principle 3
$r_k^{(t_i)'}$	rating in the range of $[R_l^{(t_i)}, R_u^{(t_i)}]$	Definition 14
$\bar{r}^{(t_i)}$	the rating that would ideally represent the trust level of the service delivered at $t_i$	Section 4.1.1
$R^{(t_i)}$	vertically aggregated rating at $t_i$	Definition 14
$R^{(t_i)'}$	weighted vertically aggregated rating	Definition 20
$R_c^{(t_i)'}$	weighted centerline	Definition 19
$R_c^{(t_i)}$	centerline of ratings $\{R_c^{(t_i)}\}$	Definition 13
$R_l^{(t_i)}$	lower control limit	Section 4.1.1
$R_l^{(t_i)'}$	weighted lower control limit	Definition 19
$R_{SRR_j}^{(t_i)}$	the <i>SRR</i> value for client $j$ at $t_i$	Principle 3
$R_u^{(t_i)}$	upper control limit	Section 4.1.1
$R_u^{(t_i)'}$	weighted upper control limit	Definition 19
<i>SPCL</i>	Service Performance Consistency Level	Section 3.1
<i>SRR</i>	service rating reputation	Section 4.1.2
<i>STT</i>	Service Trust Trend	Section 3.1
$t^*$	the root of $T_{SPCL}^{[t_1, t]} = \epsilon_{MTI}$	Section 4.2.3
$t_{lb_i}$	the left boundary of the $i$ th time interval	Section 4.2.3
$t_{left}$	the left boundary of the $i$ th time interval	Section 4.2.3
$t_{mid}$	the midpoint of interval	Section 4.2.3
$t_{rb_i}$	the right boundary of the $i$ th time interval	Section 4.2.3
$t_{right}$	the right boundary of the $i$ th time interval	Section 4.2.3
$T_{FTL}^{[t_1, t_n]}$	the <i>FTL</i> value for the time interval $[t_1, t_n]$	Definition 1
$T_{SPCL}^{[t_1, t_n]}$	the <i>SPCL</i> value for the time interval $[t_1, t_n]$	Definition 5
$T_{STT}^{[t_1, t_n]}$	the <i>STT</i> value for the time interval $[t_1, t_n]$	Definition 6

**Table A.5:** Notations Used in Chapter 4 (continued)

Notation	Representation	First occurrence
$\tilde{T}_{FTL}(\{[t_{lb_i}, t_{rb_i}]\})$	final trust value aggregated from the set of MTI	Definition 21
$v_i$	the vertex which represents point $(t_i, R^{(t_i)})$	Section 4.2.4
$V_{SRR_j}^{(t_i)}$	the <i>SRR</i> value for client $j$ from $t_1$ to $t_i$	Definition 17
$w_{t_i}$	weight at $t_i$	Definition 1
$\gamma$	$\max_{j_{dis}} r_j^{(t_i)}$	Definition 18
$\epsilon_{SRR}$	the threshold of $R_{SRR_k}^{(t_i)}$	Definition 20
$\epsilon_{MTI}$	the threshold of $T_{SPCL}^{[t_1, t_i]}$ used to determine MTI	Section 4.2.2

**Table A.6:** Notations Used in Chapter 5

Notation	Representation	First occurrence
As	asynchronous activation	Section 5.2.1
$c_Y$	the cost of service component $Y$	Section 5.5.3
Ci	circular invocation	Section 5.2.1
$e$	a directed edge in $G$	Definition 23
$E$	a finite set of directed edges	Definition 23
<i>END</i>	service invocation terminal	Definition 26
$G$	service invocation graph	Definition 23
$I_{p_i}$	a set of direct predecessors invoking $v_i$	Definition 26
$I_{s_i}$	a set of direct successors invoked by $v_i$	Definition 26
$l$	simulation times	Section 5.4.2
$m_{QoS}$	the total number of QoS constraints	Section 5.5.2
$\langle M_{ij} \rangle$	service invocation matrix	Definition 28
<i>MCBA</i>	Monte Carlo method based algorithm	Section 5.4.2
<i>MCOP</i>	Multi-Constrained Optimal Path	Section 5.5.1
$O_T$	trust-based SEF optimality	Definition 31
$O_{T_{QoS}}$	trust-based QoS constrained SEF optimality	Definition 32
Pa	parallel invocation	Section 5.2.1
$P_A$	the probability for vertex $S$ to choose successor $A$	Section 5.4.2
Pr	probabilistic invocation	Section 5.2.1
$q_{cost}$	the aggregated value of cost	Section 5.5.3
$q_i(X)$	the aggregated value of the $i^{th}$ QoS attribute about <i>SEF</i> '	Section 5.5.2

**Table A.7:** Notations Used in Chapter 5 (continued)

Notation	Representation	First occurrence
$q_{time}$	the aggregated value of time	Section 5.5.3
$QC\_MCBA$	QoS constrained Monte Carlo method based algorithm	Section 5.5.2
QoS	Quality of Service	Section 5.5
$Q_i$	the $i^{th}$ QoS constraint	Section 5.5.2
$R$	the set of atomic invocations	Definition 23
$R_p$	a set of activation relations	Definition 26
$R_s$	a set of invocation relations	Definition 26
Se	sequential invocation	Section 5.2.1
$SEF$	service execution flow	Definition 27
$SEF'$	part of the $SEF$ from the service invocation root to service component $X$	Section 5.5.2
$SIG$	service invocation graph	Definition 23
SIM	service invocation matrix	Definition 28
$START$	service invocation root	Definition 26
Sy	synchronous activation	Section 5.2.1
$t_X$	the execution time of service component $X$	Section 5.5.3
$T_A$	the trust values $A$	Definition 29
$T_g$	the global trust value	Definition 29
$T_i$	the global trust value of $SEF\ i$	Definition 31
$T(X)$	the trust value of direct successor $X$	Section 5.5.2
$U_{QoS}$	utility function in $QC\_MCBA$	Section 5.5.2
$v$	a service in $G'$	Definition 23
$V$	a finite set of vertices	Definition 23
$x_1, x_2, \dots, x_n$	a sample of ratings	Section 5.3.1
$\delta$	the requesting client's prior subjective belief about the trust	Theorem 7
$\mu$	the mean of ratings	Section 5.3.1
$\sigma$	the variance of ratings	Section 5.3.1
$\omega_{CCS_A}$	the weight for $A$	Definition 30
$\omega_{QoS}$	the weight for QoS attributes	Section 5.5.2
$\omega_T$	the weight for trust	Section 5.5.2
$\gamma$	invocational	Definition 24
$v_1 \succ v_2$	$v_1$ is the direct predecessor of $v_2$	Definition 23
$\emptyset$	the empty invocation relation set	Section 5.2.3

Table A.8: Notations Used in Chapter 6

Notation	Representation	First occurrence
$A = (A_1 A_2 A_3 A_4 A_5)$	affiliation vector	Section 6.1.2.5
$c(X)$	certainty of $X$	Definition 35
$C_{previous}$	a previous transaction cost of the service provider	Section 6.1.2.3
$C_{upcoming}$	the upcoming transaction cost	Section 6.1.2.3
$d$	distrust	Section 6.2.1.1
$d_C$	the distrust of service component $C$	Section 6.2.1.1
$D_r$	discounting rate	Section 6.1.2.6
$k$	number of mutually exclusive ratings	Section 6.2.1
$M$	fuzzy affiliation matrix	Section 6.1.2.3
$p_i = P(r_i)$	the probability for a service to obtain the rating $r_i$	Section 6.2.1.1
$P(t_{SEF})$	subjective global trust value of a $SEF$	Definition 41
$P(u_{SEF})$	subjective global uncertain value of a $SEF$	Section 6.2.4
$r_i$	rating	Section 6.2.1.1
$R$	rating space	Definition 33
$R_{TC}$	transaction cost relativity	Section 6.1.2.3
$SEF$	service execution flow	Definition 27
$SIG$	service invocation graph	Definition 23
$t$	trust	Section 6.2.1.1
$t_C$	the trust of service component $C$	Section 6.2.1.1
$T$	trust space	Definition 34
$u$	uncertainty	Section 6.2.1.1
$u_C$	the uncertainty of service component $C$	Section 6.2.1.1
$(v_{11} v_{12} v_{13} v_{14} v_{15})$	the frequency about the comment set {terrible, poor, medium, good, excellent} of the transaction cost relativity	Section 6.1.2.3
$(v_{21} v_{22} v_{23} v_{24} v_{25})$	the frequency about the comment set {terrible, poor, medium, good, excellent} of the transaction category similarity	Section 6.1.2.3
$(v_{31} v_{32} v_{33} v_{34} v_{35})$	the frequency about the comment set {terrible, poor, medium, good, excellent} of the social relationship influence	Section 6.1.2.3
$V$	$(p_1, p_2, \dots, p_{k-1})$	Section 6.2.1.2
$W = (\omega_{N_6} \omega_{N_7} \omega_{N_8})$	weight vector	Section 6.1.2.4

**Table A.9:** Notations Used in Chapter 6 (continued)

Notation	Representation	First occurrence
$x_i$	the number of occurrences of rating $r_i$ in the rating sample	Section 6.2.1.1
$X$	$(x_1, x_2, \dots, x_k)$	Definition 33
$Z(X)$	transformation function	Section 6.2.1.4
$\alpha(X)$	expected positiveness of $X$	Definition 36
$\beta(X)$	expected negativeness of $X$	Definition 36
$\epsilon_{SEF_1}$ $\epsilon_{SEF_2}$	thresholds used to compare $SEFs$	Section 6.2.4
$\omega_{N_1}$ $\omega_{N_2}$ $\omega_{N_3}$ $\omega_{N_4}$ $\omega_{N_5}$	the parameters for determining membership function curves	Section 6.1.2.3
$\Omega$	$[0, 1] \times [0, 1] \times \dots \times [0, 1]$	Section 6.2.1.3
$v_1 \succeq v_2$	$v_1$ is the direct predecessor of $v_2$	Definition 23



## Appendix B

---

# Technical Report about Subjective Global Trust Evaluation

---

Considering certainty

$$c(X) = \frac{1}{2} \int_{\Omega} \left| \frac{(1 - \sum_{j=1}^{k-1} p_j)^{x_k} \prod_{j=1}^{k-1} p_j^{x_j}}{\int_{\Omega} ((1 - \sum_{j=1}^{k-1} p_j)^{x_k} \prod_{j=1}^{k-1} p_j^{x_j}) dV} - 1 \right| dV \quad (\text{B.1})$$

Let

$$F_1(P) \triangleq \frac{(1 - \sum_{j=1}^{k-1} p_j)^{x_k} \prod_{j=1}^{k-1} p_j^{x_j}}{\int_{\Omega} ((1 - \sum_{j=1}^{k-1} p_j)^{x_k} \prod_{j=1}^{k-1} p_j^{x_j}) dV} \quad (\text{B.2})$$

Firstly, let us prove Theorem 10.

**Theorem 10:**  $c(x_i, x_j, x_k, x_l, x_m) = c(x_j, x_k, x_l, x_i, x_m)$  for fixed  $x_k, x_l$  and  $x_m$ .

**Proof:** According to the definition of certainty in Eq. (B.1), obviously we have

$$c(x_i, x_j, x_k, x_l, x_m) = c(x_j, x_k, x_l, x_i, x_m)$$

for fixed  $x_k, x_l$  and  $x_m$ . □

Secondly, prior to presenting the detailed proof for Theorem 8, we introduce some lemmas.

**Lemma 2:**  $F_1(p_i)$  is increasing with respect to  $p_i$  when  $p_i \in [0, \frac{x_i}{x_k}(1 - \sum_{j=1}^{k-1} p_j)]$ , and decreasing with respect to  $p_i$  when  $p_i \in [\frac{x_i}{x_k}(1 - \sum_{j=1}^{k-1} p_j), 1]$ . Meanwhile,  $F_1(p_i)$  is maximized at  $p_i = \frac{x_i}{x_k}(1 - \sum_{j=1}^{k-1} p_j)$ .

**Proof:** Let

$$F_1(p_i) \triangleq C_1 \left(1 - \sum_{j=1}^{k-1} p_j\right)^{x_k} p_i^{x_i} \quad (\text{B.3})$$

The derivative of  $F_1(p_i)$  can be computed as

$$\begin{aligned} \frac{\partial F_1(p_i)}{\partial p_i} &= C_1 \left[ -x_k \left(1 - \sum_{j=1}^{k-1} p_j\right)^{x_k-1} p_i^{x_i} + x_i \left(1 - \sum_{j=1}^{k-1} p_j\right)^{x_k} p_i^{x_i-1} \right] \\ &= C_1 \left(1 - \sum_{j=1}^{k-1} p_j\right)^{x_k-1} p_i^{x_i-1} \left[ -x_k p_i + x_i \left(1 - \sum_{j=1}^{k-1} p_j\right) \right] \end{aligned}$$

When  $p_i \in [0, \frac{x_i}{x_k} (1 - \sum_{j=1}^{k-1} p_j)]$ , we have  $\frac{\partial F_1(p_i)}{\partial p_i} > 0$ , then  $F_1(p_i)$  is increasing with respect to  $p_i$ .

In contrast, when  $p_i \in [\frac{x_i}{x_k} (1 - \sum_{j=1}^{k-1} p_j), 1]$ , we have  $\frac{\partial F_1(p_i)}{\partial p_i} < 0$ , then  $F_1(p_i)$  is decreasing with respect to  $p_i$ .

Therefore, we have the maximum value of  $F_1(p_i)$  when  $p_i = \frac{x_i}{x_k} (1 - \sum_{j=1}^{k-1} p_j)$ .  $\square$

**Lemma 3:**  $\min F_1(p_i) = 0$ ;  $\max F_1(p_i) > 1$ .

**Proof:** From Lemma 2, we have

$$\min F_1(p_i) = \min\{F_1(0), F_1(1)\} = 0. \quad (\text{B.4})$$

According to the definition of integration, we know

$$\max\left\{\left(1 - \sum_{j=1}^{k-1} p_j\right)^{x_k} \prod_{j=1}^{k-1} p_j^{x_j}\right\} > \int_{\Omega} \left(\left(1 - \sum_{j=1}^{k-1} p_j\right)^{x_k} \prod_{j=1}^{k-1} p_j^{x_j}\right) dV \quad (\text{B.5})$$

i.e.

$$\max F_1(p_i) > 1. \quad (\text{B.6})$$

$\square$

**Lemma 4:** Given  $A$  and  $B$  defined by  $F_1(A) = F_1(B) = 1$ ,

$$0 < A < \frac{x_i}{x_k} \left(1 - \sum_{j=1}^{k-1} p_j\right) < B < 1,$$

we have

$$c = \int_0^1 \cdots \int_A^B \cdots \int_0^1 (F_1(p_i) - 1) dp_1 \cdots dp_i \cdots dp_{k-1}. \tag{B.7}$$

**Proof:** According to Lemma 2, we have

$$F_1(p_i) \begin{cases} > 1, & \text{when } A < p_i < B; \\ = 1, & \text{when } p_i = A \text{ or } B; \\ < 1, & \text{when } p_i < A \text{ or } p_i > B; \end{cases} . \tag{B.8}$$

As the cumulative probability of the probability distribution of  $V$  within  $\Omega$  must be 1, i.e.

$$\int_{\Omega} g(V) dV = 1, \tag{B.9}$$

we have

$$\int_0^1 \cdots \int_0^1 \cdots \int_0^1 (F_1(p_i) - 1) dp_1 \cdots dp_i \cdots dp_{k-1} = 0. \tag{B.10}$$

Hence,

$$\begin{aligned} & \int_0^1 \cdots \int_0^A \cdots \int_0^1 (F_1(p_i) - 1) dp_1 \cdots dp_i \cdots dp_{k-1} \\ & + \int_0^1 \cdots \int_B^1 \cdots \int_0^1 (F_1(p_i) - 1) dp_1 \cdots dp_i \cdots dp_{k-1} \\ & + \int_0^1 \cdots \int_A^B \cdots \int_0^1 (F_1(p_i) - 1) dp_1 \cdots dp_i \cdots dp_{k-1} = 0, \end{aligned} \tag{B.11}$$

i.e.

$$\begin{aligned}
& \int_0^1 \cdots \int_A^B \cdots \int_0^1 (F_1(p_i) - 1) dp_1 \cdots dp_i \cdots dp_{k-1} \quad (\text{B.12}) \\
&= \int_0^1 \cdots \int_0^A \cdots \int_0^1 (1 - F_1(p_i)) dp_1 \cdots dp_i \cdots dp_{k-1} \\
&+ \int_0^1 \cdots \int_B^1 \cdots \int_0^1 (1 - F_1(p_i)) dp_1 \cdots dp_i \cdots dp_{k-1}.
\end{aligned}$$

In addition,

$$\begin{aligned}
& \int_0^1 \cdots \int_0^1 \cdots \int_0^1 |F_1(p_i) - 1| dp_1 \cdots dp_i \cdots dp_{k-1} \quad (\text{B.13}) \\
&= \int_0^1 \cdots \int_0^A \cdots \int_0^1 (1 - F_1(p_i)) dp_1 \cdots dp_i \cdots dp_{k-1} \\
&+ \int_0^1 \cdots \int_B^1 \cdots \int_0^1 (1 - F_1(p_i)) dp_1 \cdots dp_i \cdots dp_{k-1} \\
&+ \int_0^1 \cdots \int_A^B \cdots \int_0^1 (F_1(p_i) - 1) dp_1 \cdots dp_i \cdots dp_{k-1}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
& \frac{1}{2} \int_{\Omega} |F_1(p_i) - 1| dV \quad (\text{B.14}) \\
&= \frac{1}{2} \int_0^1 \cdots \int_0^1 \cdots \int_0^1 |F_1(p_i) - 1| dp_1 \cdots dp_i \cdots dp_{k-1} \\
&= \int_0^1 \cdots \int_A^B \cdots \int_0^1 (F_1(p_i) - 1) dp_1 \cdots dp_i \cdots dp_{k-1}. \square
\end{aligned}$$

According to the above lemmas, Theorem 8 can be proved as follows.

**Theorem 8:** If the ratio of  $x_i : x_j$  ( $i \neq j$ ) is fixed, then the certainty of ratings increases with respect to the total number of ratings, given the fixed  $x_h$  ( $h \neq i, h \neq j$ ).

**Proof:** Let  $x_i : x_j \triangleq \alpha : 1$ , and  $sum \triangleq x_i + x_j$ . Hence,

$$x_i = \frac{\alpha}{\alpha + 1} sum \quad \text{and} \quad x_j = \frac{1}{\alpha + 1} sum. \quad (\text{B.15})$$

Let

$$F_1(p_i) \triangleq \frac{C_2(p_i)p_i^{x_i}p_j^{x_j}}{\int_{\Omega} C_2(p_i)p_i^{x_i}p_j^{x_j} dV} \quad (\text{B.16})$$

According to Eq. (B.1) and Lemma 4, we have

$$\begin{aligned} c'(sum) &= \int_0^1 \cdots \int_{A(x_i)}^{B(x_i)} \cdots \int_0^1 \frac{\partial}{\partial sum} \left( \frac{C_2(q_i)q_i^{x_i}q_j^{x_j}}{\int_{\Omega} C_2(p_i)p_i^{x_i}p_j^{x_j} dV} - 1 \right) dq_1 \cdots dq_i \cdots dq_{k-1} \\ &= \int_0^1 \cdots \int_{A(x_i)}^{B(x_i)} \cdots \int_0^1 \frac{\partial}{\partial sum} \left( \frac{C_2(q_i)q_i^{x_i}q_j^{x_j}}{\int_{\Omega} C_2(p_i)p_i^{x_i}p_j^{x_j} dV} \right) dq_1 \cdots dq_i \cdots dq_{k-1} \\ &= \frac{1}{\left( \int_{\Omega} C_2(p_i)p_i^{x_i}p_j^{x_j} dV \right)^2} \int_0^1 \cdots \int_{A(x_i)}^{B(x_i)} \cdots \int_0^1 \\ &\quad \left( \frac{\partial}{\partial sum} (C_2(q_i)q_i^{x_i}q_j^{x_j}) \int_{\Omega} C_2(p_i)p_i^{x_i}p_j^{x_j} dV \right. \\ &\quad \left. - C_2(q_i)q_i^{x_i}q_j^{x_j} \frac{\partial}{\partial sum} \int_{\Omega} C_2(p_i)p_i^{x_i}p_j^{x_j} dV \right) dq_1 \cdots dq_i \cdots dq_{k-1} \\ &= \frac{1}{\left( \int_{\Omega} C_2(p_i)p_i^{x_i}p_j^{x_j} dV \right)^2} \int_0^1 \cdots \int_{A(x_i)}^{B(x_i)} \cdots \int_0^1 \\ &\quad \left( C_2(q_i)q_i^{x_i}q_j^{x_j} \left( \frac{\alpha}{\alpha+1} \ln q_i + \frac{1}{\alpha+1} \ln q_j \right) \int_{\Omega} C_2(p_i)p_i^{x_i}p_j^{x_j} dV \right. \\ &\quad \left. - C_2(q_i)q_i^{x_i}q_j^{x_j} \int_{\Omega} C_2(p_i)p_i^{x_i}p_j^{x_j} \left( \frac{\alpha}{\alpha+1} \ln p_i + \frac{1}{\alpha+1} \ln p_j \right) dV \right) \\ &\quad dq_1 \cdots dq_i \cdots dq_{k-1} \\ &= \frac{1}{\left( \int_{\Omega} C_2(p_i)p_i^{x_i}p_j^{x_j} dV \right)^2} \int_0^1 \cdots \int_{A(x_i)}^{B(x_i)} \cdots \int_0^1 \int_{\Omega} \\ &\quad \left( C_2(q_i)q_i^{x_i}q_j^{x_j} C_2(p_i)p_i^{x_i}p_j^{x_j} \ln \frac{q_i^{\frac{\alpha}{\alpha+1}} q_j^{\frac{1}{\alpha+1}}}{p_i^{\frac{\alpha}{\alpha+1}} p_j^{\frac{1}{\alpha+1}}} \right) dV dq_1 \cdots dq_i \cdots dq_{k-1} \quad (\text{B.17}) \end{aligned}$$

As

$$p_i^{\frac{\alpha}{\alpha+1}} p_j^{\frac{1}{\alpha+1}} \quad (\text{B.18})$$

is maximized at  $p_i = \alpha p_j$ , according to Theorem 10 and Lemma 2, when  $q_i \in [A, B]$

and  $p_i \in [0, A] \cup [B, 1]$ , we have

$$q_i^{\frac{\alpha}{\alpha+1}} q_j^{\frac{1}{\alpha+1}} > p_i^{\frac{\alpha}{\alpha+1}} p_j^{\frac{1}{\alpha+1}} \quad (\text{B.19})$$

Thus,

$$\begin{aligned} & \int_0^1 \cdots \int_0^{A(x_i)} \cdots \int_0^1 \left( C_2(p_i) p_i^{x_i} p_j^{x_j} \ln \frac{q_i^{\frac{\alpha}{\alpha+1}} q_j^{\frac{1}{\alpha+1}}}{p_i^{\frac{\alpha}{\alpha+1}} p_j^{\frac{1}{\alpha+1}}} \right) dp_1 \cdots dp_i \cdots dp_{k-1} \\ & + \int_0^1 \cdots \int_{B(x_i)}^1 \cdots \int_0^1 \left( C_2(p_i) p_i^{x_i} p_j^{x_j} \ln \frac{q_i^{\frac{\alpha}{\alpha+1}} q_j^{\frac{1}{\alpha+1}}}{p_i^{\frac{\alpha}{\alpha+1}} p_j^{\frac{1}{\alpha+1}}} \right) dp_1 \cdots dp_i \cdots dp_{k-1} > 0; \end{aligned}$$

Hence, we have

$$\begin{aligned} & \int_0^1 \cdots \int_{A(x_i)}^{B(x_i)} \cdots \int_0^1 \int_0^1 \cdots \int_0^{A(x_i)} \cdots \int_0^1 \\ & \left( C_2(q_i) q_i^{x_i} q_j^{x_j} C_2(p_i) p_i^{x_i} p_j^{x_j} \ln \frac{q_i^{\frac{\alpha}{\alpha+1}} q_j^{\frac{1}{\alpha+1}}}{p_i^{\frac{\alpha}{\alpha+1}} p_j^{\frac{1}{\alpha+1}}} \right) dq_1 \cdots dq_i \cdots dq_{k-1} dp_1 \cdots dp_i \cdots dp_{k-1} \\ & + \int_0^1 \cdots \int_{A(x_i)}^{B(x_i)} \cdots \int_0^1 \int_0^1 \cdots \int_{B(x_i)}^1 \cdots \int_0^1 \\ & \left( C_2(q_i) q_i^{x_i} q_j^{x_j} C_2(p_i) p_i^{x_i} p_j^{x_j} \ln \frac{q_i^{\frac{\alpha}{\alpha+1}} q_j^{\frac{1}{\alpha+1}}}{p_i^{\frac{\alpha}{\alpha+1}} p_j^{\frac{1}{\alpha+1}}} \right) dq_1 \cdots dq_i \cdots dq_{k-1} dp_1 \cdots dp_i \cdots dp_{k-1} > 0 \end{aligned}$$

In addition,

$$\begin{aligned} & \int_0^1 \cdots \int_{A(x_i)}^{B(x_i)} \cdots \int_0^1 \int_0^1 \cdots \int_{A(x_i)}^{B(x_i)} \cdots \int_0^1 \\ & \left( C_2(q_i) q_i^{x_i} q_j^{x_j} C_2(p_i) p_i^{x_i} p_j^{x_j} \ln \frac{q_i^{\frac{\alpha}{\alpha+1}} q_j^{\frac{1}{\alpha+1}}}{p_i^{\frac{\alpha}{\alpha+1}} p_j^{\frac{1}{\alpha+1}}} \right) dq_1 \cdots dq_i \cdots dq_{k-1} dp_1 \cdots dp_i \cdots dp_{k-1} = 0. \end{aligned}$$

Therefore, from Eq. (B.17), we have  $c'(x_i) > 0$ , so  $c(x_i)$  is increasing with respect to  $x_i$ . □

In addition, before the proof of Theorem 9, let us introduce a lemma.

**Lemma 5:** If the summation of  $x_i$  and  $x_j$  ( $i \neq j$ ) is fixed (i.e.  $sum \triangleq x_i + x_j$  and  $sum$  is fixed), we have

$$c(x_i) = c(sum - x_i), \quad (\text{B.20})$$

given the fixed all the other number of rating except for  $x_i$  and  $x_j$ .

**Proof:**  $A$  and  $B$  are two roots of equation

$$C_1 \left(1 - \sum_{j=1}^{k-1} p_j\right)^{sum-x_i} p_i^{x_i} = \frac{(sum - x_i)! x_i!}{(sum + 1)!} \quad (\text{B.21})$$

As

$$C_1 \left(1 - \sum_{j=1}^{k-1} p_j\right)^{sum-x_i} p_i^{x_i} \quad (\text{B.22})$$

is symmetrical with  $x_i = \frac{sum}{2}$ , and

$$\frac{(sum - x_i)! x_i!}{(sum + 1)!} \quad (\text{B.23})$$

is symmetrical with  $x_i = \frac{sum}{2}$ , we have

$$B(x_i) - A(x_i) = B(sum - x_i) - A(sum - x_i), \quad (\text{B.24})$$

i.e.

$$c(x_i) = c(sum - x_i). \quad (\text{B.25})$$

□

**Theorem 9:** If  $sum = x_i + x_j$  ( $i \neq j$ ) is fixed, given the fixed  $x_h$  ( $h \neq i, h \neq j$ ), the certainty of ratings is increasing with respect to  $x_i$  when  $x_i < sum/2$ ; otherwise, the certainty of ratings is decreasing with respect to  $x_i$  when  $x_i > sum/2$ .

**Proof:** From the certainty defined in Eq. (B.1), certainty is evaluated based on the deviation from the prior distribution (uniform distribution). Hence, as the function

$$C_1(1 - \sum_{j=1}^{k-1} p_j)^{sum-x_i} p_i^{x_i} \quad (\text{B.26})$$

has a bell shape, the certainty  $c$  decreases with respect to

$$C_3(p_{max})p_{max}^{x_i}, \quad (\text{B.27})$$

where the function in Eq. (B.26) is maximized at

$$p_{max} = \frac{x_i}{sum - x_i} (1 - \sum_{j=1}^{k-1} p_j). \quad (\text{B.28})$$

As

$$\frac{\partial C_1(1 - \sum_{j=1}^{k-1} p_j)^{sum-x_i} p_i^{x_i}}{\partial x_i} = C_1(1 - \sum_{j=1}^{k-1} p_j)^{sum-x_i} p_i^{x_i} \ln \frac{x_i}{sum - x_i}, \quad (\text{B.29})$$

we can obtain that

$$C_1(1 - \sum_{j=1}^{k-1} p_j)^{sum-x_i} p_i^{x_i}$$

is  $\begin{cases} \text{decreasing with respect to } x_i, & \text{when } 0 < x_i < \frac{sum}{2} \\ \text{increasing with respect to } x_i, & \text{when } \frac{sum}{2} < x_i < sum \end{cases} \quad (\text{B.30})$

Hence,

$$C_3(p_{max})p_{max}^{x_i}$$

is  $\begin{cases} \text{decreasing with respect to } x_i, & \text{when } 0 < x_i < \frac{sum}{2} \\ \text{increasing with respect to } x_i, & \text{when } \frac{sum}{2} < x_i < sum \end{cases} \quad (\text{B.31})$

Therefore, we have

$$c(x_i) \text{ is } \begin{cases} \text{increasing with respect to } x_i, & \text{when } 0 < x_i < \frac{sum}{2} \\ \text{decreasing with respect to } x_i, & \text{when } \frac{sum}{2} < x_i < sum \end{cases} \quad (\text{B.32})$$

□

Finally, let us prove Theorem 11.

**Theorem 11:** The certainty of ratings increases with respect to  $x_i$ , the number of occurrences of rating  $r_i$ , given all the fixed  $x_h$  ( $h \neq i$ ).

**Proof:** Let

$$F_1(p_i) \triangleq \frac{C_3(p_i)p_i^{x_i}}{\int_{\Omega} C_3(p_i)p_i^{x_i} dV} \quad (\text{B.33})$$

From the certainty defined in Eq. (B.1), the certainty is evaluated based on the deviation from the prior distribution (uniform distribution). Hence, as the function

$$C_1(1 - \sum_{j=1}^{k-1} p_j)^{x_k} p_i^{x_i} \quad (\text{B.34})$$

has a bell shape, the certainty  $c$  decreases with respect to

$$C_3(p_{max})p_{max}^{x_i}, \quad (\text{B.35})$$

where the function in Eq. (B.34) is maximized at

$$p_{max} = \frac{x_i}{x_k} (1 - \sum_{j=1}^{k-1} p_j). \quad (\text{B.36})$$

As  $p_{max}$  decreases with respect to  $x_i$ ,  $c(x_i)$  increases with respect to  $x_i$ . □



---

# Bibliography

---

- [1] *eBay*. <http://www.eBay.com/>.
- [2] *Epinions*. <http://www.epinions.com/>.
- [3] *GNutella*. <http://www.gnutella.com/>.
- [4] *PayPal*. <http://www.paypal.com.au/>.
- [5] *YouTube*. <http://www.youtube.com/>.
- [6] P. S. Adler. Market, hierarchy, and trust: The knowledge economy and the future of capitalism. *Organization Science*, 12(2):215–234, 2001.
- [7] R. Ashri, S. D. Ramchurn, J. Sabater, M. Luck, and N. R. Jennings. Trust evaluation through relationship analysis. In *AAMAS 2005*, pages 1005–1011, 2005.
- [8] A. T. Azar. *Fuzzy Systems*. Intech, 2010.
- [9] M. Bakhtin. *Speech Genres and Other Late Essays*. University of Texas Press, 1986.
- [10] B. Barber. *The logic and limits of trust*. Rutgers University Press, 1983.
- [11] S. D. Boon and J. G. Holmes. The dynamics of interpersonal trust: Resolving uncertainty in the face of risk. In R. Hinde and J. Groebel, editors, *Cooperation and Prosocial Behavior*, pages 167–182. Cambridge Univ. Press, 1991.
- [12] M. B. Brewer. In-group favoritism: the subtle side of intergroup discrimination. In D. M. Messick and A. E. Tenbrunsel, editors, *Codes of Conduct: Behavioral Research and Business Ethics*, pages 160–171. Russell Sage Found, 1996.

- [13] R. S. Burt and M. Knez. Kinds of third-party effects on trust. *Rationality and Society*, 7:255–292, 1995.
- [14] C. Castelfranchi and R. Falcone. Trust is much more than subjective probability: Mental components and sources of trust. In *HICSS 2000*, 2000.
- [15] J. Coleman. *Foundations of Social Theory*. Belknap Press of Harvard University Press, 1998.
- [16] R. Collins. *Sociological Insight: an Introduction to Non-obvious Sociology*. Oxford University Press, 1992.
- [17] W. Conner, A. Iyengar, T. A. Mikalsen, I. Rouvellou, and K. Nahrstedt. A trust management framework for service-oriented environments. In *WWW*, pages 891–900, 2009.
- [18] D. W. Creed and R. E. Miles. Trust in organizations: a conceptual framework linking organizational forms, managerial philosophies, and the opportunity costs of controls. In R. Kramer and T. Tyler, editors, *Trust in organizations: Frontiers of Theory and Research*, pages 16–38. Sage Publications, 1996.
- [19] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *ACM Conference on Computer and Communications Security (CCS 2002)*, pages 207–216, 2002.
- [20] E. Damiani, S. D. C. di Vimercati, P. Samarati, and M. Viviani. A wowa-based aggregation technique on trust values connected to metadata. *Electr. Notes Theor. Comput. Sci.*, 157(3):131–142, 2006.
- [21] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [22] E. Erikson. *Identity: Youth and Crisis*. W. W. Norton & Company, 1968.

- 
- [23] S. Feng and L. D. Xu. Decision support for fuzzy comprehensive evaluation of urban development. *Fuzzy Sets and Systems*, 105(1):1–12, 1999.
- [24] J. Gentle, W. Härdle, and Y. Mori. *Handbook of Computational Statistics*. Springer, 2004.
- [25] J. Golbeck and J. A. Hendler. Inferring binary trust relationships in web-based social networks. *ACM Trans. Internet Techn.*, 6(4):497–529, 2006.
- [26] N. Griffiths. Task delegation using experience-based multi-dimensional trust. In *AAMAS 2005*, pages 489–496, 2005.
- [27] J. Gross and J. Yellen. *Handbook of Graph Theory*. CRC Press, 2003.
- [28] R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW 2004*, pages 403–412, 2004.
- [29] J. E. Haddad, M. Manouvrier, G. Ramirez, and M. Rukoz. QoS-driven selection of web services for transactional composition. In *ICWS 2008*, pages 653–660, 2008.
- [30] M. S. Hamada, A. G. Wilson, C. S. Reese, and H. F. Martz. *Bayesian Reliability*. Springer, 2008.
- [31] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2000.
- [32] C.-W. Hang, Y. Wang, and M. P. Singh. Operators for propagating trust and their evaluation in social networks. In *AAMAS (2)*, pages 1025–1032, 2009.
- [33] W. W. Hines, D. C. Montgomery, D. M. Goldsman, and C. M. Borror. *Probability and Statistics in Engineering*. John Wiley & Sons, Inc, 2003.
- [34] A. Hójek. *Probability, Logic, and Probability Logic*, pages 362–384. The Blackwell Guide to Philosophical Logic. Blackwell Publishing, 2001.

- [35] N. Hu, P. A. Pavlou, and J. Zhang. Can online reviews reveal a product's true quality?: empirical findings and analytical modeling of online word-of-mouth communication. In *ACM EC 2006*, pages 324–330, 2006.
- [36] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.
- [37] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, 1997.
- [38] R. Jeffrey. *Subjective Probability: The Real Thing*. Cambridge University Press, April 2004.
- [39] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.
- [40] A. Jøsang. Subjective evidential reasoning. In *IPMU*, 2002.
- [41] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.
- [42] R. Jurca and B. Faltings. Minimum payments that reward honest reputation feedback. In *ACM EC 2006*, pages 190–199, 2006.
- [43] R. Jurca and B. Faltings. Collusion-resistant, incentive-compatible feedback payments. In *ACM EC 2007*, pages 200–209, 2007.
- [44] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW 2003*, pages 640–651, 2003.
- [45] D. H. Knight and N. L. Chervany. The meaning of trust. Technical Report WP9604, University of Minnesota, Management Information Systems Research Center, 1996.

- 
- [46] T. Korkmaz and M. Krunz. Multi-constrained optimal path selection. In *INFO-COM 2001*, pages 834–843, 2001.
- [47] R. F. Korte. Biases in Decision Making and Implications for Human Resource Development. *Advances in Developing Human Resources*, 5(4):440–457, 2003.
- [48] B. Kosko. *Fuzzy Engineering*. Prentice Hall, 1997.
- [49] R. M. Kramer. Trust and distrust in organizations: Emerging perspectives, enduring questions. *Annual Review of Psychology*, 50:569–598, 1999.
- [50] R. J. Lewicki and B. B. Bunker. Trust in relationships: a model of trust development and decline. In B. Bunker and J. Rubin, editors, *Conflict, Cooperation, and Justice*. Jossey-Bass, 1995.
- [51] L. Li and Y. Wang. A trust vector approach to service-oriented applications. In *ICWS 2008*, pages 270–277, 2008.
- [52] L. Li and Y. Wang. Trust evaluation in composite services selection and discovery. In *IEEE SCC 2009*, pages 482–485, 2009.
- [53] L. Li and Y. Wang. Context based trust normalization in service-oriented environments. In *ATC 2010*, pages 122–138, 2010.
- [54] L. Li and Y. Wang. Subjective trust inference in composite services. In *AAAI 2010*, pages 1377–1384, 2010.
- [55] L. Li and Y. Wang. A subjective probability based deductive approach to global trust evaluation in composite services. In *ICWS 2011*, pages 604–611, 2011.
- [56] L. Li and Y. Wang. The study of trust vector based trust rating aggregation in service-oriented environments. *World Wide Web*, In press, 2012.
- [57] L. Li, Y. Wang, and E.-P. Lim. Trust-oriented composite service selection and discovery. In *ICSOC/ServiceWave 2009*, pages 50–67, 2009.

- [58] L. Li, Y. Wang, and E.-P. Lim. Trust-oriented composite service selection with qos constraints. *J. UCS*, 16(13):1720–1744, 2010.
- [59] L. Li, Y. Wang, and V. Varadharajan. Fuzzy regression based trust prediction in service-oriented applications. In *ATC 2009*, pages 221–235, 2009.
- [60] G. Liu, Y. Wang, and M. A. Orgun. Optimal social trust path selection in complex social networks. In *AAAI 2010*, pages 1391–1398, 2010.
- [61] Z. Malik and A. Bouguettaya. RATEWeb: Reputation assessment for trust establishment among web services. *VLDB J.*, 18(4):885–911, 2009.
- [62] J. G. March. *Primer on Decision Making: How Decisions Happen*. Free Press, 1994.
- [63] I. Marková. *Trust and Democratic Transition in Post-Communist Europe*. Oxford University Press, 2004.
- [64] I. Marková, A. Gillespie, and J. Valsiner. *Trust and Distrust: Sociocultural Perspectives*. Information Age Publishing, 2008.
- [65] S. Marti and H. Garcia-Molina. Limited reputation sharing in p2p systems. In *ACM EC 2004*, pages 91–101, 2004.
- [66] M. H. Mashinchi, L. Li, M. A. Orgun, and Y. Wang. The prediction of trust rating based on the quality of services using fuzzy linear regression. In *FUZZ-IEEE 2011*, pages 1953–1959, 2011.
- [67] B. Medjahed and A. Bouguettaya. A multilevel composability model for semantic web services. *IEEE Trans. Knowl. Data Eng.*, 17(7):954–968, 2005.
- [68] B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid. Composing web services on the semantic web. *VLDB J.*, 12(4):333–351, 2003.

- 
- [69] D. A. Menascé. Composing web services: A QoS view. *IEEE Internet Computing*, 8(6):88–90, 2004.
- [70] B. Misztal. *Trust in Modern Societies: The Search for the Bases of Social Order*. Polity Press, 1996.
- [71] D. P. Morton and E. Popova. Monte-Carlo simulations for stochastic optimization. In *Encyclopedia of Optimization*, pages 2337–2345. Springer, 2009.
- [72] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. Service-oriented computing: a research roadmap. *Int. J. Cooperative Inf. Syst.*, 17(2):223–255, 2008.
- [73] S. P. Ponzetto and R. Navigli. Large-scale taxonomy mapping for restructuring and integrating wikipedia. In *IJCAI 2009*, pages 2083–2088, 2009.
- [74] S. P. Ponzetto and M. Strube. Deriving a large-scale taxonomy from wikipedia. In *AAAI 2007*, pages 1440–1445, 2007.
- [75] S. Rao. *Applied Numerical Methods for Engineers and Scientists*. Prentice Hall, 2002.
- [76] J. B. Rotter. Interpersonal trust, trustworthiness, and gullibility. *American Psychologist*, 35(1):1–7, 1980.
- [77] J. Sabater and C. Sierra. REGRET: reputation in gregarious societies. In *Agents 2001*, pages 194–195, 2001.
- [78] A. B. Seligman. *The Problem of Trust*. Princeton University Press, 2000.
- [79] G. Simmel. *The Sociology of Georg Simmel*. The Free Press, 1950.
- [80] S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok. Trusted p2p transactions with fuzzy reputation aggregation. *IEEE Internet Computing*, 9(6):24–34, 2005.

- [81] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *KDD 2008*, pages 990–998, 2008.
- [82] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.
- [83] B. Uzzi. Social structure and competition in interfirm networks: The paradox of embeddedness. *Administrative Science Quarterly*, 42(1):35–67, 1997.
- [84] P. Victor, C. Cornelis, M. D. Cock, and P. P. da Silva. Gradual trust and distrust in recommender systems. *Fuzzy Sets and Systems*, 160(10):1367–1382, 2009.
- [85] L.-H. Vu, M. Hauswirth, and K. Aberer. QoS-based service selection and ranking with trust and reputation management. In *CoopIS 2005*, pages 466–483, 2005.
- [86] F. E. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16(1):57–74, 2008.
- [87] Y. Wang and L. Li. Two-dimensional trust rating aggregations in service-oriented applications. *IEEE Trans. Services Computing*, 4(4):257–271, 2011.
- [88] Y. Wang and E.-P. Lim. The evaluation of situational transaction trust in e-service environments. In *ICEBE 2008*, pages 265–272, 2008.
- [89] Y. Wang and K.-J. Lin. Reputation-oriented trustworthy computing in e-commerce environments. *IEEE Internet Computing*, 12(4):55–59, 2008.
- [90] Y. Wang, K.-J. Lin, D. S. Wong, and V. Varadharajan. Trust management towards service-oriented applications. *Service Oriented Computing and Applications*, 3(2):129–146, 2009.

- 
- [91] Y. Wang and M. P. Singh. Formal trust model for multiagent systems. In *International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 1551–1556, 2007.
- [92] Y. Wang and V. Varadharajan. Interaction trust evaluation in decentralized environments. In *EC-Web 2004*, pages 144–153, 2004.
- [93] R. A. Wilson and F. C. Keil. *The MIT encyclopedia of the cognitive sciences*. The MIT Press, 1999.
- [94] J. Xiao and R. Boutaba. QoS-aware service composition and adaptation in autonomic communication. *IEEE Journal on Selected Areas in Communications*, 23(12):2344–2360, 2005.
- [95] L. Xiong and L. Liu. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. Knowl. Data Eng.*, 16(7):843–857, 2004.
- [96] Z. Xu, P. Martin, W. Powley, and F. Zulkernine. Reputation-enhanced QoS-based web services discovery. In *ICWS 2007*, pages 249–256, 2007.
- [97] B. Yu and M. P. Singh. An evidential model of distributed reputation management. In *AAMAS 2002*, pages 294–301, 2002.
- [98] B. Yu, M. P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. *IEEE Symposium on Multi-Agent Security and Survivability*, pages 1–10, 2004.
- [99] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman. Sybilguard: defending against sybil attacks via social networks. *IEEE/ACM Trans. Netw.*, 16(3):576–589, 2008.
- [100] H. Yu, C. Shi, M. Kaminsky, P. B. Gibbons, and F. Xiao. Dsybil: Optimal sybil-resistance for recommendation systems. In *IEEE Symposium on Security and*

- Privacy*, pages 283–298, 2009.
- [101] T. Yu, Y. Zhang, and K.-J. Lin. Efficient algorithms for web services selection with end-to-end QoS constraints. *TWEB*, 1(1), 2007.
- [102] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.
- [103] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng. Quality driven web services composition. In *WWW 2003*, pages 411–421, 2003.
- [104] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. QoS-aware middleware for web services composition. *IEEE Trans. Software Eng.*, 30(5):311–327, 2004.
- [105] R. Zhou and K. Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Trans. Parallel Distrib. Syst.*, 18(4):460–473, 2007.