A Computational Game-Theoretic Study of Reputation

Chang Yan

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2014

To my grandfathers

Acknowledgements

First of all I would like to thank my supervisors Professor Luke Ong and Professor John Quah for their continuous support, kindness and patience in me. I am deeply grateful for the time and effort they spent on me over the last few years. I would like to thank my collaborator Dr. Edoardo Gallo, for giving me valuable guidance, insight and criticism. He is also a friend, whose enthusiasm and grit moved me forward. I further thank Professor Angel Sánchez for providing helpful comments on my work. I would also like to extend my gratitude to Professor Samson Abramsky, Dr. Alexandru Baltag, Professor Michael Wooldridge and Dr. Collin Raymond, who were the examiners for my transfer of status and confirmation of status. They offered thoughtful feedback and suggestions that helped shape this thesis.

My life as a graduate student would be incomplete without the group of the people whom I shared the office with. I thank Robin Neatherway, Martin Lester, Markus Aschinger, Michael Vanden Boom, Steven Ramsay, Christopher Broadbent, David Hopkins, Emanuele D'Osualdo, Jonathan Kochems, Marcelo Sousa and Egor Ianovski for the time we spent together and all the conversations we had. I learnt a great deal from them, and loved the bad jokes they told.

Finally, I am forever indebted to my parents and my fiancée for their love and inspiration. This work would not be possible without them.

Abstract

As societies become increasingly connected thanks to advancing technologies and the Internet in particular, individuals and organizations (i.e. agents hereafter) engage in innumerable interaction and face constantly the possibilities thereof. Such unprecedented connectivity offers opportunities through which social and economic benefits are realised and disseminated. Nonetheless, risky and damaging interaction abound. To promote beneficial relationships and to deter adverse outcomes, agents adopt different means and resources. This thesis focuses on reputation as a crucial mechanism for promoting positive interaction, and examines the topic from game-theoretic perspective using computational methods.

First, we investigate the design of reputation systems by incorporating economic incentives into algorithm design. Focusing on ubiquitous user-generated ratings on the Internet, we propose a truthful reputation mechanism that not only enforces honest reporting from individual raters but also takes into account their personal preferences. The mechanism is constructed using a blend of Bayesian Truth Serum and SimRank algorithms, both specifically adapted for our use case of online ratings. We show that the resulting mechanism is Bayesian incentive compatible and is computable in polynomial time. In addition, the mechanism is shown to be resistant to common manipulations on the Internet such as uniform fake ratings and targeted collusions. Lastly, we discuss detailed considerations for implementing the mechanism in practice.

Second, we investigate experimentally the relative importance of reputational and social knowledge in sustaining cooperation in dynamic networks. In our experiments, U.S-based subjects play a repeated game where, in each round, an endogenous network is formed among a group of 13 players and each player chooses a cooperative or non-cooperative action that applies to all her connections. We vary the availability of reputational and social knowledge to subjects in 4 treatments. At the aggregate level,

we find that reputational knowledge is of first-order importance for supporting cooperation, while social knowledge plays a complementary role only when reputational knowledge is available. Further community-level analysis reveals that reputational knowledge leads to the emergence of highly cooperative hubs, and a dense and cluster network, while social knowledge enhances cooperation by forming a large, dense and clustered community of cooperators who exclude outsiders through link removals and link refusals. At the individual level, reputational knowledge proves essential for the emergence of network structural characteristics that are associated with cooperative actions. In contrast, in treatments without reputational information, none of the network metrics is predicative of subjects' choices of action.

Furthermore, we present UbiquityLab, a pioneering online platform for conducting real-time interactive experiments for game-theoretic studies. UbiquityLab supports both synchronous and asynchronous game models, and allows for complex and customisable interaction between subjects. It offers both back-end and front-end infrastructure with a modularised design to enable rapid development and streamlined operation. For instance, in synchronous mode, all per-stage and inter-stage logic are fully encapsulated by a thin server-side module, while a suite of client-side components eases the creation of game interface. The platform features a robust messaging protocol, such that player connection and game states are restored automatically upon networking errors and dropped out subjects are seamlessly substituted by customisable program players. Online experiments enjoy clear advantages over lab equivalents as they benefit from low operation cost, efficient execution, large and diverse subject pools, etc. UbiquityLab aims to promote online experiments as an emerging research methodology in experimental economics by bringing its benefits to other researchers.

Contents

1 Introduction **1**

1.1 Overview and Motivation 2

1.1.1 Reputation Mechanisms 2

1.1.2 Reputation and Cooperation 4

1.1.3 Experimentation 6

1.2 Thesis Organisation 8

2 Background **9**

2.1 An Economic View of Reputation 9

2.1.1 Bayesian Beliefs 10

2.1.2 Repeated Interactions 11

2.2 Game Theory and Mechanism Design 12

2.2.1 Basic Definitions 12

2.2.2 Solution Concepts 13

2.2.3 Mechanism Design 15

2.3 Reputation Mechanisms 16

2.3.1 A Graph Model 17

2.3.2 Common Manipulations 17

2.3.3 Selected Mechanisms 18

2.4 Experimental Research 20

2.4.1 Experimental Computer Science 21

2.4.2 Experimental Economics 22

2.4.3 Basics of Lab Experimentation 24

3 A Truthful Reputation Mechanism for Personalised Online Ratings **28**

3.1 Online Ratings 29

3.2 Related Work 30

3.3 The Settings 31

3.3.1 A SimRank Variant . . . 32
3.3.2 Bayesian Truth Serum . . . 35
3.4 The Mechanism . . . 39
3.4.1 Design . . . 39
3.4.2 Computation . . . 44
3.4.3 Theoretical Properties . . . 46
3.4.4 Pragmatic Considerations . . . 49
3.5 Discussion . . . 51

4 Synergistic effects of reputational and social knowledge on cooperation 53
4.1 Related Studies . . . 55
4.2 Motivation . . . 57
4.2.1 Reputation and Cooperation . . . 57
4.2.2 Network and Cooperation . . . 57
4.2.3 Reputational and Social Knowledge . . . 58
4.3 Experiments . . . 59
4.3.1 Method and Design . . . 60
4.3.2 Analysis and Results . . . 67
4.4 Discussion . . . 79

5 *UbiquityLab*: An Online Platform for Game Theoretic Experiments 82
5.1 Online Experiments . . . 83
5.1.1 Introduction . . . 83
5.1.2 Methods and Tools for Online Experimentation . . . 85
5.2 Functionalities and Usage . . . 89
5.2.1 Main Functionalities . . . 89
5.2.2 Differentiating Features . . . 91
5.2.3 Recommended Workflow and Usage . . . 94
5.3 Technical Implementation . . . 99
5.3.1 Platform Architecture . . . 100
5.3.2 Messaging Protocol and Server-side Logic . . . 102
5.3.3 Customisation via Modular API . . . 107
5.4 Future Roadmap . . . 112

6 Conclusion **115**
6.1 Summary 115
6.2 Future Directions 116

A Supplementary Information of the Experiment **119**
A.1 Subject Sample 119
A.2 Subject Participation 121
A.3 Robustness Checks 124
A.4 Experimental Instructions (for treatment B) 126

B Other Experiment Screenshots **142**

C Implementation of Core Components for Synchronous Games **145**
C.1 PlayGraph 145
C.2 ActionCache 148

Bibliography **150**

List of Figures

3.1 A minimal example of the graph model G of a rating website 33

4.1 Interactive network figure of the links among all the subjects in the RN treatment. 63

4.2 Three-panel design of the game interface. The screenshot shown here is stage 1 of a round in the RN treatment. 64

4.3 In stage 1 the third block in the right panel displays other subjects' previous actions and allows subjects to propose and cut links. The example screenshot above is what a subject would see in treatments R and RN with global reputational information. A green box with an "A" indicates that the subject has chosen to cooperate, a blue box with a "B" indicates that the subject has chosen to defect, and a gray box denotes that the subject has not taken an action. For instance, in the screenshot above we have that in the previous round subject H chose to defect and N chose to cooperate. Notice that if a subject does not take an action then the computer selects the defect action by default. 65

4.4 In stage 2 the third block in the right panel displays the reputational information of other subjects and allows subjects to approve or reject other subjects' link proposals 66

4.5 In stage 3 the bottom block allows subjects to choose between "A" (cooperate) and "B" (defect) 67

4.6 Cooperation level **(a)**, average payoff **(b)**, density **(c)**, and clustering **(d)** over 13 rounds of play for treatments B (yellow), N (red), R (green) and RN (blue). Analyses are based on aggregate data after round 5 in each session. Density denotes the number of links as a proportion of the 78 potential links in the network. Global clustering denotes the number of triangles (3 nodes connected by 3 links) as a proportion of the 858 potential triangles in the network. These are the results for Rank-sum tests comparing the differences across treatments, we only include significant ($P < 0.05$) and close to significant ($P < 0.1$) findings: **(a)** Cooperation level: RN vs. B, $P = 0.013$; RN vs. N, $P = 0.025$; R vs. B, $P = 0.047$; R vs. N, $P = 0.064$; **(b)** Average payoff: RN vs. B, $P = 0.013$; RN vs. N, $P = 0.025$; R vs. B, $P = 0.048$; R vs. N, $P = 0.048$; **(c)** Density: RN vs. B, $P = 0.002$; RN vs. N, $P = 0.013$; R vs. B, $P = 0.025$; R vs. N, $P = 0.048$; **(d)** Global clustering: RN vs. B, $P = 0.006$; RN vs. N, $P = 0.009$; R vs. B, $P = 0.064$; R vs. N, $P = 0.064$ . . 68

4.7 (a) Distribution of the number of detected communities in each treatment: the percentages are calculated using $\frac{\text{number of occurrences}}{13\times7}$, where 13 is the total number of rounds in each of the 7 sessions in a treatment. (b) Distribution of the mean sizes of communities in each treatment: the labels (e.g. 1, 2, 3, etc) on the x-axis within each treatment denote communities (e.g. $C1$, $C2$, $C3$, etc). 71

4.8 Cooperation level **(a)**, density **(b)**, and clustering **(c)** of the *largest* and the *second largest* communities for treatments B (yellow), N (red), R (green) and RN (blue). Analyses are based on aggregated data after round 5 in each session. Error bars indicate ± one SEM. These are the results for rank-sum tests comparing differences across and within treatments, we only include significant ($P < 0.05$) and close to significant ($P < 0.1$) findings. Between treatments for $C1$: **(a)** Cooperation level: RN vs. B, $P = 0.035$; RN vs. N, $P = 0.025$; R vs. B, $P = 0.085$; R vs. N, $P = 0.096$; **(b)** Density: RN vs. B, $P = 0.018$; RN vs. N, $P = 0.025$; R vs. B, $P = 0.048$; **(c)** Global clustering: RN vs. B, $P = 0.018$; RN vs. N, $P = 0.018$; R vs. B, $P = 0.048$; R vs. N, $P = 0.064$; Community size: RN vs. B, $P = 0.01$; RN vs. N, $P = 0.064$; R vs. B, $P = 0.064$. Between $C1$ and $C2$ within each treatment: **(a)** Cooperation level: RN, $P = 0.025$. 73

4.9 Links removed **(a)**, link removals received **(b)**, and proposals rejected **(c)** of the *largest* and the *second largest* communities for treatments B (yellow), N (red), R (green) and RN (blue). Analyses are based on aggregated data after round 5 in each session. Error bars indicate ± one SEM. These are the results for rank-sum tests comparing differences across and within treatments, we only include significant ($P < 0.05$) and close to significant ($P < 0.1$) findings. Between treatments for $C1$: Links removed: **(a)** RN vs. N, $P = 0.064$; R vs. N, $P = 0.048$; R vs. B, $P = 0.035$; **(b)** Removals received: R vs. N, $P = 0.064$; **(c)** Proposals rejected: RN vs. B, $P = 0.064$; RN vs. N, $P = 0.018$. Between treatments for $C2$: **(a)** Links removed: RN vs. B, $P = 0.048$; RN vs. N, $P = 0.064$. Between $C1$ and $C2$ within each treatment: **(a)** Links removed: RN, $P = 0.018$; **(b)** Removals received: RN, $P = 0.002$; R, $P = 0.096$; **(c)** Proposals rejected: RN, $P = 0.018$. 74

4.10 Snapshots showing the evolution of the network and cooperative behaviour for representative sessions. Each column is a representative session for each treatment, from left to right: B, N, R and RN. Each row is a round: 3, 7 and 11 from top to bottom. The size of a node is proportional to its degree. The colour of a node corresponds to the number of cooperative actions in the previous 5 rounds: blue (0 or 1 cooperative actions), light blue (2), light green (3), and green (4 or 5). Red bubbles distinguish $C1$ communities, while blue bubbles distinguish $C2$ communities. The labels next to the bubbles list the fictitious IDs of the members of the corresponding communities. Both the bubbles and the labels apply to the first two largest communities only. 79

5.1 An optimised workflow of conducting online experiments using UbiquityLab 95

5.2 A simplified overview of the architecture of UbiquityLab 100

5.3 Client-server messaging protocol for synchronous games 105

A.1 Percentages of subjects of different age groups (left) and education levels (right) for the 364 participants. Classification of education levels: 1 = high school or less, 2 = some college, 3 = 2-year college, 4 = 4-year college, 5 = master, 6 = professional degree (e.g. MD) or PhD. 121

A.2 Usage of the interactive features of the network figure in treatments (a) N and (b) RN. The blue line shows the evolution of the usage of hovering over a node during the game, while the red line shows the evolution of the usage of node dragging (as a percentage of instances of hovering over a node). 123

A.3 Cooperation level (a), average payoff (b), density (c), and clustering (d) over 13 rounds of play for treatments B (yellow), N (red), R (green) and RN (blue) after including "drop-out" sessions. Analyses are based on aggregated data after round 5 in each session. These are the results for Rank-sum tests comparing the differences across treatments, we only include significant ($P < 0.05$) and marginally significant ($P < 0.1$) findings: (a) Cooperation level: RN vs. B, $P = 0.008$; RN vs. N, $P = 0.015$; R vs. B, $P = 0.028$; R vs. N, $P = 0.037$. (b) Average payoff: RN vs. B, $P = 0.011$; RN vs. N, $P = 0.021$; R vs. B, $P = 0.049$; R vs. N, $P = 0.049$. (c) Density: RN vs. B, $P = 0.001$; RN vs. N, $P = 0.008$; R vs. B, $P = 0.015$; R vs. N, $P = 0.028$. (d) Global clustering: RN vs. B, $P = 0.004$; RN vs. N, $P = 0.006$; R vs. B, $P = 0.037$; R vs. N, $P = 0.037$. 125

B.1 At the start of each round, subject observes her random assignment to a node 142

B.2 After 5 seconds into each round, subject is able to make a choice by placing the bar on the scale 143

B.3 After everyone inputs a choice, subject sees the result of the round . . 143

B.4 Same experiment in a different network treatment 144

List of Tables

4.1 Comparison of experimental designs between previous studies 55
4.2 Stability metrics for communities $C1$ and $C2$ in each treatment. Mean values of the *C1 Stable* and *Any Stable* metrics in rounds $[6, 13]$. Size is the mean size of the communities in each treatment. By definition the *C1 Stable* metric is only available for the $C1$ community. 72
4.3 Determinants of subjects' actions 76

A.1 Summary statistics and pairwise correlations for the main variables. . 120

Chapter 1

Introduction

In Akerlof's "the market of lemons" [Ackerloff, 1970], a pessimistic picture was painted for a market in a classic asymmetric information setting. Sellers have some private information about the value of the goods being traded, while buyers are uninformed. Given limited signal about the quality of the goods, buyers are unwilling to offer a high price. The low price in turn discourages sellers of high-quality goods, thus the quality in the market further declines until nothing but the lowest quality goods remain. Akerlof noted that the presence of cheaters or dishonesty may drive the market out of existence by severely reducing its efficiency. Fortunately, the rise of markets like eBay shows that there are viable solutions to the problem. One of such solutions is reputation, which narrows the informational gap between sellers and buyers. In this context, reputation informs potential trading partners and enhances the trust so that transactions can occur at appropriate prices and market efficiency is attained.

Turning our attention to social interactions, it is prevalent in societies that collective aims are often at odds with private interests, the so-called social dilemma. The 2-person prisoner's dilemma prescribes that defection is the rational action to take even though mutual cooperation is the socially desirable outcome. In the face of the temptation of maximising self-interest, explicit mechanisms are necessary to drive cooperation on which our communities flourish. Reputation, again, offers an effective resolution. Human interactions are rarely one-shot and never static. In re-

peated interactions with others, reputation lives on even if we frequently change whom to interact with. Maintaining sound reputation for cooperation pays for long-term benefits, whereas near-term exploitation may ultimately prove myopic. As such, reputation effect is able to support cooperative equilibria in repeated dilemmas [Friedman, 1971, Kreps et al., 1982, Friedman, 1985].

This thesis studies reputation in the vital contexts illustrated above. In short, reputation facilitates trades and underpins the functioning of markets; reputation restraints exploitation of self-interest and supports cooperation.

1.1 Overview and Motivation

We present three research works. Primarily, the first is a theoretical study of a proposed reputation mechanism for online ratings; the second is an experimental investigation of the role of reputation on cooperation in dynamic network; the third is a methodological contribution to experimental research of game theory. The remaining of this section preludes and motivates each work.

1.1.1 Reputation Mechanisms

As the world immerses itself with the Internet, more human activities and interactions move online. A large portion of commerce nowadays either transact directly through or integrate closely with the Internet. Meanwhile, web and mobile applications constantly experiment with and ultimately redefine our social lives. Furthermore, new types of goods and services appear on the Internet at a blistering speed, creating fresh human experiences. Almost a countless number of interactions result from this ongoing technological revolution. To maximise the economic and societal gains, we utilise various mechanisms to promote positive interactions and prevent damaging ones. Reputation stands at the forefront of such mechanisms, and reputation mecha-

nisms become ubiquitous on the Internet. Reputation mechanisms help reveal hidden attributes of entities (e.g. goods, services, partners, etc) *prior to* engagement based on the feedback from past users. We identify two crucial factors that contribute to the effectiveness of most reputation mechanisms.

First, reporting feedback is costly in effort and many simply do not report unless there is extreme satisfaction or dissatisfaction. If all reports are of extreme opinions, the mechanism may produce inaccurate or biased signals to prospective users. To make it worse, not every report is truthful. In fact, in some cases, the reporter may have outside interest to deliberately skew one's reputation. For example, a buyer may offer an undeserved positive report in exchange for a monetary return promised by the seller, or conversely, the seller may give an unfair negative report as the retaliation to an unhappy buyer. Lying by selfish agents could render a reputation mechanism useless. Economics contributes a potential solution to both problems, namely incentivization via payments. By offering sufficient reward to honest feedback, the mechanism may compensate the cost of reporting as well as neutralise the profit of lying. For instance, various proper scoring rules [Savage, 1971] are such payment schemes. In the game theoretic context, this usually means that adequate payment leads to individual honest reporting that constitutes a Nash equilibrium of the game induced by the mechanism.

Second, even if every agent reports honestly, the mechanism may still not accurately inform a prospective buyer. In the majority cases of reputation mechanisms, reporting feedback is largely a subjective act. Buyers A and B may hold different opinions about a seller, but that does not necessarily mean that one of them is dishonest. It may simply well be that A and B have different preferences or tastes which would translate into different evaluations of the same product or service. For example, restaurant C and restaurant S both sells Chinese food, but S specialises in Sichuanese

cuisine[1] while C specialises in Cantonese cuisine[2]. Even to Chinese customers, these two restaurants offer distinct flavours and styles of food. Depending on one's culinary experience, different customers are likely to appreciate one cuisine more than the other and hence one restaurant more than the other. However, different opinions (reflected in reviews or rating) in this case does not warrant an intrinsic difference in the quality of food or cooking. To perspective customers whom we assume to have no prior experience with either cuisine or are simply indifferent between the two, how does the reputation mechanism derive a useful signal of the attributes of each restaurant. We believe that the key lies in personalisation. More specifically, given historic reports, we can adopt collaborative filtering techniques to extract user preferences to customise the signals generated by the mechanism.

In Chapter 3, we modify and combine game theoretic and collaborative filtering algorithms to construct a reputation mechanism that not only elicits truthful reports but also customise displayed reputation to individual users.

1.1.2 Reputation and Cooperation

Cooperation is fundamental for societies to thrive. Yet, natural selection by competition seems to favour defectors over cooperators. Consider the classic prisoner's dilemma, where defection is the dominant strategy over cooperation. In other words, assuming that individuals pursue rational self-interest, mutual defection is the stable outcome of the game. Evolutionary game theory predicts that, in a well-mixed population in which each agent is equally likely to interact with every other agent, defectors unequivocally outlive cooperators as they gain a higher fitness on average in each prisoner's dilemma game. In reality, however, cooperative behaviours are widespread at all levels, from basic bacteria to intricate geopolitics. In all cases, there exists some mechanism(s) supporting the evolution of cooperation.

[1]It is famous for its bold and spicy flavours.
[2]It is renown for its balanced flavours and cooking of delicacies.

[Nowak, 2006] summarises five possible mechanisms for evolving cooperation. *Direct reciprocity* arises when two individuals engage in repeated interaction and adopt strategies that are conditional on each other's past actions [Trivers, 1971, Fudenberg and Maskin, 1986]. It constraints current temptation of defection by the prospect of future interaction, i.e. if the tomorrow matters enough, then it's better off to behave today. A known strategy that epitomises direct reciprocity is the so-called "Tit-for-Tat", which starts with cooperation and simply copies the previous action of the opponent in later encounters. *Indirect reciprocity* functions when repeated interactions happen across the population and these interactions are observable by third-parties. Individuals decide what to do with the current opponent by observing what she has done to others in the past [Alexander, 1987, Nowak and Sigmund, 1998]. A notion of reputation naturally emerges in this setting. If reputation information is easily accessible, cooperation is likely to evolve. *Network reciprocity* relies on population structures that determine who interact with whom throughout the evolution [Nowak and May, 1992]. In this setting, individuals do not randomly interact with each other, but rather interact with fixed partners/neighbours. If enough cooperators cluster themselves together, they can accrue fitness and can successfully defend against lone defectors. *Multilevel selection* works by grouping similar individuals together to form groups, and there is competition between cooperators and defectors within groups as well as between groups [Wilson, 1975]. Ultimately, groups of cooperators outperform those of defectors and produce more cooperative offspring. [Traulsen and Nowak, 2006] suggests that multilevel selection may work if there exist many small groups and individuals do not move between groups often. *Kin selection* indicates that cooperation is favoured if individuals involved are genetically related. This is first mathematically formulated by the Hamilton's rule which states that a costly altruistic act can only happen if the coefficient of genetic relatedness exceeds the cost-to-benefit ratio of the altruistic act. An closely related and arguably more generalised concept is inclusive

fitness theory [Hamilton, 1964a, Hamilton, 1964b], which advocates that altruistic social behaviour promote the overall genetic success of an organism.

In human interaction, it is likely that more than one of the mechanisms above concurrently sustain cooperation. In Chapter 4, broadly speaking, we study the effects of indirect reciprocity and the dynamic version of network reciprocity on cooperation. More specifically, we set up a repeated multiplayer prisoner's dilemma game in which players are free to choose their interaction partners, and we experimentally investigate the relative importance of reputational and social knowledge in promoting cooperation in the resulting network.

1.1.3 Experimentation

Experiments are a form of play in a sense that we learn many basics as well as valuable knowledge by engaging playfully with the physical world. Each experiment purposely explores a small piece of the world in isolation. In science, the growth of knowledge is driven by two complementary forces: theory and empirical evidence. Theory organises existing knowledge and make predictions by deduction. Predictions aim to provide guidance in real life, however, they are often less than reliable depending how closely the theory models the reality. On the other hand, empirical evidence through experimentation or observation offers the ultimate test of any theory, and often it gives surprises. When this happens, we go back and adjust our theory and then we gather more empirical evidence to evaluate our revised theory. Through this feedback loop, our theory becomes more sophisticated, and more importantly, gains more explanatory power for observations over time. Ultimately, it evolves into a validated knowledge rather than staying merely as a theory.

Experimental research in economics did not start until the early 1970s. Before that, economists focused on developing theories of which many assumptions remained debatable to date. During the 1960s, the development of game theory, social choice,

voting theory, industrial organization, etc compels economists to turn to experimentation, as different theories offer competing explanations of microeconomic events and in some case a single theory anticipates multiple equilibria. In the 1970s, pioneering researchers carried out experiments for topics ranging from risky choice to market equilibrium, and laboratories were established for continuing research. Experimental economics then saw rapid growth throughout the 1980s and 1990s as more experimentalists emerged and more experimental research were published. In 1998, the journal *Experimental Economics* was created, signalling that experimental research became a mainstream area of study. In 2002, Daniel Kahneman[3] and Vernon Smith[4] shared the Nobel Prize in economic sciences, giving the ultimate recognition of the importance of experimental research to the discipline.

The enormous advance of computing technologies combined with the dramatic decrease of cost of computing saw surging popularity of computerised experiments since the mid-1990s. Universities around the world started to install dedicated computer laboratories for experimental research in economics. Obsessing with details, economists recommend detailed specifications to set up a economics lab [Friedman, 1994]. The actual implementation may vary across the facilities, however, researchers may agree that computerised experiments offers three primary advantages: they offer simpler "in-game" logistics and better collection of data; they give a greater control of procedure and information flow to minimise potential confounding factors; they are largely standardised so it is easier to replicate the results. Since the late 1990s, the Internet has been undergoing tremendous growth, in the meantime researchers of different disciplines have carried out various research by utilizing the web [Anderson et al., 2002, Ntoulas et al., 2004, Von Ahn, 2006, Cooper et al., 2010, Khatib et al., 2011]. In contrast, there has been relatively fewer cases of economic research

[3] "For having integrated insights from psychological research into economic science, especially concerning human judgment and decision-making under uncertainty"

[4] "For having established laboratory experiments as a tool in empirical economic analysis, especially in the study of alternative market mechanisms"

conducted online. Online experiments by economists is a recent phenomena, and there have been only sporadic efforts [Horton et al., 2011]. In particular, online economic experiments involving real-time interaction between multiple subjects have only emerged lately [Suri and Watts, 2011,Rand et al., 2011,Wang et al., 2012,Shirado et al., 2013]. Recognising the promising potentials of online experiments, we present in Chapter 5 a novel experimentation platform for conducting real-time interactive experiments on the Internet.

1.2 Thesis Organisation

This thesis is divided into 6 chapters. This first chapter provides contexts and motivates for the work. Chapter 2 provides an overview of a collection of relevant concepts and topics. Chapter 3 proposes a truthful reputation mechanism for personalised online ratings and examines its properties. Chapter 4 investigates the role of reputation and its relative importance with social knowledge on cooperation in dynamic networks. Chapter 5 presents the platform for conducting game theoretic experiments on the Internet. Finally, chapter 6 concludes this work and offers directions and suggestions for future research.

Chapter 2

Background

We introduce the related topics and concepts in this thesis. Specifically, we consider the economic approach of modelling notions of reputation, review some key concepts in game theory and mechanism design, survey the field of reputation mechanisms, and discuss experimental research in computer science and economics.

2.1 An Economic View of Reputation

Reputation may be deemed as an evaluation of some characteristic or quality of someone or something[1] that we loosely call as an "entity". It may derive from first-hand information based on direct interaction with the entity in question, or from second-hand information based on others' interactions with the entity. The utility of reputation usually is to inform decision makers about the prospective gain or loss of interacting with the reputation holder, thus the latter kind of information arguably forms the more common basis of reputation. However, it should be noted that both types of information may together play a part in forming one's reputation in the eye of beholder if available.

Economics uses the term "reputation" in different contexts. Game theory, in

[1]The "someone or something" here may extend to organisations or institutions

particular, provides a nature framework to analyse reputation as it especially concerns the interaction between parties, their interacting behaviour, and the resulting welfare (see [Mailath and Samuelson, 2006] for a review). In this section we outline two possible ways of modelling reputation using the game theoretic approach.

2.1.1 Bayesian Beliefs

A notion of reputation can arise from Bayesian updating of beliefs, meaning that agents develop the belief about the *type* of an entity. For our purpose, let us focus on the case of buyers and sellers in a market. Consider a seller offers a goods (or service), and the quality of the seller can be of either type "high (H)" or type "low (L)". We assume that the goods works as expected with probability p_H if the seller is of type H and that the goods works with probability p_L if the seller is of type L, where $0 < p_L < p_H < 1$. On the other hand, buyers are assumed to be risk neutral, and for simplicity, they value the goods at 1 if it works as expected or 0 otherwise. Buyers hold a *prior* probabilistic belief μ_0 that the seller is of type H, and update this prior based on the observed history of transactions of the seller. For instance, we assume that the seller's history consists of G good transactions and B bad transactions[2], then the buyers will form a posterior belief that the seller is of type H according to

$$\mu = \frac{\mu_0 p_H^G (1-p_H)^B}{\mu_0 p_H^G (1-p_H)^B + (1-\mu_0) p_L^G (1-p_L)^B}$$

This updated posterior μ gives the Bayesian notion of reputation of the seller, and it is increasing in G and decreasing in B. Reputation here acts a signalling device, allowing one to learn from others' past behaviour. Notice that in this simple example the type or quality of the seller and its corresponding p_t (where t is the seller's type, here as H or L) is exogenously decided. A more sophisticated model may involve a

[2]That is, the goods fails to meet the expectation.

p_t whose value is a function of the seller's effort.

2.1.2 Repeated Interactions

Alternatively, a notion of reputation may arise from repeated interactions, meaning that the agents develop an expectation of future behaviour of an entity. Consider the classic prisoner's dilemma game between two rational players. In the single-shot version of the game, it is well known that the dominant strategy for both players is to defect and the unique Nash equilibrium is mutual defection. When we extend the model into an infinitely repeated game, different equilibria emerge other than the one that is simply the infinite repetition of the single-shot equilibrium. For example, consider an equilibrium in which both players start with cooperating but both would change to defection once either player defects. Suppose that the mutual cooperation yields a payoff of $r > 0$, mutual defection results $p = 0$ and unilateral defection earns $t > r$ for the defector. Given the infinite horizon, a player's discounted payoff for cooperating is

$$r \cdot \sum_{k=0}^{\infty} \delta^k = \frac{r}{1-\delta}$$

, where $0 \leq \delta < 1$ is the discount factor. By defecting in a round, the player gets t in that round but 0 afterwards, and it follows that if

$$\frac{r}{1-\delta} \geq t \Rightarrow \delta \geq \frac{t-r}{t}$$

then continuous cooperation may be sustained as long as no one defects. In other words, if the future matters a great deal, it is better to maintain cooperativeness than destroying the mutual benefit in exchange for some near-term gain. Reputation in this context consists of a history of cooperative actions and acts as a sanctioning device to punish any off-the-equilibrium behaviour. This notion of reputation is closely related to a central concept in repeated games, namely the folk theorem,

which asserts that if the players are patient enough (i.e. the discounting factor δ is large enough) then any feasible and individually rational outcome can arise as a Nash equilibrium. Common extensions to our basic model include adding noise to the observation of past behaviour and dealing with exit and entry problem.

2.2 Game Theory and Mechanism Design

Game theory formalises strategic interaction between self-interested decision makers (agents hereafter). As a closely related topic, mechanism design concerns specifying rules for a scenario of interaction such that some socially optimal goals are achieved. This section gives a quick review of some key concepts in game theory and mechanism design theory.

2.2.1 Basic Definitions

A simple game G is defined by the following.

- A set of n agents, N
- A set of strategies for each agent $i \in N$, S_i
- A payoff/utility function $u_i : S_1 \times ... \times S_n \mapsto \mathbb{R}$ for each agent i

A *strategy* $s_i \in S_i$ of agent i specifies a plan of action(s) for i in the game. A strategy can be either pure or mixed. A *pure* strategy specifies action(s) deterministically, whereas a *mixed* strategy $\sigma_i \in \Delta(S_i)$ specifies a probability distribution over the agent's set of pure strategies S_i.

A *strategy profile* $s = (s_1, ..., s_n)$ is a vector of strategies each of which is a strategy by an agent. For notational convenience, it can also be denoted as $s = (s_i, s_{-i})$ where s_i is the strategy of agent i and s_{-i} are the strategies of the rest of the agents. By

definition, a agent's payoff depends on not only her behaviour but also that of all the other agents.

2.2.2 Solution Concepts

Game theory provides predictions of agent behaviour and game outcome through identifying stable points in the space of strategy profiles, which are the equilibria of the game. A fundamental solution concept is Nash equilibrium in which every agent maximises her own utility given the knowledge that other agents are doing the same.

Definition 2.1. (Nash Equilibrium) A strategy profile $s^* = (s_1^*, ..., s_n^*)$ is a Nash Equilibrium of G if for every agent i, and every $s_i^{'} \neq s_i^*$,

$$u_i(s_i^*, s_{-i}) \geq u_i(s_i^{'}, s_{-i})$$

In other words, no agent i has the incentive to deviate from her strategy s_i^* provided that all the other agents do not deviate from theirs. In the case of mixed strategies, we simply adopt the *expected* utility $u_i(\sigma^*) = \sum_{s \in S} u_i(s) \prod_{j=1}^{n} \sigma_j(s_j)$ where S is the space of strategy profiles. [Nash, 1950] shows that there exists at least one Nash equilibrium in every finite game. However, many consider that Nash equilibrium is "weak" in practice such that it requires: 1. Every agent has perfect information about others and knows that they are rational, and this is common knowledge; 2. Every agent plays the same equilibrium even if there exist multiple equilibria in the game.

A more robust solution concept is the *dominant strategy* equilibrium. A dominant strategy s_i for agent i is one such that $u_i(s_i, s_{-i}) \geq u_i(s_i^{'}, s_{-i})$ for every s_{-i} and $s_i^{'}$, that is, s_i is the best strategy for agent i regardless which strategies other agents may choose.

Definition 2.2. (Dominant Strategy Equilibrium) A strategy profile $s^* = (s_1^*, ..., s_n^*)$ is a Dominant Strategy equilibrium of G if s_i is a dominant strategy for every agent

i.

The dominant strategy equilibrium is a stronger solution concept than the Nash equilibrium because it makes no assumption about the information known to the agents and their belief about each other's rationality.

However, not many game has a dominant strategy equilibrium. To overcome the strong assumptions associated with the Nash equilibrium, *Bayesian* games assume that agents form beliefs about the preferences of other agents. Preferences are modelled as *types* such that $\theta_i \in \Theta_i$ is a realisation of agent i's type where Θ_i is the set of all possible types for i. Furthermore, Bayesian games assume that there exists a joint probability distribution of agents' types, $p = (\theta_1, ..., \theta_n)$, which is common knowledge among the agents.

Thus $s_i(\theta_i)$ in a Bayesian game specifies i's action(s) a given her type θ_i. Given a specific type θ_i and a strategy profile (s_i, s_{-i}), i's payoff is the expected payoff over all possible types of other agents, that is,

$$u_i(\theta_i, (s_i, s_{-i})) = \mathbb{E}_{\theta_{-i}}[u_i(\theta_i, (s_i(\theta_i), s_{-i}(\theta_{-i}))] = \sum_{\theta_{-i}} p(\theta_i, \theta_{-i}) u_i(s_i(\theta_i), s_{-i}(\theta_{-i}))$$

where $p(\theta_i, \theta_{-i})$ denotes the conditional beliefs about other agents' types given i's type θ_i.

Definition 2.3. (Bayesian Nash Equilibrium) A strategy profile $s^* = (s_i^*, ..., s_n^*)$ is a Bayesian Nash equilibrium of G if for every agent i, every $\theta_i \in \Theta_i$ and every $s_i^{'} \neq s_i^*$,

$$u_i(\theta_i, (s_i^*, s_{-i}^*)) \geq u_i(\theta_i, (s_i^{'}, s_{-i}^*))$$

In other words, every agent maximises her expected payoff for every possible type of hers by playing a strategy as the best response to the distribution over the types of all other agents.

2.2.3 Mechanism Design

A mechanism M is defined by the following:

- A set of n agents, N
- A set of types for each agent i, Θ_i
- A set of strategies for each agent i, S_i
- A social choice function $f : \Theta_1 \times ... \times \Theta_n \mapsto O$, which determines an outcome $f(\theta) \in O$ for every possible type profile $\theta = (\theta_1, ..., \theta_n)$
- An outcome choice function $g : S_1 \times ... \times S_n \mapsto O$, which determines an outcome $g(s) \in O$ for every possible strategy profile $s = (s_1, ..., s_n)$

The mechanism *implements* a social function f if there exists an equilibrium strategy profile $s^* = (s_i^*, ..., s_n^*)$ in the game induced by the mechanism such that $g(s^*(\theta)) = f(\theta)$ for all possible type profile $\theta \in \Theta_1 \times ... \times \Theta_n$. That is, the outcome choice function outputs the same outcome using some equilibrium strategy profile as the one computed by the social choice function for all possible type profiles. It is desirable that the equilibrium solution concept associated with the social choice function is as robust as possible.

A subclass of mechanism is called *direct-revelation* mechanisms in which agents only need to reveal their types to determine an outcome, i.e. $S_I = \theta_i$ and $f(\theta) = g\theta$ for all possible θ. In addition, a even smaller subclass of direct-revelation mechanism are the ones in which there exists an equilibrium $s^* = (s_i^*, ..., s_n^*)$ such that $s_i^*(\theta_i) = \theta_i$ for all possible θ_i of every agent i. In these mechanisms, it is optimal for all the agents to truthfully report their respective types. We call such mechanisms *incentive compatible* or *truthful*. Moreover, given incentive compatibility, we call a mechanism *individually rational* if its social choice function f ensures that $u_i(f(\theta_i, \theta)) \geq u_i(\theta_i)$

for all possible $\theta \in \Theta$ and every agent i, where $u_i(\theta_i)$ is i's payoff for not participating in the mechanism when her type is θ_i. Simply put, individual rationality requires the mechanism to provide adequate incentive for agents to participate.

The special class of incentive compatible direct-revelation mechanisms leads to a fundamental result in mechanism design theory, the *revelation principle*. Essentially, the principle states that any mechanism can be converted into an equivalent incentive compatible direct-revelation mechanism that implements the same social choice function.

Theorem 2.4. *Suppose there exists some mechanism M that implements a social choice function f in some equilibrium, then there must exist an incentive compatible direct-revelation mechanism that implements the same f in the same equilibrium.*

In other words, a social choice function f is implementable in some chosen equilibrium if and only if it is incentive compatible in the chosen equilibrium. The revelation principle facilitates mechanism designer by restricting the attention to incentive compatible direct-revelation mechanisms. If an outcome is not achievable by a truthful direct-revelation mechanism then it is not achievable at all, conversely, if a mechanism solves some outcome optimisation problem then there must exist a truthful direct-revelation mechanism for the same problem.

2.3 Reputation Mechanisms

Reputation mechanisms are information systems that inform users about the qualities or characteristics of target entities (e.g. people, goods, services) in a given context. Reputation mechanisms operate by gathering user feedback, aggregating feedback into reputational values and disseminating these values to users. As online transactions continue strong growth and offline transactions become increasingly reliant on online information, reputation mechanisms on the Internet flourish in recent years. We

discuss below a graph model of reputation mechanisms and the common threats to reputation mechanisms.

2.3.1 A Graph Model

Suppose a directed Graph $G = (V, E, t)$ where each node $i \in V$ corresponds to an agent/entity and each directed edge $(i, j) \in E$ describes the relationship that i makes a reputational report about j and its weight $t(i, j) = t_i(j)$ indicates the i's opinion towards j. We denote all agents' reports as $R = \{(U_1, t_1), ..., (U_n, t_n)\}$ where $n = |V|$, U_i denotes the set of target agents about whom i makes a reputational report, and $t_i(\cdot) = t(i, \cdot)$ assigns a numerical opinion to each of i's targets.

A reputation mechanism M is a function $f(G, v) = (r_1, ..., r_n)$ that outputs a vector of reputational values of v in the eyes of all the nodes of G such that $f_i(G, v) = r_i$ is v's reputation from i's perspective[3]. Therefore, for each agent i, there exists a subjective ranking of reported agents such that $f_i(G, u) < f_i(G, v) \Rightarrow u \prec_i v$ means that v is perceived to have higher reputation than u by i. This naturally lends to an *asymmetric* notion of reputation in the sense that each agent subjectively evaluates others' reputation. Alternatively, a *symmetric* notion of reputation generates a single global reputation value for every agent, i.e. $f_i(G, v) = f_j(G, v), \forall i, j$.

2.3.2 Common Manipulations

Based on the formulation above, we identify three common manipulations applicable to reputation mechanisms.

First, a malicious agent can create a number of fake agents who supply fake reports to inflate or damage a target's reputation. This is called *sybil attack*. More formally, given a $G = (V, E, t)$, a sybil manipulation targeted at an agent $s \in V$ is a tuple

[3] $f_i(G, i)$, one's assessment of its own reputation, may be defined to different values depending on specific application.

$\sigma = (S, E_S, t_S)$ where S is a set of sybils, $E_S = \{(x, y) : x \in S \cup s, y \in S \cup V\}$ is a set of fraudulent edges, and $t_S : E_S \mapsto \mathbb{R}^+$ gives the weights on these edges. Applying σ to G results in $G^{'} = (V \cup S, E \cup E_S, t^{'})$ where $t^{'}(e) = t(e)$ for $e \in E$ and $t^{'}(e^{'}) = t_S(e^{'})$ for $e^{'} \in E_S$.

Second, an agent may report dishonest feedback about any other agent. Formally, given a $G = (V, E, t)$, a case of misreport(s) by an agent m is a tuple $\sigma = (S, E_m, t_m)$ where $E_m = \{(m, v) : v \in V\}$ and $t_m : E_m \mapsto \mathbb{R}^+$. In addition, we define $E_{-m} = \{(u, v) : (u, v) \in E, u \neq m\}$ as the set of existing edges in G that do not originate from m. Applying σ to G results in $G^{'} = (V, E_{-m} \cup E_m, t^{'})$ where $t^{'}(e) = t(e)$ for $e \in E_{-m}$ and $t^{'}(e^{'}) = t_m(e^{'})$ for $e^{'} \in E_m$.

Third, a disreputable agent may remove itself and create a new identity (node) to start from scratch, which usually happens when the agent has maximised the return from its cheating actions. This is called *whitewashing* and it is frequent problem on the Internet since online identities are cheap to create . Formally, given a $G = (V, E, t)$, an act of whitewashing by w is a tuple $\sigma = (\{w, w^{'}\}, E_w)$ where $w^{'}$ is the new identity and $E_w = \{(w, v) : (w, v) \in E\} \cup \{(v, w) : (v, w) \in E\}$. Applying σ to G results in $G^{'} = (V \setminus \{w\} \cup \{w^{'}\}, E \setminus E_w, t)$.

It is noted that in some cases a sophisticated attacker may apply a composite of manipulations listed above. For a detailed survey of possible threats to online reputation mechanisms, see [Hoffman et al., 2009].

2.3.3 Selected Mechanisms

A prominent reputation mechanism that utilises the underlying graph structure of entities is *PageRank* [Page et al., 1999]. It is used to calculate the relative importance of web pages based on their hyperlink structure. In a sense, we can interpret that a highly important page is highly reputable for containing relevant information with regard to a given search keyword. Intuitively, it is based on the idea that a page

is likely to be highly reputable if other highly reputable pages point to it. This is implemented using the "random surfer" model: imagine a web surfer randomly clicks on a link on any page he is on, the fraction of the time (or the probability) the surfer visits a page i during his random surf indicates the importance or "reputation" of i. Formally, the importance of i, $r(i)$, is defined as,

$$r(i) = \frac{\epsilon}{n} + (1-\epsilon) \sum_{j:(j,i)\in E} \frac{r(j)}{|out(j)|}$$

where n is the number of pages in the network, $|out(j)|$ is the number of links that originate from j, and ϵ is the probability that the surfer randomly jumps to some other page (i.e. behaviourally, the surfer may get bored with the current page, type in a new URL and go to a page somewhere else on the web; thus with probability $1-\epsilon$, the random surfer clicks on a link within current page). It is obvious that the weight on each link (j,i) is determined by the number of links that originate from j, i.e. $\frac{1}{|out(j)|}$. To compute these reputation values, one can utilise power iteration as below.

$$r_0(i) = \frac{1}{n} \qquad \text{(Initialisation)}$$

$$r_{v+1}(i) = \frac{\epsilon}{n} + (1-\epsilon) \sum_{j:(j,i)\in E} \frac{r_v(j)}{|out(j)|}$$

That is, we use the value of $r(i)$ at the vth iteration to compute its value at the $(v+1)$th iteration.

There are other graph theoretic reputation mechanisms, see [Kamvar et al., 2003, Cheng and Friedman, 2005, Hopcroft and Sheldon, 2007, Altman and Tennenholtz, 2010] for a selection of them.

In the domain of P2P applications, reputation mechanisms using probabilistic inference are popular in the literature. As one of the first examples of such mechanism, the *Beta reputation* system [Ismail and Jøsang, 2002] estimates seller's reputation us-

ing a model based on beta probability distribution of binary events, whose probability density function is defined as

$$beta(x|\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$$

where x is the random variable, B is the beta function, and $\alpha, \beta > 0$ are parameters that characterise the distribution. Ratings are binary with 1 denoting positive feedback and 0 denoting negative feedback. For each seller, the mechanism tallies its positive and negative feedback as m and n respectively, and use them to adjust the seller's beta distribution such that $\alpha = m + 1$ and $\beta = n + 1$[4]. The mechanism estimates the reputation of a seller s using the expected value of x, that is

$$r(s) = \mathbb{E}(x) = \frac{\alpha}{\alpha+\beta}$$

. [Whitby et al., 2005] extends the mechanism to deal with unfair ratings by filtering out those in the minority.

Other examples of P2P reputation mechanisms include [Yu and Singh, 2003, Le Boudec et al., 2004, Walsh and Sirer, 2005, Teacy et al., 2006], among others.

Furthermore, [Marti and Garcia-Molina, 2006, Jøsang et al., 2007] provide extensive review and analysis of online reputation mechanisms that are proposed in research or implemented in practice.

2.4 Experimental Research

Experimental research is an indispensable tool across all disciplines of science. In this section we motivate experimental research in computer science and economics respectively, and give an overview of common methodological issues.

[4] α and β are initialised to 1.

2.4.1 Experimental Computer Science

Research in computer science generally focuses on two areas: mathematical formalism and innovative engineering. The former abstracts away mechanical details and study computing models mathematically. Theoretical research to this end indeed produces stunning insights and novel paradigms which in turn give rise to a new wave of engineering endeavours. The latter focuses on building something new and useful. Research efforts in this front generates a constant stream of new techniques, algorithms and tools that benefit numerous applications. Many argue that the core of computer science always emphasises theoretical research, but a fundamental goal of the discipline remains practical implementation.

Unlike natural or physical sciences, computer science rarely follows the analytic process that some understanding is developed through observing and describing existing phenomena. As a result, the notion of experimental research in computer science often deviates from that of physics/chemistry/biology in which hypotheses of empirically driven models are tested by experiments. By contrast, the two common types of experimentation in computer science are demonstration and evaluation.

A large chunk of computer science is about building applications. Such applications range from standalone algorithms to large-scale systems. Experimental research in this regard invests a significant amount of engineering efforts to demonstrate the feasibility and the functionalities of the built software and hardware (sometimes in the form of proof-of-concept). Examples are especially abundant in areas such as computer graphics, human-computer interaction, artificial intelligence and robotics, mobile computing, web science, etc.

The other mode of experimentation in computer science is empirical evaluation of theories or implementation. Such evaluation entails concrete measurement and benchmarking that are typical in physical sciences. It is common to find experimental results evaluating algorithms or software systems, in order to highlight their

performance advantages and novel functionalities. Such results are necessary because either the subject of investigation is yet too complex to be analysed theoretically or the value of the research is mostly justified by practical deployment. Researchers often adopt simulation and testing using real-world data to carry out this line of research.

Computer science is becoming increasingly interdisciplinary by acquiring/providing concepts and methods from/to other disciplines. In recent years, computer scientists have utilised theories and techniques from economists to design and analyse applications in decentralised computing environment. In these applications, economics offers useful frameworks for predicting and regulating interaction among artificial or human agents.

2.4.2 Experimental Economics

Economics is predominantly theory-intensive and a prevailing attitude is (or at least was) that a consistent theory is the most sought-after prize. Nonetheless, economics remains a study of practical affairs. [Smith, 1982] states that (lab) experiments create live economic systems that are at least as behaviourally rich as (and often more realistic than) those constructed in theoretical models. In recent years, experimental research plays an increasingly important role in the development of the discipline. Experiments not only validates, tests and improve existing theories but also produces necessary data to address problems for which a theory is not available. Indeed, the 2002 Nobel Prize in economic sciences recognises the contribution of experimental research by awarding the prize to Vernon Smith and Daniel Kahneman who are arguably the founders of experimental economics and behavioural economics respectively.

It is worth pointing out that, despite sharing the prize, Kahnman and Smith have different research backgrounds by training: Kahnman is a psychologist, whereas Smith is an economist. Although experimental economics is influenced by behavioural

research in psychology and vice versa, there are some important differences [Hertwig and Ortmann, 2001]. First, economics experiments always follow detailed protocol and instructions while psychology experiments may adopt considerably more casual arrangements. Second, economics experiments usually conduct repeated trials while many psychology experiments have a single-shot structure. Third, monetary incentives is a must in economics experiments and they usually include a performance-based bonus and are always realised as soon as the experiment ends, in contrast, it is unusual for psychologists to pay cash as compensation and a performance bonus is even rarer. Fourth, almost any form of deception is forbidden in experimental economics, but that is acceptable or even a standard practice to some extent in experimental psychology.

In economics, there are two main types of experiments: field experiments and lab experiments.

In field experiments, researchers examine the economic decision-making of randomised subjects in a naturally occurring setting. The common goal is to preserve a mixture of control and realism such that subjects' behaviours provide the most realistic data for the question in hand. In some cases, subjects may not even be aware that they are participating in an experiment. Field experiments are closely related to large-scale social experiments conducted by policymakers for a wide range of societal issues such as energy pricing, employment programs, welfare benefits, etc. Field experiments are normally categorised into three kinds by economists [Carpenter et al., 2005]: artefactual, framed and natural. Artefactual experiments are essentially lab experiments with the use of non-standard subjects or even those that are subjects of study. For example, we may recruit stock traders instead of students for experiments on financial trading. Framed experiments move closer to the real world by putting the subjects in the field context in terms of the environment, commodity, task and information set given to the subjects. The natural experiments are the same as the

framed experiments except that subjects do not know they are part of an experiment and naturally undertake certain tasks of interest. Some argue that an advantage of field experiments in general is their *external validity*, that is, they are to different extents descriptive of the real-world conditions.

Lab experiments make subjects play as decision makers in artificial roles under a controlled environment. Lab experiments offers a less expensive and more convenient alternative when the equivalent field experiments prove to be cost prohibitive or sometimes even unethical. More importantly, lab experiments impose a set of conditions that abstract away potentially confounding factors. Such conditions may include institutions and rules, incentives, matching, subject selection and so on. This allows researchers to focus on examining the causal relationships between experimental variables (e.g. whether change in A leads to change in B), and also forces them to care more about reproducibility as a higher degree of control demands more robustness in results. The ability of imposing various controls over subjects arguably make the *internal validity* a strength of lab experiments[5], that is, experimenters can be more confident about the causal conclusions they draw based on collected data.

2.4.3 Basics of Lab Experimentation

In this section we give a concise discussion of the common methodological issues in the context of lab experiments that are most relevant to our research.

Before designing and implementing any experiment, it is useful to know the common types of experiments conducted in lab. In terms of research topics, the most frequently run lab experiments are: *market experiments*, *game theoretic experiments*, and *individual decision experiments*. Market experiments investigate the effects of different designs and rules of a trading institution. For example, trading in various financial markets is often a subject of investigation in lab. Game theoretic experiments

[5]For a thorough discussion of internal validity and external validity in economic experiments, see [Schram, 2005].

evaluate theoretical predictions of game models. In particular, in games with multiple equilibria, experiments directly verify which equilibrium is practically obtainable. Individual decision experiments examine behavioural basis of fundamental assumptions or theories in economics. For example, many experiments test expected utility theory by eliciting responses to lottery-based questions. In terms of general research goals, there are normally three possibilities: testing theory, informing theory and trialling policy. When there is a single theory involved in the experiment, researchers normally test its robustness to some deviation from its assumptions; When there are multiple theories present in the experiment, researchers are usually interested in the results of a competition between them, especially when they make distinct predictions in the same settings. In addition to theory testing, experiments can also suggest refinements and improvements to existing theories. Moreover, identifying unexpected regularities in experimental data may assist in creating new theories (e.g. from Allais paradox to bounded rationality). Finally, experiments can also advise policymakers as well as practitioners about the implications and the performance of some prospective policy. Most policy-oriented experiments concern market design scenarios such as selecting the right auctions for selling spectrum.

When designing experiments, researchers often need to consider the following choices. First, the experimenters needs to decide how to expose subjects to different treatments. *Between-subject* design exposes each subject to exactly one treatment, whereas *within-subject* design exposes each subject to more than one treatments. Within-subject design offers stronger statistical power but may lead to spurious effects and is susceptible to "demand effect" where subjects attempt to interpret the experimenter's intentions and change their behaviour accordingly. Between-subject design requires random assignment of subjects and may require a large sample size to achieve statistical power, which may prevent experiments that involve a large number of variants (combinations of treatment variables). [Charness et al., 2012] recommends

between-subject design in general to eliminate the substantial risk of confounds introduced by within-subject design. Second, the experimenter needs to choose between *single-shot* and *repeated* interaction depending on the research question. Single-shot interaction enhances incentive for decision, has no strategic spillovers over time, and is simple to perform as it takes relatively little time. Repeated interaction gives subjects more time to learn in complicated experiments and generates more observations that can reveal interesting dynamics of the experimental process. For a design with repeated interaction, the experimenter needs to decide the matching protocol. *Partners* matching makes subjects play with the same partners throughout the experiment; *Strangers* matching randomly rematches subjects every time; *Perfect strangers* matching ensures that no pair of subjects is matched up more than once. [Andreoni and Croson, 2008] suggests that a Strangers condition will be the most appropriate if the experiment aims to test predictions based on single-shot equilibria. Third, the experimenter needs to settle on a payment protocol. Subjects may be paid by aggregating the payoffs of all their decisions. This assume that there is no significant discounting of payoffs among subjects. However, there may exist "wealth effects" such that subjects can behave differently in later stage of the experiment depending on their accumulated payoffs. Alternatively, subjects may be paid based on a lottery that randomly picks n decisions they make during the experiment. This may incentivize subjects to treat every decision equally if n is sufficiently small.

In conducting experiments, it is important to control for different aspects of subject participation. First, as subjects may differ in a number unobservable and uncontrollable characteristics, it is necessary to randomise as much as possible to eliminate such variations given a sufficiently large sample size. For example, we can randomise invitation of subjects to sessions, assignment of treatments to sessions, and allocation of subjects to roles or groups within a session. Moreover, experiments normally take place under fully anonymised condition, meaning that every subject is assigned

a fictitious ID and real identities are never revealed. It is also a common practice and potentially helpful to include a questionnaire after each experiment session to collect demographic information and to elicit subjective attitudes (e.g. risk and fairness) which will allow the experimenter to control for these intrinsic variables when performing the analysis.

Chapter 3

A Truthful Reputation Mechanism for Personalised Online Ratings

Merchants increasingly rely on their presence on the Internet to promote their products. Besides traditional banner ads, merchants place particular importance on their reputation on various rating websites, where a merchant's goods or service receives user-generated ratings. This new form of marketing ought to create a win-win situation for both merchants and consumers, as ratings are freely crowdsourced for merchants while consumers enjoy the wisdom of crowds. Unfortunately, this is often not the case in practice. Both merchants and consumers cry foul at displayed reputation from time to time, and evidences of manipulation on such sites have caught attention in recent media reports [Segal, 2011, Streitfeld, 2012]. Additional intricacy in displayed reputation points to the fact that such ratings are usually biased, resulting in preference mismatches between users who submit the ratings and those who use the same set of ratings for evaluation. In this chapter, we propose a novel reputation mechanism for crowdsourced online ratings by incorporating both game-theoretic and data mining methods. Our mechanism not only enforces truthful reporting from raters, but also takes into account preference similarities among them to personalize

displayed reputation. Furthermore, our mechanism is shown to be computable in polynomial time, paving the way for empirical implementation and testing.

3.1 Online Ratings

As merchants use the Internet to promote products or services, maintaining a sound online image becomes crucial for them to bolster future sales. This is driven by the fact that consumers now enjoy an abundance of choices as well as relevant information, and their search cost has been vastly reduced by the Internet. In particular, a number of crowdsourcing rating websites (e.g. Yelp, TripAdvisor, Opez, etc) accrue, aggregate and publish user-generated ratings, and have emerged as a major decision factor for consumers before they interact with merchants. While the wisdom of crowds displaces traditional advertising, scoring a high displayed reputation on such websites has arguably become a primary focus of merchant's sales and marketing strategy.

Unfortunately, both consumers and merchants hold doubts about the reliability and the usability of such rating sites. In many cases, both sides claim that displayed reputation deviates far from underlying quality. We believe that the fundamental problem is that the ratings are noisy, and current reputation mechanisms largely fail to extract and differentiate informative ratings for the inquiring users. In particular, the following three factors mainly contribute to the noisiness in online ratings:

1. Genuine raters sometimes simply fail to report their honest opinions, especially when there exist information cascades or herding effect among them.

2. Rating manipulation directly undermines the credibility of aggregated reputation. Such manipulation mostly comes in the form of fake ratings that are either unearned positive reports ("ballot stuffing") or unsubstantiated negative reports ("bad mouthing").

3. Raters may have different personal preferences and most of them are not experts, hence their ratings are most likely *subjective and biased* assessments. For example, while some traveller downrates a hotel due to room size, others may rate it highly because of its convenient location.

The first problem may be dealt with by some appropriate game-theoretic design of incentives. The key idea is to design a payoff scheme under which agents are incentivized to report honestly given a chosen solution concept. It is harder to tackle the second challenge by implementing economic incentives alone, since the incentive structure of misbehaving users may be external and unknown to the system. The third problem concerns how to derive and make use of preferences comparison between the users who contribute to displayed reputation and those who evaluate that reputation. As it is recognized that ratings are biased [Hu et al., 2006], differences among ratings by honest raters are primarily due to different personal preferences. In order to tackle all three challenges, we explore the approach of blending a game-theoretic method with a data mining technique.

3.2 Related Work

Various reputation mechanisms have been proposed for areas such as P2P networks [Kamvar et al., 2003], Internet marketplaces [Jsang and Ismail, 2002], online communities [Adler and De Alfaro, 2007], etc. We choose to focus on designing a mechanism specifically for rating websites, for which we consider two primary design questions. The first is about how we elicit truthful opinions from raters, and the second is about how we personalize the reputation evaluation for users since opinions contain subjective judgement. We have seen different proposals [Jurca and Faltings, 2009, Lambert and Shoham, 2009, Miller et al., 2005] made to attack the first question, and we choose to utilize the BTS method [Prelec, 2004]. The method has also been applied in other

areas: [Howie et al., 2011] adopts the BTS to forecast product adoption; [Weiss, 2009] applies the method to elicit and combine expert opinions for policy analysis; [Carvalho and Larson, 2011] uses it to distribute a shared reward among agents. Particularly, an extensive empirical study [Shaw et al., 2011] compares the BTS with many other incentive schemes for eliciting effortful answers from online users, and has found that the BTS with monetary rewards produces the most accurate answers, despite that many other methods also offer financial incentives. This result suggests that the BTS may be more able to elicit quality feedback when some effort is involved, which corresponds to our situation where thoughtful ratings are much useful than careless or even arbitrary ones. Our latter design question is attacked by a SimRank-based algorithm. SimRank [Jeh and Widom, 2002] is widely studied within data mining community, where its derivatives and other link-based similarity algorithms have emerged in recent years, such as [Antonellis et al., 2008, Li et al., 2009, Xi et al., 2005], etc. More specifically, some have applied SimRank-based algorithms in collaborative filtering applications where personalization is a key feature. For examples, [Desrosiers and Karypis, 2010] utilizes SimRank to construct general recommendation systems, while [Chen et al., 2010] combines SimRank with clustering to match users in online dating networks.

3.3 The Settings

There are two sub-settings we consider, and together they form the backdrop of our proposed mechanism. The first setting depicts a "high-level picture" of relationships between raters and targets on a rating website, and these relationships can then be exploited using a variant of SimRank [Jeh and Widom, 2002] to estimate the level of similarity between any two raters. In the second setting we look into the strategic behavior of a rater when she rates a target, which is the individual act that forms

the relationships captured by the first setting, and we apply the Bayesian Truth Serum [Prelec, 2004] to measure the degrees of truthfulness in ratings. We try to keep the notations consistent across two settings.

3.3.1 A SimRank Variant

SimRank [Jeh and Widom, 2002] is an influential algorithm for computing similarity between objects using their link structure on a graph. The algorithm makes use of the object-to-object relationships, and is based on the intuition that "two objects are similar if they are related to similar objects". However, the original SimRank does *not* consider the weights but only the existence of edges. In our case, the graph structure is bipartite, modeling the dichotomy between raters and targets, connected by edges representing rating relationships. Particularly, our variant incorporates the edge weights when computing similarities, as they denote the specific ratings submitted by raters.

Formally, we use a weighted bipartite graph $G = (A, T, E, w)$ to illustrate the "ecosystem" of a rating website. Vertex set A and T represents the set of raters and the set of targets respectively, and E denotes the set of directed edges from A to T in which an edge $e \in E$ depicts the act of submitting a rating. For simplicity, we focus on only the numerical component of a rating here[1]. More specifically, for a target $t \in T$, an agent $a \in A$ chooses to submit a positive integer rating $x_a^t = w(at)$, for which the weight function $w : E \to \{1, \ldots, m\} = R$ assigns a weight to edge at, where the integer parameter $m \geq 2$ defines the size of rating set R. Figure 1 gives an example of this setup. Moreover, let $O(a) = \{t | at \in E, t \in T\}$ denote the set of *targets* that agent a has rated, and let $I(t) = \{a | at \in E, a \in A\}$ denote the set of *agents* who have rated target t. Now we are ready to lay out a modified version of *SimRank* algorithm for computing the preferences similarity as below.

[1]Numerical representation of reputation is arguably the strongest signal to consumers in practice.

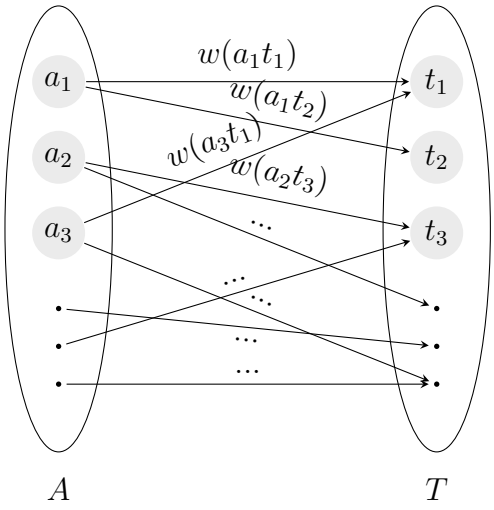


Figure 3.1: A minimal example of the graph model G of a rating website

We describe three types of measure of similarity: $S(x, y)$ (where $x, y \in R$) measures the *numerical* similarity between two *ratings*; $S(a, b)$ (where $a, b \in A$) measures the *preference* similarity between two *agents*; and $S(s, t)$ (where $s, t \in T$) measures the *quality* similarity between two *targets*. We define that, for any $a, b \in A$, $S(a, b) = 1$ if $a = b$ and $S(a, b) = 0$ if $O(a) = \emptyset$ or $O(b) = \emptyset$. Simply put, an agent is maximally similar to herself, and if an agent contribute no rating at all then she is minimally similar to any other agent. The same definitions apply to $S(s, t)$ for any $s, t \in T$, except that we replace $O(\cdot)$ with $I(\cdot)$.

We compute $S(x, y) \in [\frac{1}{m}, 1]$ of any two ratings x and y using

$$S(x, y) = 1 - \frac{|x - y|}{m} \tag{3.1}$$

which measures the degree of difference between two ratings, normalized by the range of ratings. We use the following set of equations to compute $S(a, b) \in [0, 1]$ for any pair of agents a and b such that $a \neq b$,

$$
\begin{aligned}
S_a(a,b) &= \frac{\lambda_1}{|O(a)|} \sum_{i=1}^{|O(a)|} \max_{j=1}^{|O(b)|} S\left(O_i(a), O_j(b)\right) \cdot S\left(x_a^{O_i(a)}, x_b^{O_j(b)}\right) \\
S_b(a,b) &= \frac{\lambda_1}{|O(b)|} \sum_{j=1}^{|O(b)|} \max_{i=1}^{|O(a)|} S\left(O_i(a), O_j(b)\right) \cdot S\left(x_a^{O_i(a)}, x_b^{O_j(b)}\right) \\
S(a,b) &= \min\left(S_a(a,b), S_b(a,b)\right) \qquad (3.2)
\end{aligned}
$$

where the intermediate term $S_a(a,b)$ can be interpreted as agent a's affinity for agent b's preferences, and is derived by averaging the similarities between how a has rated each of her targets (i.e. $O_i(a)$ where $1 \leq i \leq |O(a)|$) and how b has rated *the most similar* target among his $O(b)$ to each $O_i(a)$. The calculation for $S_b(a,b)$ follows the same logic. The reasoning behind the maximum approach here is that given the diversity of targets each agent may have rated on a rating site it is ineffective to compare two targets that are unrelated or very different. For example, many rating websites collect ratings for businesses in different categories (e.g. restaurants, theaters, bars, hairdressers, etc), thus it does not make sense to compare a target of one category with another of a different category. Moreover, even within the same category, there may well exist distinct types of offerings. For example, it makes little sense to compare a luxury resort with a budget hostel regardless the mostly subjective opinions they receive respectively[2]. The minimum choice in the end ensures that the level of preference similarity is mutual between two agents, that is, both agents must be interested in each other's targets. We can interpret the constant $\lambda_1 \in (0,1)$ (and λ_2 below) as a decay confidence factor that discounts the degree of similarity as comparisons progress.

Likewise, we compute $S(s,t) \in [0,1]$ for any two targets s and t such that $s \neq t$

[2]While both offering travel accommodation, they are by design very different products and cater for raters with very different preferences. These intrinsic features are separate from their reputation: a resort does not necessarily assure a high reputation nor does a hostel guarantee a low one

using

$$
\begin{aligned}
S_s(s,t) &= \frac{\lambda_2}{|I(s)|} \sum_{i=1}^{|I(s)|} \max_{j=1}^{|I(t)|} S\left(I_i(s), I_j(t)\right) \cdot S\left(x^s_{I_i(s)}, x^t_{I_j(t)}\right) \\
S_t(s,t) &= \frac{\lambda_2}{|I(t)|} \sum_{j=1}^{|I(t)|} \max_{i=1}^{|I(s)|} S\left(I_i(s), I_j(t)\right) \cdot S\left(x^s_{I_i(s)}, x^t_{I_j(t)}\right) \\
S(s,t) &= \min\left(S_s(s,t), S_t(s,t)\right) \qquad (3.3)
\end{aligned}
$$

As seen above, the similarity definitions are recursive and they are mutually reinforcing, such that similar agents rate similar targets similarly, while similar targets are rated similarly by similar agents. Further, it is not hard to see that the resulting similarity scores are symmetric, i.e. $S(a,b) = S(b,a)$ for any $a, b \in A$ and $S(s,t) = S(t,s)$ for any $s, t \in T$.

It is worth noting that when estimating similarity between any two nodes u and v, SimRank-based algorithms take into account nodes *beyond* u and v's common neighbors. In contrast, other similarity algorithms such as SCAN [Xu et al., 2007] considers only common neighbors of any two nodes. Given our context here, it is likely that two agents may not have rated many shared targets but have rated some similar ones. For example, agents u and v may have rated two different Italian restaurants of similar quality, and it would be wasteful to exclude such information from our estimation. We argue that it is sensible to adopt an SimRank-based approach here, as it offers more extensive assessments. We will discuss the computational cost of this estimation in Section 3.4.2.

3.3.2 Bayesian Truth Serum

Bayesian Truth Serum is a scoring method for eliciting honest answers to multiple-choice questions, and is particularly suited for questions where answers are more of

a matter of subjective opinions. The core feature of BTS is that it is not biased towards the consensus answer, and truthful reporting is always the best strategy for an agent even if she is certain that her answer is of a minority. The scoring criterion of BTS rewards answers that are "surprisingly common" and punishes those that are "surprisingly" otherwise, based on the comparison between the collectively predicted outcome and the actual outcome of answers. In addition, unlike [Miller et al., 2005], BTS does not require extensive knowledge of agents' prior beliefs, except that it assumes that they share a common prior that does not need to be known.

We are interested in obtaining *truthful reports from genuine raters*. By truthful and genuine, we mean rater reports her honest opinion about a target she has indeed interacted with. Given the subjective nature of ratings, we assume that an absolutely objective rating is unknowable for any given target. We emphasize that we apply BTS primarily to ratings from genuine raters though it is also useful for resisting fake ratings, as we will see later.

Formally, a set of agents $\{1, \ldots, n\} = N \subseteq A$, where $n \geq 3$, choose to rate a given target $t \in T$. Each $i \in N$ observes a private signal $o_i^t \in \{1, \ldots, m\}$ about t's quality after interacting with t, and o_i^t is the private opinion as well as the *truthful rating* i ought to give to t. Further, for each target t, let ω^t be an unknown parameter representing the population distribution of the truthful ratings for t. Based on their respective truthful rating, each i predicts how target t is rated by the population, and we denote this prediction by $d_i^t = (d_i^{t^1}, \ldots, d_i^{t^m})$, where $0 \leq d_i^{t^k} \leq 1$ and $\sum_{k=1}^{m} d_i^{t^k} = 1$. In other words, $d_i^t = \mathbb{E}[\omega^t | o_i^t]$.

Furthermore, the following assumptions are made for the genuine raters:

1. There exists a *common prior* distribution $p(\omega^t)$ over ω^t.

2. Every rater i is *self-interested* and aims to maximize the expected score from the BTS game.

3. Every rater i is *Bayesian rational* such that i performs Bayesian update on prior $p(\omega^t)$ with her private opinion o_i^t to form posterior $p(\omega^t|o_i^t)$.

4. The private signal received by a rater is *stochastic relevant*, i.e. $\forall i, j \in N, p(\omega^t|o_i^t) = p(\omega^t|o_j^t)$ if, and only if, $o_i^t = o_j^t$.

5. There exists a *sufficiently large population* of raters so that a target's empirical distribution of ratings cannot be distorted by a single agent's rating.

In effect, BTS induces a n-person reporting game among the raters for a target t in which each agent i chooses a strategy $s_i = (x_i^t, y_i^t)$, where $x_i^t \in \{1, \ldots, m\}$ is the reported rating and $y_i^t = (y_i^{t^1}, \ldots, y_i^{t^m})$ $(\sum_{k=1}^m y_i^{t^k} = 1)$ is the reported prediction of the percentage of agents choosing rating k. We define the notion of truthful reporting below.

Definition 3.1. (Truthful Reporting) An agent i's strategy is truthful if and only if i's reports are equal to her truthful opinions regarding a given target t, i.e. $x_i^t = o_i^t$ and $y_i^t = d_i^t$

Moreover, S_i is the set of strategies available for agent i, and is determined by the set of available ratings R in Section 3.3.1. We represent $s = (s_1, \ldots, s_n) \in S = S_1 \times \ldots \times S_n$ a strategy profile, and $s_{-i} = (s_1 \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$ a profile of strategy excluding i's. We then define our chosen solution concept, Bayesian Nash equilibrium, given the payoff function $\Pi : S \to \mathbb{R}^n$.

Definition 3.2. (Bayesian Nash Equilibrium) A strategy profile $s = (s_1, \ldots, s_n)$ is a Bayesian Nash equilibrium in a reporting game for a target t, if for each agent i and for every strategy $s_i^{'} \neq s_i$ in S_i, $\mathbb{E}[\Pi_i(s_i, s_{-i})|o_i^t, d_i^t] \geq \mathbb{E}[\Pi_i(s_i^{'}, s_{-i})|o_i^t, d_i^t]$, where $\Pi_i : S \to \mathbb{R}$ is the payoff received by agent i.

In the BTS setting, Π_i is the scoring function that assigns a score to rating x_i^t and

prediction y_i^t, and is defined as

$$\Pi_i(s = (s_1, \ldots, s_n)) = \sum_{k=1}^{m} b(x_i^t, k) \log \frac{\bar{x}_k}{\bar{y}_k} + \alpha \sum_{k=1}^{m} \bar{x}_k \log \frac{y_i^{t^k}}{\bar{x}_k} \tag{3.4}$$

where $0 < \alpha \leq 1$, and

$$b(x_i^t, k) = \begin{cases} 1 & \text{if } x_i^t = k \\ 0 & \text{otherwise} \end{cases}$$

$$\bar{x}_k = \frac{1}{n} \sum_{i \in N} b(x_i^t, k)$$

$$\log \bar{y}_k = \frac{1}{n} \sum_{i \in N} \log y_i^{t^k}$$

The first summand of the BTS payoff is called the *information score*, and it evaluates i's reported rating x_i^t using the log-ratio between its empirical frequency endorsed by the population and its geometric mean frequency as collectively predicted by the population. The second summand is called the *prediction score*, which penalizes any difference between i's reported predicted distribution and the empirical distribution, using their relative entropy. The expected prediction score is maximized by reporting the expected frequencies, i.e. $y_i^t = d_i^t = \mathbb{E}[\omega^t | o_i^t]$. The best possible prediction score is zero, and is obtained when $y_i^{t^k} = \bar{x}_k$ for every $k \in \{1, \ldots, m\}$. The constant α is thus the weight assigned to prediction penalty.

Given the assumptions and the payoff function as above, [Prelec, 2004] shows that the following theorem holds:

Theorem 3.3. *Truthful reporting by every agent is the pareto-optimal Bayesian Nash equilibrium of the induced game of the BTS.*

Theorem 3.3 means that, provided that all other agents report truthfully, in expectation, each agent strictly maximizes her own score by reporting truthfully as

well, i.e. $x_i^t = o_i^t$ and $y_i^t = d_i^t$. Moreover, agents' payoffs in the truth-telling equilibrium cannot be improved without reducing at least one agent's payoff. It should be noted that there may exist differences in scores across the agents even if they all report truthfully, and such differences reflect the relative accuracies in their reports based on the "surprisingly common" criterion mentioned before. In other words, in the truth-telling equilibrium, a lower score does not indicate dishonest reporting, but that the corresponding opinions are relatively less accurate compared to the collective opinions of the crowd. We will take into account this element of implied relative accuracy when we derive the aggregated reputation later.

3.4 The Mechanism

We extend one of the BTS assumptions to ours: raters are self-interested and aim to maximize their expected payoffs from the mechanism. The payoff structure of the mechanism is designed such that raters are paid based on how informative they (and hence their ratings) are to the evaluating agent. The informativeness is measured in two aspects: truthfulness and similarity.

3.4.1 Design

The mechanism, denoted by $\mathcal{M}$, adopts the notion of asymmetric reputation [Cheng and Friedman, 2005]. A target's reputation value is local to individual agent who evaluates the target. This asymmetric notion aligns with our goal of personalizing displayed reputation. We focus on the case where the agent who is evaluating a target's reputation has *not* interacted with thus has *not* rated the target, as reputation mechanisms are made precisely for providing an estimation of quality *prior to* engagement with targets. As before, we call such agents *evaluating agents*.

In $\mathcal{M}$, an agent i rates any single target *exactly once*, and she rates a target t by

making a report s_i consisting of a rating x_i^t and a predicted distribution of ratings y_i^t for the target, i.e. $s_i = (x_i^t, y_i^t)$. In order to compute a target t's reputation value from evaluating agent v's perspective, $\mathcal{M}$ performs a two-stage computation as below.

In the first stage, $\mathcal{M}$ computes the pairwise preference similarity between v and each of t's existing raters $r \in I(t)$, using v's historic ratings for other targets and r's historic ratings excluding r's rating for t, and we denote this preference similarity measure as $S^{-t}(v, r)$. Mathematically, by adapting (3.2), we compute

$$\begin{aligned}
S_v^{-t}(v,r) &= \frac{\lambda_1}{|O(v)|} \sum_{i=1}^{|O(v)|} \max_{j=1}^{|O^{-t}(r)|} S\left(O_i(v), O_j^{-t}(r)\right) \cdot S\left(x_v^{O_i(v)}, x_r^{O_j^{-t}(r)}\right) \\
S_r^{-t}(v,r) &= \frac{\lambda_1}{|O(r)-1|} \sum_{j=1}^{|O^{-t}(r)|} \max_{i=1}^{|O(v)|} S\left(O_i(v), O_j^{-t}(r)\right) \cdot S\left(x_v^{O_i(v)}, x_r^{O_j^{-t}(r)}\right) \\
S^{-t}(v,r) &= \min\left(S_v^{-t}(v,r), S_r^{-t}(v,r)\right) \qquad \text{for every } r \in I(t) \qquad (3.5)
\end{aligned}$$

where $O^{-t}(r) = O(r) - \{t\}$. In addition to the similarity score[3], $\mathcal{M}$ also computes the BTS score of each existing rater r's report using

$$\Pi_r(s_1, \ldots, s_{|I(t)|}) = \sum_{k=1}^{m} b(x_r^t, k) \log \frac{\bar{x}_k}{\bar{y}_k} + \sum_{k=1}^{m} \bar{x}_k \log \frac{(1-\epsilon)y_r^{t^k} + \epsilon}{\bar{x}_k} \qquad (3.6)$$

where

$$\begin{aligned}
b(x_r^t, k) &= \begin{cases} 1 & \text{if } x_r^t = k \\ 0 & \text{otherwise} \end{cases} \\
\bar{x}_k &= (1-\epsilon)\left[\frac{1}{|I(t)|} \sum_{i \in I(t)} b(x_i^t, k)\right] + \epsilon \\
\log \bar{y}_k &= \frac{1}{|I(t)|} \sum_{i \in I(t)} \log\left[(1-\epsilon)y_i^{t^k} + \epsilon\right]
\end{aligned}$$

[3]For brevity, we only highlight preference similarity here as this is ultimately what we need.

We note that (3.6) slightly differs from the original BTS payoff function (3.4), so that ill-defined values such as $\log 0$ and $\log \frac{0}{0}$ can be avoided. (3.4) is essentially (3.6) with $\epsilon = 0$, and we can minimize any distortion in incentives to arbitrarily small level by making ε sufficiently small. This minor modification also allows us to bound the BTS scores, making it possible to scale the scores to an appropriate range later.

Lemma 3.4. *The induced BTS game using the payoff function* (3.6) *is a zero-sum game, i.e.* $\sum_i \Pi_i = 0$.

Proof. We show that the average BTS score across all existing raters equals zero, hence the game is zero-sum.

$$
\begin{aligned}
\bar{\Pi} &= \frac{1}{|I(t)|} \sum_{r=1}^{|I(t)|} \Pi_i \\
&= \frac{1}{|I(t)|} \left(\sum_{r=1}^{|I(t)|} \sum_{k=1}^{m} b(x_r^t, k) \log \frac{\bar{x}_k}{\bar{y}_k} \right) + \frac{1}{|I(t)|} \left(\sum_{r=1}^{|I(t)|} \sum_{k=1}^{m} \bar{x}_k \log \frac{(1-\epsilon) y_r^{t^k} + \epsilon}{\bar{x}_k} \right) \\
&= \sum_{k=1}^{m} \left(\frac{1}{|I(t)|} \sum_{r=1}^{|I(t)|} b(x_r^t, k) \right) \log \frac{\bar{x}_k}{\bar{y}_k} + \sum_{k=1}^{m} \bar{x}_k \left(\frac{1}{|I(t)|} \sum_{r=1}^{|I(t)|} \log \frac{(1-\epsilon) y_r^{t^k} + \epsilon}{\bar{x}_k} \right) \\
&= \sum_{k=1}^{m} \bar{x}_k \log \frac{\bar{x}_k}{\bar{y}_k} + \sum_{k=1}^{m} \bar{x}_k \left(\frac{1}{|I(t)|} \log \frac{\prod_r^{|I(t)|} \left((1-\epsilon) y_r^{t^k} + \epsilon \right)}{(\bar{x}_k)^{|I(t)|}} \right) \\
&= \sum_{k=1}^{m} \bar{x}_k \log \frac{\bar{x}_k}{\bar{y}_k} + \sum_{k=1}^{m} \bar{x}_k \left(\log \frac{\sqrt[|I(t)|]{\prod_r^{|I(t)|} \left((1-\epsilon) y_r^{t^k} + \epsilon \right)}}{(\bar{x}_k)} \right) \\
&= \sum_{k=1}^{m} \bar{x}_k \log \frac{\bar{x}_k}{\bar{y}_k} + \sum_{k=1}^{m} \bar{x}_k \log \frac{\bar{y}_k}{\bar{x}_k} = 0
\end{aligned}
$$

□

Corollary 3.5. *If every rater* r *plays the same strategy* $s = \{x, y\}$ *in the induced BTS game defined by* (3.6)*, then* $\Pi_r = 0$ for every $r \in I(t)$ and any t.

Proof. We first observe that the BTS game is symmetric, that is, the payoff for playing a particular strategy depends only on the other strategies played, but not on which

rater is playing it. It follows that the payoff must be the same for every rater r when everyone plays the same strategy s. Given Lemma 3.4, this identical payoff can only be $\Pi_r = 0$ for every $r \in I(t)$. □

Lemma 3.6. *For any t and all $r \in I(t)$, $\Pi_r \in \left[2\log\epsilon, \log\frac{1}{\epsilon}\right]$.*

Proof. Given (3.6), we observe that $\epsilon \leq \bar{x}_k \leq 1$ and $\epsilon \leq \bar{y}_k \leq 1$. We then derive the lower and upper bounds of the BTS payoffs by examining the information and the prediction scores respectively.

For the lower bound of information score, we have

$$\begin{aligned}\sum_{k=1}^{m} h(x_r^t, k) \log \frac{\bar{x}_k}{\bar{y}_k} &\geq \sum_{k=1}^{m} h(x_r^t, k) \log \bar{x}_k \\ &\geq \sum_{k=1}^{m} h(x_r^t, k) \log \epsilon \\ &= \log \epsilon\end{aligned}$$

while the lower bound of prediction score is

$$\begin{aligned}\sum_{k=1}^{m} \bar{x}_k \log \frac{(1-\epsilon)y_r^{t^k} + \epsilon}{\bar{x}_k} &\geq \sum_{k=1}^{m} \bar{x}_k \log \frac{\epsilon}{\bar{x}_k} \\ &\geq \sum_{k=1}^{m} \bar{x}_k \log \epsilon \\ &= \log \epsilon\end{aligned}$$

combining the two, we have $\Pi_r \geq 2\log\epsilon$ for all r. Switching to the upper bounds, we have

$$\begin{aligned}\sum_{k=1}^{m} h(x_r^t, k) \log \frac{\bar{x}_k}{\bar{y}_k} &\leq \sum_{k=1}^{m} h(x_r^t, k) \log \frac{1}{\bar{y}_k} \\ &\leq \sum_{k=1}^{m} h(x_r^t, k) \log \frac{1}{\epsilon} \\ &= \log \frac{1}{\epsilon}\end{aligned}$$

for the information score, and the best possible prediction score is 0, as mentioned

before. Thus we have $\Pi_r \leq \log \frac{1}{\epsilon}$ for all r. □

We then apply a positive-affine transformation to (3.6) to re-scale the BTS payoffs to $[0, 1]$ using

$$\dot{\Pi}_r = \begin{cases} \text{undefined} & \text{if } \Pi_r = 0 \text{ for } \textit{all } r \in I(t) \\ \frac{\Pi_r - 2\log\epsilon}{\log\frac{1}{\epsilon} - 2\log\epsilon} = \frac{1}{3\log\frac{1}{\epsilon}}\Pi_r + \frac{2}{3} & \text{otherwise} \end{cases} \tag{3.7}$$

and we call $\dot{\Pi}_r$ the *truthfulness score*. At this point, we discard all the ratings with an *undefined* truthfulness score and those with a *zero* similarity score.

In the second stage, we combine the two scores for each r to obtain

$$\Omega_r = \dot{\Pi}_r + S^{-t}(v, r) \tag{3.8}$$

which measures, *from evaluating agent v's point of view*, how informative an existing rating from rater r is. The informativeness of a rating takes into account two aspects: how truthful and relatively accurate r's rating is and how similar r and v are in preferences.

Provided with individual Ω_r scores of existing ratings, $\mathcal{M}$ aggregates the ratings by first calculating the average Ω_r score for each rating value $k \in R$ and then taking the weighted arithmetic mean over all available rating values. Let $R^k = \{i | i \in I(t), x_i^t = k\}$ be the set of raters with rating value k. $\mathcal{M}$ derives the reputation value of target t for evaluating agent v as

$$\mathrm{R}_v^t = \sum_{k=1}^{m} w_k k \tag{3.9}$$

where

$$w_k = \frac{\frac{\sum_{i \in R^k} \Omega_i}{|R^k|}}{\sum_{k \in R} \frac{\sum_{i \in R^k} \Omega_i}{|R^k|}}$$

is the weight measuring on average how informative a particular rating value k is to

the evaluating agent v. Lastly, if no rating is available for the second stage, then target t remains unrated.

Finally, the payoff each rater $r \in I(t)$ receives from the mechanism is

$$\dot{\Omega}_r = \dot{\Pi}_r + \frac{\sum_{v \in V_r} S^{-t}(v, r)}{|V_r|} \tag{3.10}$$

where V_r is the set of agents whose evaluation of target t includes r's reports. The second summand rewards rater r based on on average how closely her preferences match with those of the evaluating agents who make use of r's submission, as a higher preference similarity makes her rating more informative to them.

3.4.2 Computation

In this section, we examine the computation process of the proposed mechanism $\mathcal{M}$. Given an evaluating agent v and a target t, $\mathcal{M}$ computes four values: pairwise similarity $S^{-t}(v, r)$ between v and each existing rater $r \in I(t)$; truthfulness score $\dot{\Pi}_r$ for each existing rater; combined information score Ω_r for each existing rating; and finally, displayed reputation value R_v^t. Combining them together, this process is formulated into Algorithm 3.1 using pseudocode.

We examine the computational complexity of this algorithm. To compute pairwise similarities, we can adopt the original approach in [Jeh and Widom, 2002] by iterating the calculation of equations (3.2) to a fixed-point. Given any two raters a and b, we iteratively compute $R^k(a, b)$ the preference similarity of a and b at the kth iteration, based on which we successively derive $R^{k+1}(a, b)$ and so on. We start with $R^{k=0}(a, b)$ where

$$R^0(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

Algorithm 3.1 Compute displayed reputation R_v^t for evaluating agent v

Require: $G = \{A, T, E, w\}$; $v \in A \wedge t \in T$
Ensure: R_v^t
if $I(t) = \emptyset$ **then**
  return $\mathrm{R}_v^t = 0$
end if
compute $S^{-t}(v, r)$ for all $r \in I(t)$ and t
compute $\Pi_r\left((x_1^t, y_1^t), \ldots, (x_{|I(t)|}^t, y_{|I(t)|}^t)\right)$ for all $r \in I(t)$
if $\Pi_r = 0$ for all $r \in I(t)$ or $S^{-t}(v, r) = 0$ for all $r \in I(t)$ **then**
  return $\mathrm{R}_v^t = 0$
else
  for all $r \in I(t)$ where $S^{-t}(v, r) \neq 0$ **do**
    compute $\dot{\Pi}_r$
    compute Ω_r
    compute $\dot{\Omega}_r$
  end for
  for all $k \in R$ **do**
    compute w_k
  end for
  return $\mathrm{R}_v^t = \sum_k^m w_k k$
end if

and then

$$R_a^k(a,b) = \frac{\lambda_1}{|O(a)|} \sum_{i=1}^{|O(a)|} \max_{j=1}^{|O(b)|} R^k\left(O_i(a), O_j(b)\right) \cdot S\left(x_a^{O_i(a)}, x_b^{O_j(b)}\right)$$

$$R_b^k(a,b) = \frac{\lambda_1}{|O(b)|} \sum_{j=1}^{|O(b)|} \max_{i=1}^{|O(a)|} R^k\left(O_i(a), O_j(b)\right) \cdot S\left(x_a^{O_i(a)}, x_b^{O_j(b)}\right)$$

$$R^{k+1}(a,b) = \min\left(R_a^k(a,b), R_b^k(a,b)\right)$$

for which there exist special cases

$$R^{k+1}(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } I(a) = \emptyset \text{ or } I(b) = \emptyset \end{cases}$$

[Jeh and Widom, 2002] shows that: $R^k(\cdot,\cdot)$ is nondecreasing in k, and that

$\lim_{k\to\infty} R^k(\cdot\cdot) = S(\cdot,\cdot)$; $S(\cdot,\cdot)$ always exists and is unique as the solution to the system of equations above[4]. Experiment results [Jeh and Widom, 2002, Li et al., 2009,Lizorkin et al., 2010] recommend that $k \in [5, 7]$ results a fast convergence. Given the graph $G = (A, T, E, w)$, the time complexity for the SimRank-based computation is $O\left(k(|A|^2 + |T|^2)d\right) < O\left(k(|A| + |T|)^2 d\right) = O(kn^2 d)$, where k is the number of iteration, $n = |A| + |T|$ is the total number of nodes, and d is the average of $|O(a)| \cdot |O(b)|$ or $|I(s)| \cdot |I(t)|$ over every pair of homogeneous nodes in G. This result comes from the observations that: there are $|A|^2$ and $|T|^2$ number of ordered node-pairs among agents and targets respectively; similarity score of a node-pair at each iteration is calculated using the scores of their neighboring pairs (either in- or out-neighbors), and there are on average d neighboring pairs for each pair.

Assuming a worst-case scenario where there are $|A|$ number of existing raters for a given target, the time complexity of computing (3.6) for each rater is simply $O(|A| \cdot m)$, where m is the number of rating values available. The positive-affine transformation takes linear time in the number of existing raters, so does the computation of their combined information scores. Computing all raters' payoffs takes maximally $O(|A|^2)$. Deriving the weights for all rating values can take $O(|A| + m^2)$, and the final calculation of the reputation value takes $O(m)$.

It should be noted that a number of optimization and approximation techniques exist for improving the computation of SimRank scores, such as [Jia et al., 2010, Li et al., 2010,Lizorkin et al., 2010]. While the details of these techniques are not within the scope of this chapter, we would like to point out that the computational cost of our SimRank-based algorithm may be further reduced by applying such methods.

3.4.3 Theoretical Properties

We show the following desirable properties of our mechanism.

[4]Note that in our formulation above, the numerical similarity $S(x, y)$ for any given pair of ratings $x, y \in R$ is always defined and bounded.

Proposition 3.7. *The mechanism $\mathcal{M}$ is computable in polynomial time.*

Proof. Given Algorithm 3.1 and our analysis in Section 3.4.2, step 4 takes less than $O(kn^2d)$, step 5 takes at most $O(|A|^2m)$, steps $6-8$ take $O(|A|)$, step $9-13$ take no more than $O(|A|^2)$, and steps $14-16$ take $O(|A|+m^2)$. Thus the whole Algorithm 3.1 runs within $O(n^2d)$, where n is the size of the input bipartite graph G, and d is the average number of incoming or outgoing link pairs between two homogeneous nodes in G. □

Proposition 3.8. *The mechanism $\mathcal{M}$ is incentive compatible. That is, given any target t, in an expected sense, each rater r maximizes her payoff $\dot{\Omega}_r$ for t by truthfully reporting her rating and prediction (i.e. $x_r^t = o_r^t$ and $y_r^t = d_r^t$), provided that the others are doing the same.*

Proof. Given our extended assumption of self-interestness, we analyze the two parts of $\dot{\Omega}_r$ (see (3.10)) in turn. For the similarity score part, $\frac{\sum_{v\in V_r} S^{-t}(v,r)}{|V_r|}$ measures the average similarity score between rater r and a set of evaluating agents, *without* taking into account r's rating for target t. Therefore, whatever opinions r reports about t does not affect $\frac{\sum_{v\in V_r} S^{-t}(v,r)}{|V_r|}$. Turning our attention to the truthfulness score $\dot{\Pi}_r$, which is an positive-affine transformation of the BTS score Π_r. Given Theorem 1, and $0 < \epsilon \ll 1$, we have

$$\begin{aligned}(o_r^t, d_r^t) &= \underset{(x_r^t, y_r^t)}{\operatorname{argmax}} (\Pi_r) \\ &= \underset{(x_r^t, y_r^t)}{\operatorname{argmax}} \left(\frac{1}{3\log\frac{1}{\epsilon}} \Pi_r \right) \\ &= \underset{(x_r^t, y_r^t)}{\operatorname{argmax}} \left(\frac{1}{3\log\frac{1}{\epsilon}} \Pi_r + \frac{2}{3} = \dot{\Pi}_r \right)\end{aligned}$$

in other words, truthful reporting $s_r = (o_r^t, d_r^t)$ strictly maximizes r's truthfulness score $\dot{\Pi}_r$ in expectation, assuming everyone else reports truthfully. Combining $S^{-t}(v,r)$ and $\dot{\Pi}_r$ additively, $\dot{\Omega}_r$ is maximized when $x_r^t = s_r^t$ and $y_r^t = d_r^t$. □

Proposition 3.9. *The mechanism $\mathcal{M}$ is resistant to manipulations in the following cases of collusion: 1. raters will not coordinate themselves for reaching any other equilibrium except for the truthful one; 2. a target remains unrated if it receives only identical fake reports; 3. a target is unaffected by fake reports submitted by colluding raters who rate only this target.*

Proof. The first claim follows from Theorem 3.3, as it proves that truth-telling is the *pareto-optimal* equilibrium of the BTS game. Given the self-interestness assumption, the agents should not collude on any worse-paying equilibrium and choose to play the truthful equilibrium that pays the most.

Second, by fake report we mean a pair of made-up rating and prediction submitted to manipulate a target's reputation. We consider an theoretically extreme but practically probable case where a target t only receives *identical* fake reports (of any rating, prediction and size). Because of Corollary 3.5, we obtain that $\Pi_r = 0$ for every $r \in I(t)$ and thus $\dot{\Pi}_r$ is undefined according to (3.7). By our design of $\mathcal{M}$, it follows that $\mathrm{R}_v^t = 0$ for all v, and t remains unrated as viewed by any evaluating agent. Note that even if fake reports are slightly different, as truthful reports come in and become surprisingly common, the weights on fake reports will diminish and the manipulation will become ineffective.

Third, considering a common case where a group of colluding raters C rate a target t and t only, our SimRank variant will derive zero similarity score between every colluding rater $r \in C$ and any evaluating agent v. By our design of $\mathcal{M}$, it follows that all colluding reports from C will be filtered out at the end of the first stage of the mechanism, thus the target t is unaffected by the collusion. □

While filtering by zero similarity score might exclude some genuine ratings initially, we argue that it is still a worthwhile tactic as genuine raters are much more likely than colluding raters to rate more than one target, so that their reports will be taken into account sooner or later.

3.4.4 Pragmatic Considerations

In this section we discuss some practical considerations for implementing our mechanism $\mathcal{M}$.

Payoff : As implied by our design, the payoff of a rater for a given target is not fixed and will be updated as new ratings come in. Therefore, an accounting system may be needed to keep track of such changes. We may also need timestamps for each rater's reports, so that we can track down the set of evaluating agents corresponding to each rating. The next section provides additional information. Moreover, there exist two possible implementations of the payoffs: reputation points for higher visibility and privileges within the user community, or those translated into monetary prizes such as discounts. The question of which form of payoffs (bragging rights vs. materialistic rewards) is more effective in incentivizing raters remains a separate but interesting research topic.

Score updates : Both the similarity scores $S(v, \cdot)$ and truthfulness scores $\dot{\Pi}_r$ for a given target t need to be regularly updated to extract the latest informative ratings for any evaluating agent v, as well as to update existing raters' payoffs. For similarity scores, we think it is suffice to perform a system-wide update periodically (e.g. ranging from once per day to once per week), as each update covers the whole rating graph and should capture as many changes in rating relationships as possible since the last update[5]. An incremental real-time updating approach may not be economical and necessary here. On the other hand, the truthfulness scores $\dot{\Pi}_r$ can be updated for each new rating, since it is local to individual targets and the computation takes linear time. The subsequent aggregation computation can also be computed in real-time, for which the sim-

[5]We note that the mechanism needs to compute a separate version of similarity scores for each target t, as it needs to exclude every existing rater's rating for each t. However, such a system-wide update still computes in polynomial time.

ilarity score component will remain static between updates. The only caveat here is that in order to apply the BTS method we need to conceal the actual distribution of ratings.

Evaluation after self-submission : We consider two options for an agent v evaluating a target's reputation after she herself has submitted a rating. First, we can simply display two separate reputation values: the rating value submitted by v, and the reputation value aggregated by $\mathcal{M}$ that may or may not include v's report. This option respects an agent's personal opinion, while showing the aggregated reputation as a reference. The online movie database IMDb.com (`www.imdb.com`) adopts this approach. The second option is a single aggregate reputation value using our mechanism, in which v's report is evaluated with all other reports using the BTS while her similarity score with herself is one. This approach maximally respects the mechanism, without differentiating between the evaluating agent's own report and the others'.

Number of reports : The BTS method relies on a sufficiently large population of agents to enforce truthful reporting. Admittedly, this is a constraint in theory. In practice, this requirement may be relaxed as long as the number of rating values is significantly smaller than the number of reports [Carvalho and Larson, 2011][6]. A feasible tactic here is to withhold computing and publishing the reputation value of a target before this condition is met, as it is a common for many rating websites to require a certain minimum number of ratings to display any reputation value.

'Blackbox' approach : Many real-world reputation systems keep their algorithms secret in order to resist malicious attacks. We can partially adopt this approach

[6] [Carvalho and Larson, 2011] suggests that roughly m^2 number of reports are needed to meet the condition, which translates into 25 reports for a typical five-star scale ($m = 5$). As of the end of 2011, there were over 25 million ratings and about 600,000 business listings on Yelp (`www.yelp.com`) [Yelp Inc, 2012].

too, by making only the working of the BTS known to raters. Given our analysis in Proposition 3.8, the knowledge (or the lack of) of similarity scoring does not affect the incentive compatibility, and the raters can maximize their payoffs only by truthful reporting. By concealing the full working of the mechanism, we make it even harder for manipulations to succeed.

3.5 Discussion

In this chapter, we designed a reputation mechanism for online ratings. We recognise that most consumer ratings online are subjective in nature, and set out to construct a mechanism that not only elicits truthful ratings (personal opinions) but also personalises them to individual users of the mechanism when they evaluate a given target. To do so, we selected the SimRank and the BTS algorithms from separate fields of research and customised both of them to suit our problem of interest. We then built our own two-stage mechanism using our adapted SimRank and BTS, which crowdsources personal opinions to differentiate and promote informative ratings for individual evaluating agents. Given our design, the mechanism enforces truthful reporting by preserving the desirable game theoretic properties of BTS and personalises the displayed reputation for each evaluating agent by utilising data mining features of SimRank. By theoretically showing various properties of our mechanism, we believe that it improves in three areas of online ratings: 1. Genuine rates are encouraged to report as truthfully as possible, assuming that they aim to maximise their payoffs from the mechanism; 2. Common cases of collusion using fake ratings can be deterred after filtering of the mechanism; 3. Displayed reputation is personalised to the evaluating agent based on her preferences similarity to other raters. Note that it is also possible to replace the two main components with improved algorithms in the future to enhance the mechanism. The central idea remains the same: the game

theoretic algorithm enforces truthful reporting of personal opinions while the data mining algorithm helps individualise these opinions to derive the most informative reputation for each evaluating agent.

Some may argue that a weakness of the BTS algorithm is that it assumes that all participating agents reason and behave based on Bayesian updating. However, there is a substantial amount of experimental evidences which suggest that humans rely on their own choices to gauge the popularity of those choices among others. [Marks and Miller, 1987] provides a review of a number of experimental studies that replicate the so-called false consensus effect, such that, people often egocentrically assume that others are similar to themselves. Moreover, [Dawes, 1989] demonstrates that there exists a positive correlation between one's own opinion and the consensus opinion, indicating that the prediction of others' behaviour correlates with one's own behaviour. Such findings offer behavioural basis to the practicality of the BTS algorithm. Indeed, [Shaw et al., 2011] provides further experimental support to the effectiveness of the BTS in sourcing user-generated ratings.

For future improvement, one could enhance the BTS component of the mechanism by further relaxing its assumptions and making it robust to more sophisticated collusion. Furthermore, the possibility of offering more detailed statistics of the ratings *without* compromising the truthfulness of the BTS deserves further exploration, as the lack of such statistics may create a lack of transparency from user's point of view. Some efforts [Witkowski and Parkes, 2012b, Radanovic and Faltings, 2013] have been made in this respect. Furthermore, one could incorporate social elements into the SimRank component of the mechanism so that not only users' similarity in ratings but also their social relationships are taken into account by the mechanism. Social relationships may provide another useful dimension to measuring the informativeness of one's rating, and several studies offer promising directions in this area [Jamali and Ester, 2010, Ma et al., 2011, Yang et al., 2012, Sharma and Cosley, 2013].

Chapter 4

Synergistic effects of reputational and social knowledge on cooperation

The emergence and sustenance of cooperation is essential for societies to thrive, and has been a fundamental area of inquiry across the social and biological sciences. It is thought that a large share of the population are conditionally cooperative (one's own cooperation being conditional on others' cooperation) while the rest consists of determined cooperators and defectors. Given a mixed population, it is difficult to maintain cooperation without explicit enforcement such as punishment and reward, since conditional cooperators are prone to propagate defection when interacting with defectors.

Extensive research have put forward many theories on what may sustain cooperation in various settings [Fehr and Fischbacher, 2004, Perc and Szolnoki, 2010, Rand and Nowak, 2013]. A significant amount of research have focused on cooperation in a network context. After all, humans routinely engage each other in social networks, and their behaviours are influenced by their social ties. Theoretical studies [Ohtsuki

et al., 2006, Santos et al., 2006, Taylor et al., 2007, Saavedra et al., 2009] propose network reciprocity as a mechanism for supporting the evolution of cooperation in static networks, and some show that network structures may play an important role in promoting cooperation [Lieberman et al., 2005, Santos et al., 2008]. However, empirical studies using behavioural experiments have found little effect on promoting cooperation by static network structures: cooperation declines over time and there is no significant difference in level of cooperation between different structures [Cassar, 2007, Kirchkamp and Nagel, 2007, Grujić et al., 2010, Traulsen et al., 2010, Suri and Watts, 2011, Gracia-Lázaro et al., 2012, Grujić et al., 2014].

A related but distinct mechanism offers a potential solution to promoting cooperation, namely network dynamics. Earlier studies neglect that not only behavioural strategies but also network connections can evolve over time through repeated interaction. This co-evolution of behaviour and social structure offers a powerful mechanism for supporting cooperation, as the capability of forming and breaking social links offers individuals an alternative yet effective way to reciprocate past behaviour. It is also intuitive that agents tend to manipulate their social connections based on past treatments by others: they keep or form beneficial relationships and discard or refuse damaging ones. The latter serves as retaliation as well as prevention of defection. Credible social exclusion (or in its extreme form, ostracism) may isolate defectors and may even convert their behaviour if a strong cooperative norm is established.

In this chapter, we experimentally explore the simultaneous evolution of cooperation and social structure using a repeated game. More specifically, we investigate the relative importance of reputational knowledge (what you know abut others' past actions) and social knowledge (information on who is connected to whom) in the emergence and sustenance of cooperation by running a series of behavioural experiments on the Internet. We set up groups of 13 subjects playing the prisoner's dilemma game with their chosen connections. Subjects are free to form and cut as many connections

as they deem necessary, and they do not discriminate their choices of action among their connections. We conduct 4 treatments that vary in the degree of information available to subjects and compare the resulting cooperative behaviours and network structures.

4.1 Related Studies

Four studies have experimentally showed that dynamic networks can promote cooperation [Rand et al., 2011, Fehl et al., 2011, Wang et al., 2012, Shirado et al., 2013]. Although all these studies allow subjects to update their links, they adopt different experimental designs in almost every other aspects. Some of these differences are summarised in the table below.

Study	**Size**	**Length**	**Initial Network**	**Link Update**	**Action Info**
[Rand et al., 2011]	20	11	20% of all possible links in each round	10% or 30% of all possible links in each round	last round
[Fehl et al., 2011]	10	30	regular network of degree 3	all possible links in each round	last round (neighbours only)
[Wang et al., 2012]	24	12	4 cliques or regular network of degree 5	every r rounds up to k updates per player, $r = 1, 3, 6$ and $k = 1, 3, 5$	last 5 rounds
[Shirado et al., 2013]	17	15	random network	p% ($p \in \{0, 5, 10, 30, 50, 70, 80, 90, 100\}$) of all possible links in each round	last round

Table 4.1: Comparison of experimental designs between previous studies

Furthermore, these studies implement different rules for link updating. [Rand et al., 2011, Fehl et al., 2011, Shirado et al., 2013] use similar "active linking" schemes [Pacheco et al., 2006]. Subjects cut existing links that are randomly chosen in each

round. To form new links, [Rand et al., 2011, Shirado et al., 2013] randomly pick unlinked pairs and asks for approval from one subject (randomly picked again), whereas [Fehl et al., 2011] randomly designates a new partner without approval to replace the cut link. [Wang et al., 2012] uses a more endogenous process in which subjects themselves choose whom to link to and unlink from, and new links require mutual consent from both ends. Moreover, in contrast to the common design of the prisoner's dilemma game on networks, [Fehl et al., 2011] allows subjects to choose independent action towards each of their connections. Lastly, all four studies use the sum of points or money won over the entire course of experiment for subject payment.

All these studies find that dynamic networks sustain cooperation to different extents. [Fehl et al., 2011] finds that allowing subjects to cut links significantly increases cooperation even if each subject can only have a maximum of three links and new links are created randomly. It also indicates that cooperators form clusters through link updating. [Rand et al., 2011] finds that cooperation evolves only when the number of updated links (i.e. formed or broken) per round reaches over 30%. Slower rate of link updating helps as little as fixed and random networks do. Moreover, cooperative connections are longer lived and cooperative subjects have more connections. [Wang et al., 2012] also finds that link updating supports cooperation by experimenting with a number of updating rates and payoffs. It also demonstrates that the effects of updating rate on cooperation is concave (i.e. the marginal return or benefit of increasing the rate is decreasing), and that the cost of meeting a defector needs to outweigh the benefit of meeting a cooperator in order to effectuate promotion of cooperation. Lastly, [Shirado et al., 2013] finds that intermediate re-wiring rates best promote cooperation in their settings[1].

[1]Both the formation and cutting of links are unilateral

4.2 Motivation

We seek to advance the understanding of the sustenance of cooperation in dynamic networks, more specifically, we focus on the effects of reputational and social knowledge on cooperation.

4.2.1 Reputation and Cooperation

A well-studied mechanism to encourage cooperative behaviour is reputation: having access to an individual's past actions allows the reciprocation of cooperative behaviour [Fudenberg and Maskin, 1986, Milinski et al., 2002]. Given reputational knowledge of others, individuals are able to exercise direct and indirect reciprocity in both repeated and fresh interactions. Especially in the case of indirect reciprocity, reputational knowledge is essential to enable reward to cooperators and punishment to defectors [Nowak and Sigmund, 1998, Milinski et al., 2001]. Furthermore, when reputational information is made available, individuals may choose to strategically build up their reputation through which cooperation emerges [Engelmann and Fischbacher, 2009].

Simulation-based studies [Fu et al., 2008, Cong et al., 2012, Chen et al., 2012] show that reputation is also important to cooperation when interaction becomes selective. In experiments of dynamic networks [Rand et al., 2011, Fehl et al., 2011, Wang et al., 2012], reputational knowledge[2], of various degrees, is always accessible to subjects by default. No experimental study has directly examined reputational knowledge as a treatment variable in the context of dynamic networks.

4.2.2 Network and Cooperation

Network reciprocity has long been proposed as a mechanism for sustaining cooperation [Nowak and May, 1992, Lieberman et al., 2005, Ohtsuki et al., 2006]. It is believed

[2]More precisely, *global* reputation information is available such that subjects are able to observe past actions of an unlinked player.

that spatial structure determines interactions between individuals, and cooperators thrive by assorting themselves into clusters. Studies have shown theoretically and by simulation that heterogeneous network structures can support cooperation [Santos and Pacheco, 2005, Pacheco et al., 2006, Santos and Pacheco, 2006]. Nevertheless, as aforementioned, a number of experimental studies fail to find significant positive effects of static network structures on promoting cooperation. Possible causes of these negative results may be that people do not always imitate each other so that behaviour assortment does not occur on static structures and that behavioural reciprocity (direct or indirect) alone does not distinguish well between defectors and cooperators when it comes to punishment[3], thereby spreading uncooperative behaviour within the fixed group [Grujić et al., 2012].

Recent experimental contributions show that cooperation emerges if individuals are able to choose their partners. In all these studies the subjects know who they are connected to, but they do not have access to the social knowledge of who is connected to whom in the network. In general, there is little empirical research on the effects of social knowledge on cooperation or coordination. To our best knowledge, [Enemark et al., 2014] is the only experimental study so far that investigates the impact of this knowledge (in static networks). They find that more knowledge about network structure leads to better performance in decentralised coordination task and the magnitude of improvement depends on the network structure.

4.2.3 Reputational and Social Knowledge

In contrast to prior efforts, we investigate the relative importance of reputational and social knowledge, and how they may complement each other to promote cooperative behaviours. In dynamic cooperative networks, reputational and social knowledge are inextricably related. Reputation determines decisions to form connections, and it

[3]This may be dictated by the fact that subjects are asked to choose a single action that applies to *all* of her connections in these experiments.

therefore shapes the structural features of the social network. Social structure is a conduit of reputational information about neighbours in the absence of an explicit reputation mechanism, and social knowledge may help individuals to identify loci of cooperative activity.

There has been no study that examines the effectiveness of social knowledge at promoting cooperation, and how it varies depending on whether reputational knowledge is only available locally through the network or if reputational information is available on everyone. Furthermore, we do not know how reputational and social knowledge complement each other and how much each contributes towards the emergence of the network features that are associated with cooperative activity. We experimentally explore the answers to these questions by conducting Web-based experiments, and produce the first results on the relative importance of these knowledge and on their combined effects.

4.3 Experiments

We conducted all our experimental sessions using subjects recruited from Amazon Mechanical Turk (AMT), a popular online labour market. Workers (called Turkers) at AMT complete short-term tasks (called Human Intelligence Tasks or HITs) in return for small monetary compensation. AMT hosts a large and diverse population of Turkers, which provides researchers with on-demand access to its labour [Paolacci et al., 2010, Horton et al., 2011, Rand, 2011, Mason and Suri, 2012]. AMT enables researchers to conduct research with significantly larger and more diverse subject pools compared to the typical laboratory experiments.

4.3.1 Method and Design

In our experiments, all the subjects are U.S-based and we enforce this requirement by restricting our HITs to U.S. Turkers as well as checking their IP addresses upon participation. All collected data are associated with subjects' Turker IDs only, and not with any personally identifiable information. There are considerable differences in recruitment and experimental procedures among the few published studies that involve real-time interaction between subjects from AMT. In particular, some previous studies [Suri and Watts, 2011, Mason and Watts, 2012, Wang et al., 2012] allow subjects to participate in multiple sessions of the same experiment: subjects are allowed to participate in the same treatment multiple times and in different treatments. Moreover, their main analysis is based on data collected in later sessions only. In contrast, our experiment actively prevents multiple or repeated participation even in cases where the subject only saw the Instructions and then abandoned the experiment. Similar to other contributions [Rand et al., 2011, Rand et al., 2012, Shirado et al., 2013], our study requires subjects to read the Instructions and complete the experimental Game in a single uninterrupted session, emulating the workflow of laboratory settings.

Participation in our experiment consists of two parts. First, subjects have to pass a Qualification HIT in which we collect demographic information, and ask basic comprehension questions about the Janken Step game [Dixit and Nalebuff, 2010], which is a simple variant of the popular rock, paper and scissors game. The comprehension questions serve the purpose of filtering out a small minority of subjects who are extremely careless in reading and/or have difficulties in understanding the simple Janken Step game and its payoff matrix. We believe that this filtering leads to a negligible bias in our subject pool, because the chosen game is so simple and common that it merely requires rudimentary reading and arithmetic skills (i.e. addition of single-digit numbers) and the failure rate is only 6.6%. Second, we randomly invite qualified subjects by email using AMT's ID-based notification system. Invitations

were sent individually so that subjects did not know who else was invited in the same session. In our invitation email, we specify the time and date of the experiment and do not disclose any details of the experimental design. To participate in the actual experiment, subjects need to locate and accept the HIT, and click on a URL to access an external web page hosted on *UbiquityLab*, our purpose-built platform for online experimentation. The experiment consists of the following steps: Waiting Room; Instructions and Interface Tour; Quiz; Game; Final Questionnaire; and Payment Confirmation. At the end of the experiment subjects need to copy a unique confirmation code and submit it on the AMT HIT page for payment. In our main analysis, we use data collected from 364 subjects who constitute a more diverse subject pool compared to the typical student sample in laboratory studies (see Appendix A.1).

During the Game, each subject is assigned to a group of 13 and plays $13 - 16$ rounds of a multiplayer prisoner's dilemma game on an endogenous network. Each round consists of 3 stages. First, subjects can propose costless links to any of the other subjects and can unilaterally remove any of their existing links. Second, subjects accept/reject link proposals from others. Third, they play a prisoner's dilemma game by choosing a cooperate (C) or defect (D) action that applies to all their neighbours on the resulting network. The first round starts with the empty network with no link between subjects. The game is completely symmetric in payoffs: (C, C) gives 3 points to each subject, (D, D) gives -3 points, (C, D) gives 5 points to the defector and -5 points to the cooperator, and both subjects get 0 points if they are not linked. The symmetry of the payoff structure leads to intuitive welfare implications: a society of social isolates has welfare 0, and changes to welfare are exclusively driven by the (C, C) and (D, D) links. In the data the welfare level is driven by the number of (C, C) links so in our analysis we focus on the level of cooperation, which we define as the links where both subjects choose a cooperative action as a proportion of the

78 potential links in the network. The Game lasts for at least 13 rounds and at most 16 rounds. Starting from the 13th round, the game may randomly terminate with a 50% probability, which subjects are informed about in the Instructions. At the end of the experiment for each subject we select 12 pairings (2 in each of the 6 randomly selected rounds) and convert the sum of points won in those pairings into dollars. Notice that a subject could have a randomly drawn pairing with a subject she was not linked with in the drawn round and in that case she would get 0 points, so from a game-theoretic point of view this is an extended prisoner's dilemma game where the "no link" action gives 0 payoff.

We conduct 4 treatments (30 sessions in total)[4] to examine the relative importance of reputational and social knowledge. In the baseline (B) treatment subjects only have access to local knowledge on both dimensions, who their neighbours are and the last 5 actions chosen by each of these neighbours. In the reputation (R) treatment they have access to global reputational knowledge and local social knowledge, who their neighbours are and the last 5 actions of every other subject. In the network (N) treatment they have access to global social knowledge and local reputational knowledge, who is connected to whom in the whole network and the last 5 actions of their neighbours. Finally, in the reputation and network (RN) treatment they have access to global knowledge on both dimensions, who is connected to whom in the network and the last 5 actions for every other subject.

Our graphical game interface requires mouse inputs only, and provides information to subjects depending on the current stage of the game and treatment.

In the first stage, subjects see their own previous five actions, their current neighbours' previous five actions, and, in treatments R and RN only, their non-neighbours' previous five actions. In all treatments, there is a figure that shows the subject's cur-

[4]In 2 of the 30 sessions, some subject(s) dropped out of the experiment after the game started. We exclude these 2 sessions in all of our main analysis (which uses the data from 4 treatments $\times$ 7 sessions per treatment = 28 sessions), but we perform additional robustness checks with the inclusion of these data and the results are available in Appendix A.3.

rent neighbours. In the N and RN treatments, there is also an interactive figure (see Figure 4.1) that displays the current network in the group: subjects can highlight any node and its neighbour(s) by hovering the mouse pointer onto the node, and they can also rearrange the network layout by dragging the nodes around as the link(s) will automatically adapt to the changes. In the B and R treatments, the network figure is replaced by a figure of all the non-neighbours.

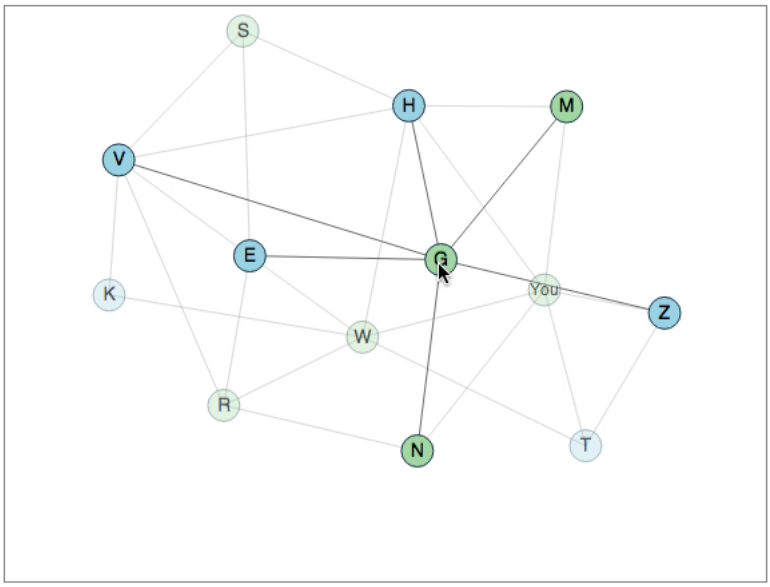


Figure 4.1: Interactive network figure of the links among all the subjects in the RN treatment.

In the second stage, subjects see the same reputational and network information as in the first stage. Additionally, they receive notification(s) of link proposals from any non-neighbour who has chosen to propose a link in the first stage, and have to decide whether to accept or reject each proposal. In treatments with network information, they can continue interacting with the network figure while making their decisions.

In the third stage, both network and reputational information are updated to reflect the outcome of the link formation or removal choices and proposal acceptances or rejections in stages 1 and 2 respectively. Subjects observe the updated information about their neighbours. In treatments B and N, subjects can see the reputational

information of their new neighbours, and cannot access any longer the reputational information of removed neighbours. Based on the updated network and reputational information available in each treatment, each subject chooses to cooperate or defect.

Visually, the interface is organised into three main panels, as shown in Figure 4.2.

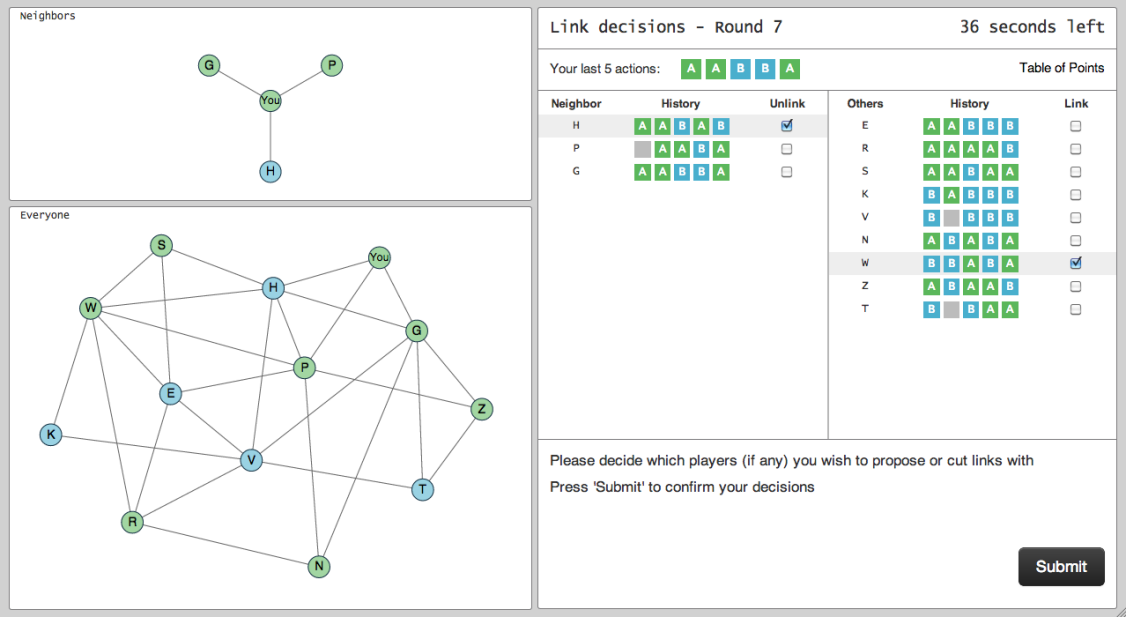


Figure 4.2: Three-panel design of the game interface. The screenshot shown here is stage 1 of a round in the RN treatment.

The top left panel shows the graphical representation of the subject's current neighbours, with the subject herself at the center. The bottom left panel shows the graphical representation of either all other unlinked subjects (in B and N) or all the subjects including the subject herself (in R and RN). In R and RN, the network is laid out using a customised force-based layout algorithm [Bostock et al., 2011] that we fine-tune specifically for networks of 13 individuals. In each stage, the network layout is exactly the same for all subjects in a session except for the labeling of the "You" node which denotes the position of the subject herself. We performed extensive testing to ensure that the algorithm effectively captures and visualizes important structural features such as symmetries and regularities in the network.

The right panel contains four blocks from top to bottom:

The first block indicates the current stage and current round of the game, and the time left for making an input. In the first stage, subjects have 70 seconds to propose and cut links. In the second stage, subjects have 45 seconds to accept or reject any link proposal(s) they may have received. In the third stage, subjects have 25 seconds to choose an action. At the end of a round, subjects have 10 seconds to view the results.

The second block reminds the subject of her own past actions. The "Table of Points" link on the right allows the subject to see the payoff information of the game by hovering her mouse pointer on the link.

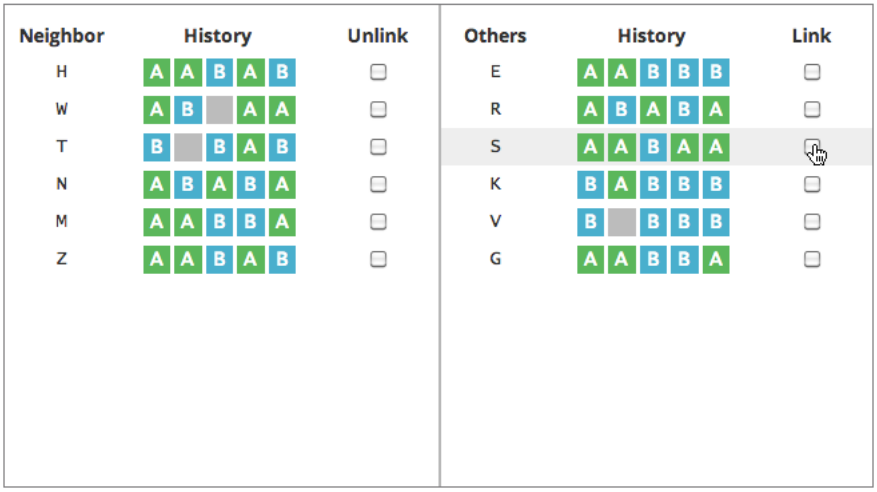


Figure 4.3: In stage 1 the third block in the right panel displays other subjects' previous actions and allows subjects to propose and cut links. The example screenshot above is what a subject would see in treatments R and RN with global reputational information. A green box with an "A" indicates that the subject has chosen to cooperate, a blue box with a "B" indicates that the subject has chosen to defect, and a gray box denotes that the subject has not taken an action. For instance, in the screenshot above we have that in the previous round subject H chose to defect and N chose to cooperate. Notice that if a subject does not take an action then the computer selects the defect action by default.

In the first stage, the third block lists the neighbours (on the left) and non-neighbours (on the right) that are carried over from the last round, and their respective actions in the previous five rounds. In the B and N treatments with local

reputational information, subjects can only see the history of actions of their neighbours. Subjects are free to propose and cut any link by ticking the box next to the fictitious initial of any other subject (see Figure 4.3). In the second stage, any unmatched link proposal is shown in the non-neighbours list: a "Y/N" button appears next to the non-neighbour who sent the proposal, and the subject then needs to choose 'Y' (for Yes) or 'N' (for No) for each proposal (see Figure 4.4). In the third stage, both neighbours and non-neighbours lists are updated based on the results of the earlier stages.

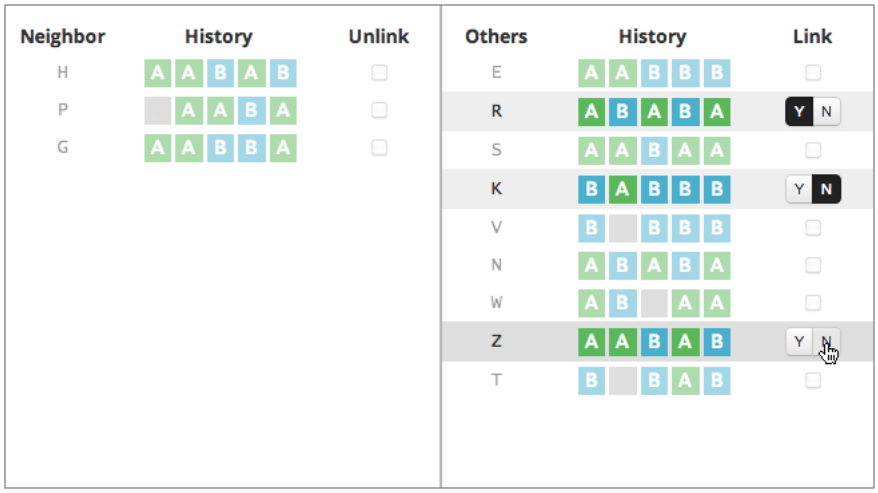


Figure 4.4: In stage 2 the third block in the right panel displays the reputational information of other subjects and allows subjects to approve or reject other subjects' link proposals

The bottom block indicates what the subject needs to do in each stage and shows the button(s) for making an input. In the first two stages, it displays a "Submit" button the subject can use to confirm her choices. In the third stage, it displays the choices between the "A" (denoting cooperate) and "B" (denoting defect) buttons, and a separate "Submit" button for confirming the choice (see Figure 4.5).

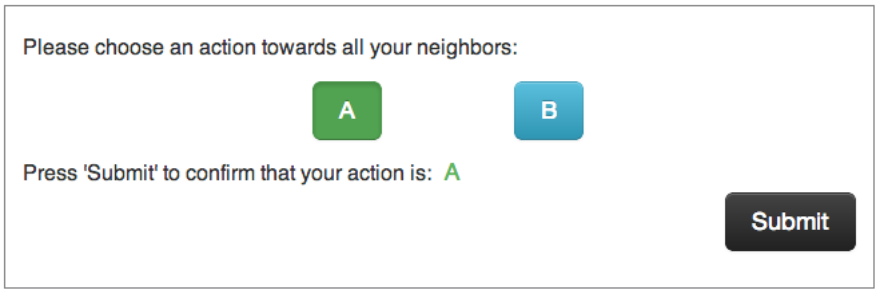


Figure 4.5: In stage 3 the bottom block allows subjects to choose between "A" (cooperate) and "B" (defect)

We include in Appendix A.4 the sample Instructions for treatment B. Complete Instructions of other treatments are available upon request.

4.3.2 Analysis and Results

Our data consists of a total of 9 meta-sessions which took place at 11am EST between 12th and 21st of December, 2013. Each meta-session comprises between 2 and 5 simultaneous experimental sessions, with the exact number depending on turnout. We dissect our data and carry out analysis at the following levels: aggregate group level, derived community level, and individual subject level.

We first present the results at aggregate group level across treatments. Figure 4.6a shows the evolution of the level of cooperation: the availability of global reputational knowledge is the prerequisite for the emergence and sustenance of a high level of cooperation. Subjects in the RN treatment achieve an average cooperation level which is significantly larger than the B (rank sum test, $P = 0.013$) and N (rank sum, $P = 0.025$) settings. The R treatment shows that global reputation alone accounts for most, but not all, of the effect: it leads to a cooperation level that is marginally larger than the B treatment, but the difference is insignificant compared to the N treatment. Consequently, subjects in the global reputation R and RN treatments achieve payoffs that are approximately twice as large as subjects in the N and B settings (Fig. 4.6b). The high cooperation level in the RN setting is associated with

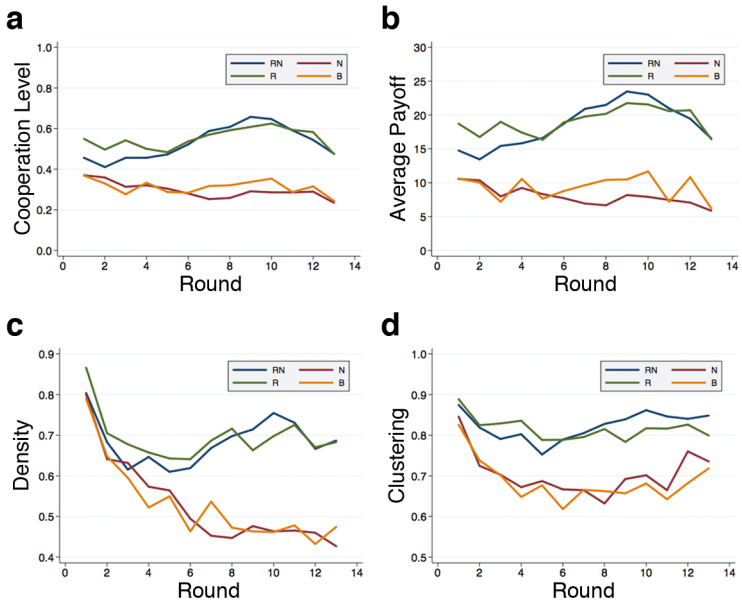


Figure 4.6: Cooperation level **(a)**, average payoff **(b)**, density **(c)**, and clustering **(d)** over 13 rounds of play for treatments B (yellow), N (red), R (green) and RN (blue). Analyses are based on aggregate data after round 5 in each session. Density denotes the number of links as a proportion of the 78 potential links in the network. Global clustering denotes the number of triangles (3 nodes connected by 3 links) as a proportion of the 858 potential triangles in the network. These are the results for Rank-sum tests comparing the differences across treatments, we only include significant ($P < 0.05$) and close to significant ($P < 0.1$) findings: **(a)** Cooperation level: RN vs. B, $P = 0.013$; RN vs. N, $P = 0.025$; R vs. B, $P = 0.047$; R vs. N, $P = 0.064$; **(b)** Average payoff: RN vs. B, $P = 0.013$; RN vs. N, $P = 0.025$; R vs. B, $P = 0.048$; R vs. N, $P = 0.048$; **(c)** Density: RN vs. B, $P = 0.002$; RN vs. N, $P = 0.013$; R vs. B, $P = 0.025$; R vs. N, $P = 0.048$; **(d)** Global clustering: RN vs. B, $P = 0.006$; RN vs. N, $P = 0.009$; R vs. B, $P = 0.064$; R vs. N, $P = 0.064$

the emergence of a denser and more clustered network (Fig. 4.6c-d) than the B (rank sum, $P = 0.002$ for density, $P = 0.004$ for clustering) and N (rank sum, $P = 0.013$ for both) settings. The network emerging in the R treatment is denser than the baseline B and marginally denser than the N setting, but it displays no significant difference in clustering level. The stronger significance of the cooperation level and network features associated with high cooperation in the RN setting suggest that the access to global social knowledge plays a synergistic role when it comes to shaping the network structure most suitable for a highly cooperative society.

We also verify that treatment RN enjoys a larger share of cooperators in each round comparing to treatments B (rank sum, $P = 0.021$) and N (rank sum, $P =$

0.025), while the share of cooperators in R is only marginally larger than that in N (rank sum, $P = 0.047$) and is not significantly larger than that in B (rank sum, $P = 0.063$).

In the RN and N settings, we display information about the social network through an interactive figure that allows subjects to highlight the neighbours of a node by mouse hovering and to rearrange the network layout by dragging nodes around. Mouse movement tracking reveals that subjects make active use of the network information. We include auxiliary information of subject behaviour during participation in Appendix A.2.

Next, given the accessibility of social knowledge and subjects' utilisation in treatments N and RN, we further explore the role of this knowledge and the differences in the emerged network structure across treatments by using the Louvain [Blondel et al., 2008] algorithm that detects communities in the network.

The Louvain method adopts a hierarchical design to greedily partition the network into communities that obtain the highest modularity. Modularity is defined as:

$$Q = \frac{1}{2L} \sum_{C} (2l_C - \frac{K_C^2}{2L})$$

where L is the total number of links in the network, C denotes a community, l_C is the number of links in community C, $D_C = \sum_{i \in C} d_i$ is the sum of the number of links, or degree, d_i for each subject $i \in C$. The algorithm starts by placing each node in its own community and seeks to merge it with neighbouring communities such that the modularity gain is maximised by iterating through every node. A node that is previously merged to a community can be detached to join a new community in subsequent iterations, which allows the algorithm to correct initial suboptimal choices. This phase is repeated until it is not possible to move a node to achieve an increase in modularity. The second phase builds a new network by collapsing all

the nodes of the same community into a "super node", and then it runs the first phase with this new network consisting of super nodes. This alternating two-phase procedure is repeated several times until there is no more improvement in modularity. The pseudo-code in Algorithm 4.1 summarizes the algorithm.

Algorithm 4.1 Louvain Community Detection Method

Require: G the original network
1: **loop**
2: Initialise each node of G as a cluster
3: **while** some node(s) can be moved **do**
4: **for** every node i in G **do**
5: Move i to a neighbouring community to maximise the modularity gain if possible, otherwise retain i in its own community
6: **end for**
7: **end while**
8: **if** resulting modularity is higher than before **then**
9: Collapse each cluster into a super node
10: $G \leftarrow$ the network of super nodes
11: **else**
12: **return** resulting clusters and terminate
13: **end if**
14: **end loop**

Using the Louvain method, we uncover communities in every round of the game and order them by size in descending order. Figure 4.7a shows the distribution of the number of detected communities in each treatment. There is no difference across treatments in terms of the fragmentation of the networks. The most frequent outcome of the algorithm is the decomposition of the network into two communities, followed by the decomposition into three communities. There is no significant difference across treatments in terms of the number of communities that emerge in the networks. Figure 4.7b shows the distribution of the mean sizes of communities for each treatment. We find that community 1 ($C1$ hereafter) is between 30% (B and N treatments) and 60% (R and RN) larger than community 2 ($C2$ hereafter), and together they encompass over 85% of the group in all the treatments.

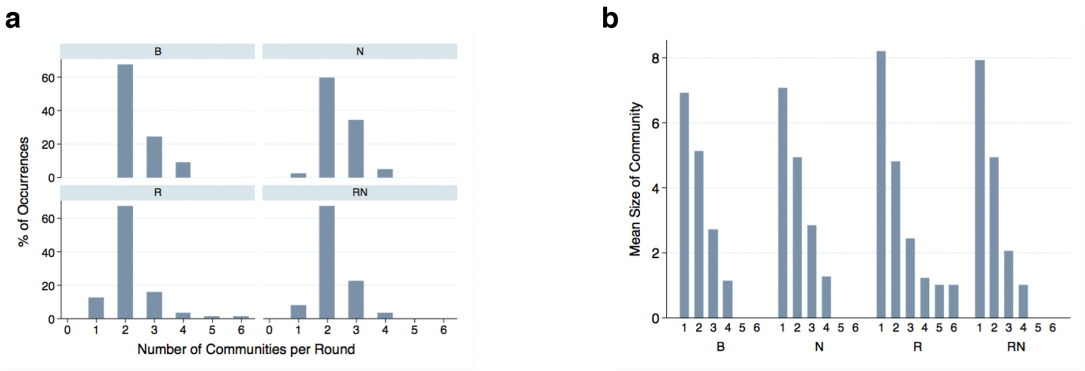


Figure 4.7: (a) Distribution of the number of detected communities in each treatment: the percentages are calculated using $\frac{\text{number of occurrences}}{13\times7}$, where 13 is the total number of rounds in each of the 7 sessions in a treatment. (b) Distribution of the mean sizes of communities in each treatment: the labels (e.g. 1, 2, 3, etc) on the x-axis within each treatment denote communities (e.g. $C1$, $C2$, $C3$, etc).

We use the detected communities to show that subjects' play becomes more stable from round 6 onwards, motivating our choice to focus our analysis to rounds 6 to 13 only. For each round, we compare membership of corresponding communities in consecutive rounds to see if the communities stabilise over time. We focus on $C1$ and $C2$ only as together they make up the vast majority of the group in each round. We first define two stability metrics, and then use them to investigate when community membership becomes stable.

The first stability metric we define is *C1 Stable*: it captures the stability of the largest community ($C1$) in any given round by comparing the membership of $C1$ in round t to that of the $C1$ community in the $t-1$ and $t-2$ rounds respectively. If the $C1$ community in at least one round between $t-1$ and $t-2$ has at most two changes in membership compared to the $C1$ community at t, then we set the value of *C1 Stable* to 1 for round t, otherwise we set it to 0.

The second stability metric we define is *Any Stable*, which expands the definition of *C1 Stable* by extending the comparison to more communities. The *Any Stable* metric compares the membership of a given community ($C1$, $C2$, or $C3$) in round t to that of *any* other community in the $t-1$ and $t-2$ rounds. If the difference is

less than 2 for any comparison, then we set the value of *Any Stable* to 1 for round t, otherwise we set it to 0. If any community in at least one round between $t-1$ and $t-2$ has at most two changes in membership compared to the given community at t, then we set the value of *Any Stable* to 1 for round t, otherwise we set it to 0.

We compute the values of both metrics for every community in every round of the game, and aggregate it by computing the mean value over the $[13-x, 13]$ range of rounds for each treatment. We select the largest integer x such that the means of both the *C1 Stable* and the *Any Stable* metrics are above 0.5 with the additional constraint that $x < 9$ so that the history of the previous 5 actions is already available to subjects. Moreover, we require that if the metrics are above 0.5 in the $[13-x, 13]$ range of rounds, then they must be above 0.5 in the $[13-y, 13]$ range, where $y < x$. The largest x that satisfies these criteria is $x = 7$, and Table 4.2 shows the values of the *C1 Stable* and the *Any Stable* metrics for each community in all 4 treatments.

Treatment	**Community**	**Size**	$C1$ **Stable**	**Any Stable**
B	1	6.6	0.55	0.73
B	2	5.0		0.79
N	1	6.8	0.59	0.71
N	2	4.7		0.73
R	1	8.1	0.52	0.63
R	2	4.9		0.68
RN	1	8.0	0.5	0.57
RN	2	4.9		0.57

Table 4.2: Stability metrics for communities $C1$ and $C2$ in each treatment. Mean values of the *C1 Stable* and *Any Stable* metrics in rounds $[6, 13]$. Size is the mean size of the communities in each treatment. By definition the *C1 Stable* metric is only available for the $C1$ community.

Using the derived data based on the emergence and stabilisation of communities, our analysis at community-level reveals additional insights into the synergistic effects

of reputational and social knowledge on the evolution of cooperation. The difference between RN and the other treatments is almost exclusively driven by the highly cooperative largest community. Community $C1$ in RN is more cooperative than $C1$ in the B (rank sum, $P = 0.035$) and N (rank sum, $P = 0.025$) settings (see Figure 4.8a), and this higher level of cooperation is associated with a denser (Figure 4.8b) and more clustered (Figure 4.8c) $C1$ community than in the B and N treatments. In contrast, there is no difference across treatments in the level of cooperation or in the network structural features of the $C2$ community (Figure 4.8a-c). The synergistic effects of global reputational and network knowledge are key to allow the large community $C1$ of cooperators to differentiate itself: RN is the only treatment where $C1$ is more cooperative than $C2$ (Figure 4.8a; rank sum, $P = 0.025$).

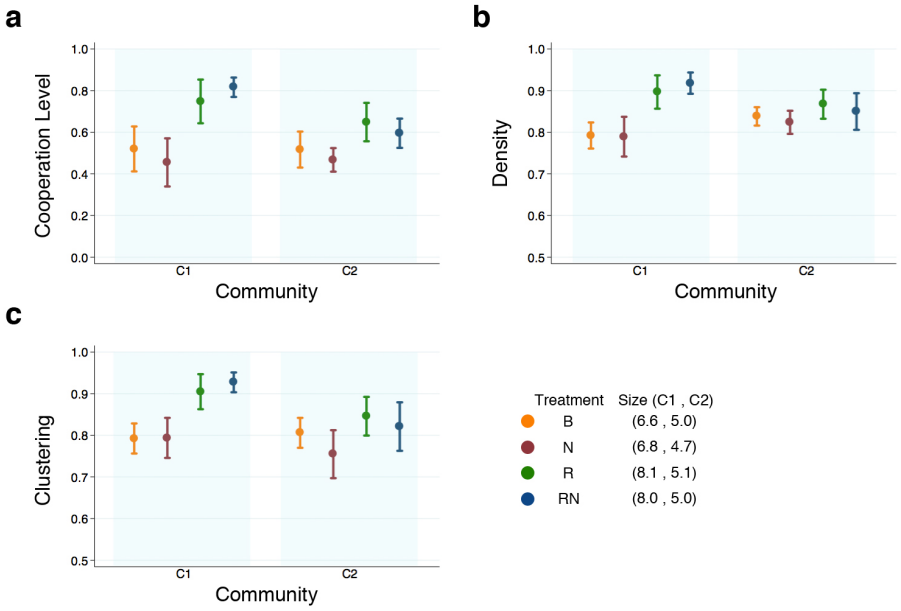


Figure 4.8: Cooperation level **(a)**, density **(b)**, and clustering **(c)** of the *largest* and the *second largest* communities for treatments B (yellow), N (red), R (green) and RN (blue). Analyses are based on aggregated data after round 5 in each session. Error bars indicate ± one SEM. These are the results for rank-sum tests comparing differences across and within treatments, we only include significant ($P < 0.05$) and close to significant ($P < 0.1$) findings. Between treatments for $C1$: **(a)** Cooperation level: RN vs. B, $P = 0.035$; RN vs. N, $P = 0.025$; R vs. B, $P = 0.085$; R vs. N, $P = 0.096$; **(b)** Density: RN vs. B, $P = 0.018$; RN vs. N, $P = 0.025$; R vs. B, $P = 0.048$; **(c)** Global clustering: RN vs. B, $P = 0.018$; RN vs. N, $P = 0.018$; R vs. B, $P = 0.048$; R vs. N, $P = 0.064$; Community size: RN vs. B, $P = 0.01$; RN vs. N, $P = 0.064$; R vs. B, $P = 0.064$. Between $C1$ and $C2$ within each treatment: **(a)** Cooperation level: RN, $P = 0.025$.

Members of $C1$ in RN differentiate themselves by actively excluding outsiders from their community (Figure 4.9): they remove more links than members of $C2$ (rank sum, $P = 0.018$), and, conversely, members of $C2$ have more links removed by others (rank sum, $P = 0.002$) and have more link proposals rejected (rank sum, $P = 0.018$). Global reputational knowledge alone does not suffice: the difference in cooperation in $C1$ between the R and B treatments is insignificant, $C1$ in R is only marginally denser and more clustered than $C1$ in B, and there is no difference in R between $C1$ and $C2$ in either cooperation level or link formation patterns.

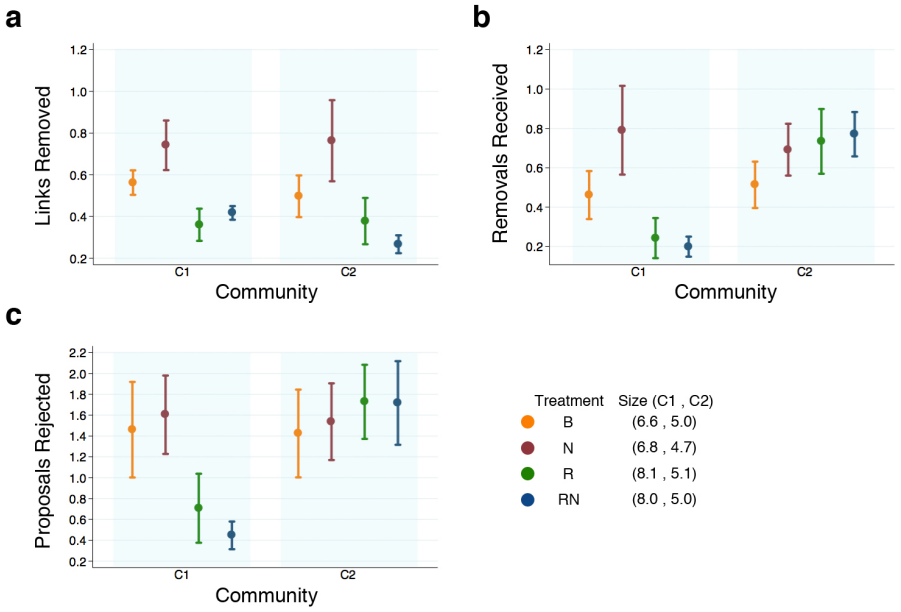


Figure 4.9: Links removed **(a)**, link removals received **(b)**, and proposals rejected **(c)** of the *largest* and the *second largest* communities for treatments B (yellow), N (red), R (green) and RN (blue). Analyses are based on aggregated data after round 5 in each session. Error bars indicate ± one SEM. These are the results for rank-sum tests comparing differences across and within treatments, we only include significant ($P < 0.05$) and close to significant ($P < 0.1$) findings. Between treatments for $C1$: Links removed: **(a)** RN vs. N, $P = 0.064$; R vs. N, $P = 0.048$; R vs. B, $P = 0.035$; **(b)** Removals received: R vs. N, $P = 0.064$; **(c)** Proposals rejected: RN vs. B, $P = 0.064$; RN vs. N, $P = 0.018$. Between treatments for $C2$: **(a)** Links removed: RN vs. B, $P = 0.048$; RN vs. N, $P = 0.064$. Between $C1$ and $C2$ within each treatment: **(a)** Links removed: RN, $P = 0.018$; **(b)** Removals received: RN, $P = 0.002$; R, $P = 0.096$; **(c)** Proposals rejected: RN, $P = 0.018$.

Thus global social knowledge alone is indistinguishable from the baseline B. Global reputational knowledge is the primary driver of the emergence of coopera-

tion, but global social knowledge enhances and complements the effect of reputation: it allows a large community of cooperators to thrive by clearly differentiating themselves and pruning out defectors.

Delving deeper into the data, we examine subject-level actions to show how reputational and social knowledge lead to an association between network position and cooperative behaviour. For each treatment, we perform panel logistic regression with random effects per subject and standard errors clustered at the session level. The dependent variable is the action taken by subjects (1 = cooperate), and we exclude observations (35 out of $5,018$ or 0.7% of the total) in which subjects take no action.

We include the following network metrics as independent variables to investigate the association between network structure and cooperation: *Degree* is the number of connections; *Clustering* is the local clustering coefficient [Watts and Strogatz, 1998]; *Betweenness* is the betweenness centrality index [Freeman, 1977]; *Eigenvector* is the eigenvector centrality index [Bonacich, 1972]; and *Community* equals to 1 if the individual is part of the largest community $C1$ and equals to 0 if the individual is part of the second largest community $C2$.

Payoff captures the number of points the individual wins in a round. The *L1.action*, *L2.action*, *L3.action*, *L4.action*, and *L5.action* variables are controls for lagged actions, so *Lx.action* denotes the action the individual took x rounds before. We also include a number of socio-economic characteristics (*Gender, Age, Education*), a measure of *Trust* from the standard question from the World Values Survey, and the results of a Holt-Laury risk elicitation test from the Questionnaire, see the caption of Table S1 for details on the coding of the variables.

We have a number of controls for the linking behaviour of subjects in a given round. *Links added* is the number of new links the subject adds in a given round; *Total links removed* is the number of links the subjects loses (either by removing others or by being removed by others); *Proposals received* is the number of link proposals

received by the subject; *Removals received* is the number of link removals received by the subject; *Proposals approved* is the number of a subject's link proposals approved by others; and *Proposals rejected* is the number of a subject's link proposals rejected by others.

Time 1, Time 2 and *Time 3* denote the time in seconds that subjects take to complete stage 1, 2 and 3 respectively in each round. *Mouse hover 1*, *Mouse hover 2* and *Mouse hover 3* capture the number of times a subject hovers over a node in the network figure in stages 1, 2 and 3 respectively. *Mouse drag 1*, *Mouse drag 2* and *Mouse drag 3* capture the number of times a subject clicks and drags nodes in the network figure in stages 1, 2 and 3 respectively. *Quiz tries* denotes the number of times a subjects has to take the Quiz before getting all the questions correct. *Quiz errors 1* and *Quiz errors 2* denotes the number of errors a subject makes when she takes the Quiz the first and second time respectively. *Session* denotes session fixed effects, which are included in all regressions.

As Table 4.3 shows, access to global reputational knowledge is necessary for the emergence of cooperative hubs: the number of connections (*degree*) is a highly significant correlate of cooperative actions only in the RN and R treatments ($P = 0.000$ for both). Defectors are poorly connected and/or they are the bridges (*betweenness* [Freeman, 1977]) between different communities ($P = 0.001$ for R and $P = 0.007$ for RN). The crucial additional role of global social knowledge is to enable cooperators to locate themselves in the highly cooperative $C1$ community. The *community* variable captures whether an individual is located in the $C1$ or the $C2$ community: membership in the $C1$ community is associated with a high level of cooperativeness only in the RN treatment ($P = 0.01$).

Table 4.3: Determinants of subjects' actions

Treatments	**B**	**N**	**R**	**RN**

Dep. Var.: Action								
Degree	1.226	(2.504)	2.832	(1.614)	13.38***	(2.485)	36.80***	(8.774)
Clustering	-0.202	(0.747)	1.205*	(0.514)	-2.080*	(0.907)	-0.171	(1.478)
Betweenness	4.672	(2.605)	1.550	(1.531)	-10.85***	(3.144)	-19.61**	(7.433)
Eigenvector	7.147	(4.451)	1.459	(1.787)	-3.636	(4.043)	-2.528	(11.48)
Community	0.550	(0.393)	0.464	(0.286)	0.658	(0.439)	2.018*	(0.797)
Payoff	-0.234***	(0.0300)	-0.126***	(0.0177)	-0.303***	(0.0352)	-0.950***	(0.177)
L1.action	0.609	(0.495)	0.134	(0.345)	1.509**	(0.575)	0.834	(0.945)
L2.action	0.922*	(0.428)	0.778**	(0.300)	0.371	(0.573)	2.327	(1.251)
L3.action	0.876*	(0.413)	0.607*	(0.299)	0.459	(0.505)	2.318*	(1.090)
L4.action	1.357***	(0.387)	0.847**	(0.299)	-0.263	(0.486)	-0.310	(0.767)
L5.action	0.642	(0.390)	0.324	(0.304)	-0.398	(0.513)	-1.726	(0.970)
Gender	-0.169	(0.376)	0.0345	(0.288)	-0.0692	(0.445)	-0.713	(0.831)
Age	-0.0130	(0.0185)	0.00189	(0.0166)	-0.0305	(0.0220)	0.0211	(0.0426)
Trust	0.155	(0.365)	-0.0176	(0.298)	0.0216	(0.505)	-0.721	(0.795)
Education	-0.0722	(0.142)	-0.0707	(0.106)	0.0559	(0.174)	-0.781*	(0.320)
Risk	-0.0246	(0.464)	0.0957	(0.268)	0.134	(0.435)	0.690	(0.844)
Links added	-1.645***	(0.290)	-1.166***	(0.210)	-0.512	(0.317)	-0.843	(0.704)
Total links removed	0.768***	(0.222)	0.132	(0.158)	0.912**	(0.333)	-0.242	(0.398)
Proposals received	1.416***	(0.241)	0.801***	(0.170)	0.909***	(0.225)	0.674	(0.636)
Removals received	-0.445	(0.287)	0.0348	(0.182)	-0.584	(0.354)	0.110	(0.426)
Proposals approved	1.264***	(0.348)	1.156***	(0.286)	0.138	(0.417)	-0.0206	(0.826)
Proposals rejected	0.000763	(0.103)	0.0207	(0.0755)	0.00624	(0.129)	-0.255	(0.242)
Time 1	-0.0389	(0.0240)	0.0135	(0.0167)	-0.0587	(0.0360)	-0.0705	(0.0789)
Time 2	-0.0305	(0.0283)	-0.00429	(0.0217)	-0.00961	(0.0455)	0.118	(0.101)

Time 3	-0.0600	(0.0377)	-0.0658*	(0.0270)	-0.0750	(0.0577)	-0.255*	(0.129)
Mouse hover 1	-		0.00990	(0.0221)	-		0.155	(0.244)
Mouse drag 1	-		0.191	(0.284)	-		0.809	(1.391)
Mouse hover 2	-		0.0808	(0.0727)	-		0.0233	(0.166)
Mouse drag 2	-		1.781	(1.661)	-		-0.995	(0.608)
Mouse hover 3	-		-0.0134	(0.0690)	-		-0.0272	(0.223)
Mouse drag 3	-		0.149	(0.256)	-		-0.607	(1.121)
Quiz tries	0.134	(0.672)	-0.548	(0.511)	0.763	(0.829)	-1.044	(1.521)
Quiz errors 1	-0.0795	(0.446)	0.207	(0.340)	-0.113	(0.628)	1.443	(0.894)
Quiz errors 2	-0.313	(0.580)	0.487	(0.617)	-2.256*	(1.140)	-2.136	(2.152)
Session	Y		Y		Y		Y	
Constant	1.788	(1.852)	-2.378*	(1.068)	-3.381	(2.029)	8.992*	(4.171)
Observations	658		722		802		756	
χ^2	132.7		188.9		114.0		39.66	

Standard errors in parentheses. * $P < 0.05$, ** $P < 0.01$, *** $P < 0.001$

Finally, Figure 4.10 visualises our findings using snapshots of representative sessions. Without global reputational knowledge, in the B and N treatments, the level of cooperation is low leading to sparse networks and no association between cooperation and position in the social structure. The introduction of global reputational knowledge in R leads to a sharp increase in cooperation level and the emergence of cooperative hubs: the network starts becoming denser and more clustered. The addition of global social knowledge in RN allows the cooperators to differentiate themselves from the rest of the group by forming a large, dense and clustered community which is characterized by cooperative hubs and the exclusion of defectors, and further increases the density and clustering level of the overall network.

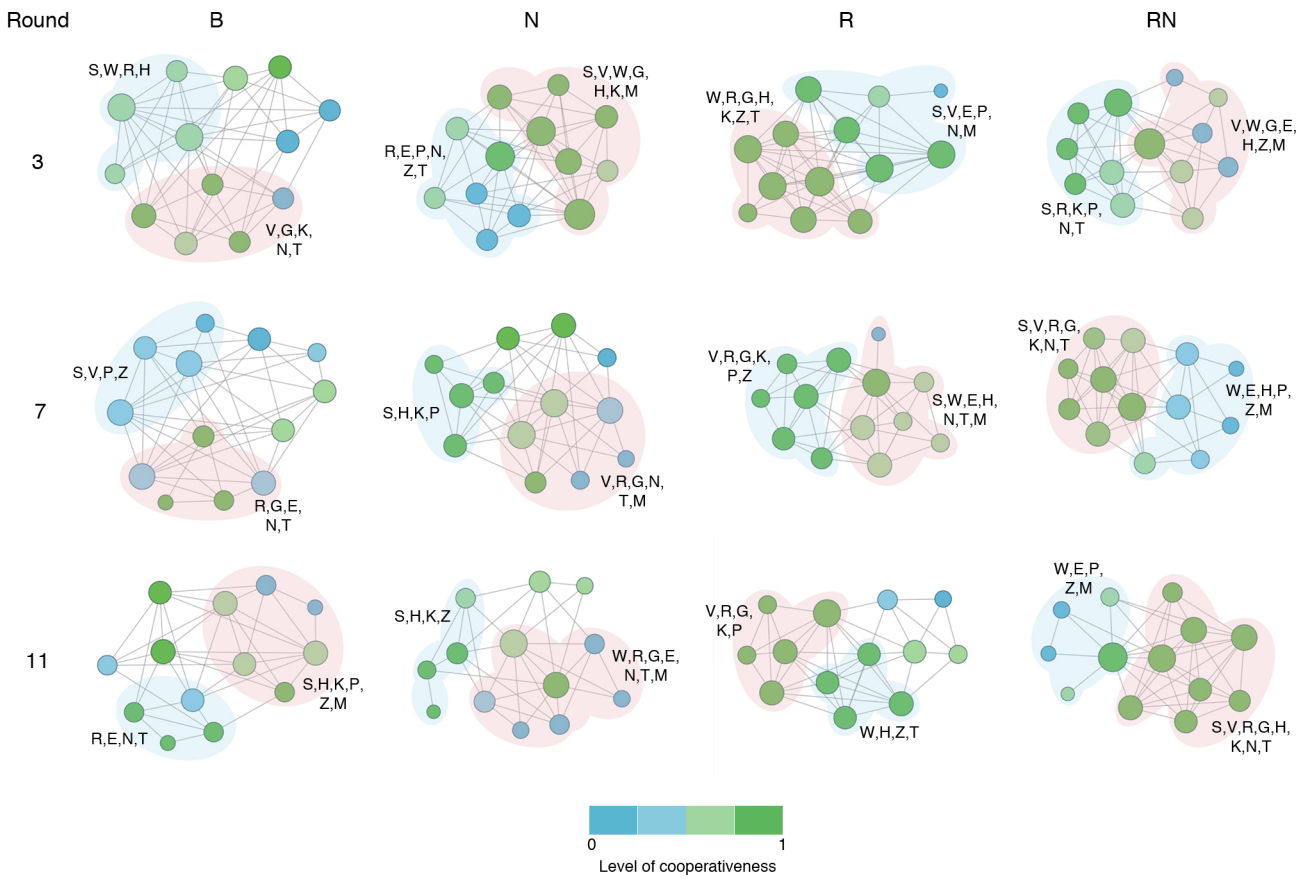


Figure 4.10: Snapshots showing the evolution of the network and cooperative behaviour for representative sessions. Each column is a representative session for each treatment, from left to right: B, N, R and RN. Each row is a round: 3, 7 and 11 from top to bottom. The size of a node is proportional to its degree. The colour of a node corresponds to the number of cooperative actions in the previous 5 rounds: blue (0 or 1 cooperative actions), light blue (2), light green (3), and green (4 or 5). Red bubbles distinguish $C1$ communities, while blue bubbles distinguish $C2$ communities. The labels next to the bubbles list the fictitious IDs of the members of the corresponding communities. Both the bubbles and the labels apply to the first two largest communities only.

4.4 Discussion

In this chapter we present an experimental investigation into the effects of different knowledge in dynamic cooperative networks. Unlike previous experimental studies on cooperation in endogenous networks [Rand et al., 2011, Fehl et al., 2011, Wang et al., 2012], we implement experiments in which subjects have the maximum degree of freedom to form and cut their links with each other in completely endogenous networks

and play the prisoner's dilemma games with their chosen connections repeatedly. We examine the co-evolution of cooperative behaviour and social structure given access to different degrees of reputational knowledge and social knowledge.

In our experimental design, the choice of symmetric payoffs (with negative payoffs) deters subjects from creating as many links as possible thanks to risk aversion effect [Holt and Laury, 2002]. In addition, our incentive structure of randomly drawing 12 pairings of interactions instead of summing over all available payoffs makes a link with a defector much more costly than one with a cooperator. Despite of having no restriction on the number of links cut and formed per round, subjects maintain links selectively rather than greedily based on available information, which allows us to observe different treatment effects while imposing as few unrealistic rules as possible.

We find that reputational knowledge is a prerequisite and the primary driver of cooperation. Treatments R and RN both benefit from the emergence of a large and highly cooperative hub that lead to a dense and clustered network. On the contrary, treatments without access to reputational knowledge suffer low level of cooperation, resulting poor social welfare (low network density with less cooperative actions). Having access to social knowledge alone makes little effect in promoting cooperation. A likely cause may be that the lack of reputational knowledge prevents identification of cooperators thus it becomes difficult for sizeable clusters of cooperators to form. Without large enough clusters, scattered cooperators are unable to establish the cooperative norm and defend against defectors, worse still, conditional cooperators are vulnerable to be turned to defectors due to shortage of cooperative interaction.

However, social knowledge does enhance cooperation when reputational knowledge is available. Social knowledge allows cooperators and conditional cooperators to locate the emerging cooperative hubs effectively, which facilitates the formation of a large, dense and highly clustered community of cooperators. Membership of this community may also amplify one's reputation of being cooperative, encouraging

more cooperative behaviour between members. On the other hand, being outside of or being poorly connected to this community may send a strong signal (in addition to reputation) about one's uncooperative behaviour, resulting link removals and/or link refusals. It would be interesting to see that how tolerant members of the cooperative community are to once-cooperative defectors[5], i.e. to what extent are they excluded after defection and for how long are they excluded?

In the past decade social networking tools (e.g. Facebook, LinkedIn) have become prevalent in our lives. These tools augment individuals' profiles with social knowledge by providing information about their social networks. We still have a limited theoretical and empirical understanding of the effects of having access to this additional social knowledge. The results of our study suggests that a major benefit of these tools is to facilitate the formation of communities in which members share behavioural commonalities and mutual benefits accrue. In the context of cooperation, we find that access to global social knowledge strengthens the well-known effect of reputational knowledge on cooperativeness through the emergence of a large community of cooperators that enhances the structural features of the network associated with cooperative behaviour. Exploring whether this type of effect of social knowledge extends to other domains is an important direction for future inquiries.

[5]Some defectors may strategically build up their reputation only to defect later to maximize their payoffs in a round.

Chapter 5

UbiquityLab: An Online Platform for Game Theoretic Experiments

Behavioural experiments is the method for empirical study of game theoretic models. Thanks to the growing capacity of the Internet and the rise of online labour markets, a handful of game theoretic experiments [Suri and Watts, 2011, Rand et al., 2011, Wang et al., 2012, Shirado et al., 2013] have offered a glimpse of the promising potentials of online experiments as a new methodology in experimental economics. However, a significant amount of time and efforts are duplicated by different researchers in developing and deploying common functionalities. Ad-hoc implementations also leave room for improvement and create difficulties for replicating results.

In this chapter, we present *UbiquityLab*, a novel platform that is purposely built for developing and conducting game theoretic experiments on the Internet. UbiquityLab is designed to cater for interactive real-time experiments, and aims to promote online experiments in general for empirical research of game theoretic models. The platform significantly simplifies experiment implementation and operation, while striving to meet research standards endorsed by experimental economists. So far we have used the platform to successfully conduct two complex experiments [Gallo and Yan, 2013b,

Gallo and Yan, 2013a][1] with a mean sample size of 312 participants and a mean completion time of 8 days.

5.1 Online Experiments

Researchers of different disciplines have been utilising the Internet since the 1990s. Some treat it as a test bed for new theories and tools [Leskovec et al., 2009], some take advantage of its ubiquitous coverage to carry out studies of unprecedented scale [Gracia-Lázaro et al., 2012, Bell et al., 2013], and some others employ the collective minds to solve hard problems creatively [Larson et al., 2002, Von Ahn and Dabbish, 2004]. The Internet itself is also a constant source of new research topics where researchers of almost every discipline find valuable problems to solve and intriguing phenomena to explain.

5.1.1 Introduction

Within the social sciences, psychologists pioneered online experiments using e-mail, live chat, and web survey. [Reips, 2002, Reips, 2000b, Reips, 2000a] discuss the advantages of online experiments from a psychologist's perspective and recommend some guidelines for conducting such types of experiments. The methodology is increasingly adopted by researchers from different fields of the social sciences. It is now common that anthropologists, linguists, political scientists and sociologists all operate experiments using the Internet. However, the vast majority of these experiments are survey based, which are taken individually with little or no interactive elements.

On the other hand, economists, and game theorists in particular, have relied on lab and field experiments to test their theoretic predictions in elaborate models. Comparing to other fields in economics, game theory often lacks readily available

[1] For screenshots of [Gallo and Yan, 2013b], see Appendix B

data for empirical research. When observing how people behave in strategic encounters depicted by theoretical models, game theorists often find themselves turning to behavioural experiments. In some cases, researchers may obtain insights from experimental results into situations that are yet too complicated to be elucidated by a theoretic model, and use them to help develop a theory. In the past two decades, computerised lab experiments have gained significant popularity among game theorists, as computer programs exert exact and detailed control over subject interactions and control for assumptions in the model.

Nevertheless, lab experiments suffers from several problems as the following.

1. **Slow turnover and iteration**: lab experiments usually take a long time to complete due to various logistical constraints. All too often, experiments are delayed due to difficulty of recruiting enough subjects and/or limited availability of lab facilities that are usually a shared resource among researchers. Such delays can be exacerbated when the experimental design needs to be revised after pilots.

2. **High cost**: in addition to the aforementioned time cost, lab experiments are financially expensive to run. Subject payment usually consists of two parts: a guaranteed fixed fee and a performance-based bonus. In most cases, payment can easily reach over $20 per subject. Furthermore, in order to retain subjects for future participation, most labs opt to pay the so-called "show-up" fee of around $5 to redundant subjects who are necessarily invited to every session.

3. **Limited subject pool**: subject pool of university labs are limited in both size and demographic diversity. Such pools are constructed using local population of which demographic profiles are restricted to university students and local residents, and most pools contain no more than a couple of thousands of subjects. Localised subjects also create difficulties for cross-border research where

researchers need to access a population sample of a specific country.

More recently, several Web-based game theoretic experiments have emerged [Suri and Watts, 2011,Rand et al., 2011,Suri and Watts, 2011,Mason and Watts, 2012,Rand et al., 2012,Wang et al., 2012,Rand et al., 2013]. The main difference in these studies compared to other online experiments is that they facilitate synchronous real-time interaction between subjects. Most of these experiments exploratively investigate various game theoretic problems, focusing on the theme of human cooperation. It is noted that related problems have been studied extensively in economics literature and that none of the aforementioned studies involves an economist. Furthermore, some of these studies adopt experimental procedures that deviate from the standard practices in experimental economics. By contrast, as we will show later in our methodological discussion, UbiquityLab is implemented to cater for the practices endorsed by economists wherever possible.

5.1.2 Methods and Tools for Online Experimentation

In the past five years, social computing and crowdsourcing have given rise to a number of online labour markets such as Amazon Mechanical Turk (AMT), oDesk, Elance, etc. The rapid growth of such markets offers an alternative venue for conducting experimental research. For example, using AMT, researchers (called *Requesters*) recruit human subjects (called *Turkers*) by posting temporary jobs (called *Human Intelligence Tasks* or HITs), operate experiments online, and pay the participants. So far most researchers have chosen AMT for online experimentation, and several methodology papers [Paolacci et al., 2010,Horton et al., 2011,Rand, 2011,Mason and Suri, 2012] have discussed its validity and quality as a research method. In particular, besides replicating classical results observed in lab, these studies also highlight several key advantages of AMT that address exactly the problems faced by lab experiments. More specifically, AMT offers an on-demand subject pool with an integrated payment

system, enables remote access to subjects of a targeted geographic location, and allows fast iteration with low monetary cost. In addition, [Ipeirotis, 2010a, Ross et al., 2010, Buhrmester et al., 2011, Berinsky et al., 2012] show that this pool is more demographically diverse than the standard Internet samples, significantly more diverse than American college samples, and so far is *currently* more representative of the U.S population[2]. Some may argue that Turkers' low wage provides inadequate economic incentives, but this concern is mostly mitigated by [Amir et al., 2012].

Despite potential advantages of using AMT as a source of subjects, it is important to note that much care needs to be taken with the Turkers to ensure the validity of the results and that AMT is certainly not the only source of online subjects. We can identify three potential problems with AMT subjects:

1. There may exist a selection bias among the subjects from AMT. While one could argue the same for lab subjects who are mostly students, Turkers are more susceptible to be an atypical sample of the general population. Indeed, many Turkers claim that they are either unemployed or underemployed, and intend to use AMT as a financial crutch to maintain their income. In addition, the fact that Turkers adopt the crowdsourcing concept of AMT may indicate that they are less likely to be a conservative bunch. If the nature of a study involves a particular societal group (e.g. conservative males in political science experiments), then researchers should go to great lengths to verify sample validity before drawing any conclusion.

2. Because Turkers participate online *and* their primary objective is to earn money by completing HITs, there may exist a considerable incentive for them to misreport various information in order to participate in experiments. For example, many studies on MTurk restrict participants to a particular country in order

[2]As AMT is still young and growing, its use demographics evolve constantly. In May 2012, [Economist, 2012] stated that roughly 40% of workers are U.S based, 30% are from India, and rest come from more than 100 other countries.

to conduct research on the specific group of residents. Unfortunately, some Turkers may lie about their residential location in order to complete the HIT. Researchers need to take extra precaution to ensure that the essential requirements of participation are met, and in this case, geo-mapping subjects' IP may become an effective method to eliminate most unqualified subjects.

3. Possibly the most obvious and direct cause of problems for researchers using AMT is prior exposure to similar studies or even repeated exposure to the same studies. As more and more researchers conduct experiments using the shared AMT sample, it is almost inevitable that some experiments will be very similar (or perceived as similar) to each other. If a subset of Turkers participate in multiple such experiments, then the data produced in later experiments may be contaminated by their learning in prior experiments. This is particularly bad for any experiment that involves measurement of participants' performance in certain cognitive tasks, since their performance can vary significantly due to prior learning/training. This problem may be exacerbated if "crosstalk" happens between subjects such that the details of the experiments are discussed prior to participation. Researcher should closely monitor all possible communication channels online to detect and prevent such misbehaviour. Furthermore, researchers should take all possible measures to prevent repeated participation in the same experiment, as such practice is methodologically flawed.

As more researchers recognise and utilise MTurk, various software tools are created to assist different types of research, several of which are summarised below.

TurKit [Little et al., 2009] is a library for creating human computation tasks on AMT. It provides a Java and JavaScript API that allows programmatic iteration of tasks such that current tasks can utilise the results of earlier tasks. It is most suitable for HITs that are completed by Turkers independently.

MTurk Tracker [Ipeirotis, 2010b] is a web service which crawls the AMT website to obtain various HIT-related statistics (e.g. numbers of fresh and completed HITs, classifications, top requesters, etc) and visualize them with charts. It gives a quick overview of the past and the current activities on AMT.

Turkalytics [Heymann and Garcia-Molina, 2011] is an analytic software for tracking and analysing worker demographics and worker-related statistics. Researchers may use such tool to sort/group workers for demographic-specific experiments which are of great interest among social scientists.

TurkServer [Mao et al., 2012] is a Java-based platform built on top of MTurk API for running synchronous and longitudinal experiments. It is designed for research in human computation. It may seem at first sight that TurkServer is similar to UbiquityLab, but many details differs the two as we shall discuss in the next section.

Qualtrics [Qualtrics, 2013] is a web survey and data collection service. It is intended for research and is subscribed by numerous universities as well as corporate organisations. It is probably the most popular choice for survey-based experiments and is a common tool used in most of academic HITs.

PsiTurk [Coenen et al., 2013] is a platform for running psychological experiments. Using JavaScript, PsiTurk enables conditional execution during experiments given inputs and offers a small subset of functionalities in Qualtrics that are adapted for psychological studies. It only allows individual-based experiments so that no communication or interaction is possible among participants.

5.2 Functionalities and Usage

UbiquityLab is designed for game theoretic experimentation by economists. To achieve this aim, we design and implement the platform with careful considerations.

At the high level, the platform supports both synchronous and asynchronous game theoretic models and allows for complex interaction. In the most common synchronous mode, a group of participants play a n-stage ($n \geq 1$) base game for a number of rounds. In each stage, participants submit their actions before receiving feedback and proceeding. This mode of interaction corresponds to simultaneous moves by agents at predefined intervals (e.g. a series of discrete timesteps) in theoretical models, and is the most common setting in lab experiments. It is suitable for studying models of repeated interactions[3], such as repeated prisoner's dilemma games, ultimatum or dictator games, public goods games, etc. By contrast, asynchronous mode does not enforce everyone to act at the same time. Participants can act at any time during a specified interval, and the frequency of actions may be configured. This mode usually features a shared history of actions based on which participants take further decisions. Typical examples include auctions and market trading games.

5.2.1 Main Functionalities

Besides the high-level support for two different types of experiments, UbiquityLab offers the following core functionalities.

Experiment creation: Experimenter only needs to implement game-specific logics on the server-side and user interface on the client side using our specified messaging protocol. The server-side development is minimised to thin Python modules that customise different aspects of the game, while the client-side development is helped by a suite of common functions and templates. Experimenters can configure each experiment to a great extent with reduced effort. For example, experimenter can

[3]One can turn any experiment into a one-shot game by setting the number of rounds to 1.

adjust number of players, game length, game initialisation, action processing, data recording, among other configurations for every experiment. Once the experiment is developed, an HTTP API is available for managing experimental sessions. Internally, UbiquityLab has a simple web interface for session management.

Experiment execution: The platform is capable of running *multiple* experimental sessions in parallel. These simultaneous sessions can be either of the same experiment or of different experiments. The platform uses separate servers to handle synchronous and asynchronous experiments in order to maintain clean server logics. Moreover, it comes with a complete and configurable workflow that includes the following steps: waiting room, instructions, interface tour, Q&A chat, comprehension quiz, experimental game, final questionnaire, and payment confirmation. All the steps are optional except for instructions, experimental game and payment confirmation. Data is recorded from the moment the participants start reading the instructions to the end of the final questionnaire. Game-specific data collection allows customisation over both data format and data content. Finally, all experimental data is stored in `.csv` format and is retrievable as `.zip` files.

User management: UbiquityLab hosts two types of users: experimenters and subjects. For experimenters, UbiquityLab is capable of providing account-based accesses such that experimenters can manage and run experiments independently. For subjects, UbiquityLab records their IDs (e.g. those given by AMT), IP addresses, and received payment for each participation. The purpose of collecting these records is two-fold: 1. It helps prevent repeated participation in the same or related studies as well as multiple or colluded participation within the same session; 2. It builds a dedicated subject pool of the platform for the long term. At the moment, demographic information are obtained using questionnaires hosted by Qualtrics and are currently not stored by the platform. However, the platform plans to migrate these data and directly utilise such information once it becomes independent from AMT.

5.2.2 Differentiating Features

In addition to the core functionalities above, UbiquityLab possesses several differentiating features.

Best practices: It caters for economists who are particularly demanding over details of experiment implementation. The platform takes extra care to ensure that it meets the common standards in the field. For example, by default no subject can participate in more than one session of a given study (even if subjects only see the instructions and then abandon participation). It can also be configured to prevent subjects from participating more than once in similar or related studies. This is a standard practice in experimental economics since repeated participation undermines data integrity and makes it difficult for experimenters to control for undesired effects. By contrast, earlier online experiments [Suri and Watts, 2011,Wang et al., 2012,Mason and Watts, 2012] repeatedly use the same subjects in different sessions of the same treatments[4]. The platform also checks IP addresses within each session as well as across sessions so that few can "cheat" by pretending to be different participants. This is necessary in online experiments as experimenter cannot fully verify every single participation. It also helps prevent colluding participants who participate at the same location[5]. Furthermore, UbiquityLab enforces a seamless workflow to ensure that: 1. Subjects spend some minimum amount of time on each page of the instructions; 2. Subjects cannot skip any step even if they know all the URLs of different parts of the experiment; 3. Subjects need to pass the quiz in order to participate in the actual experimental game; 4. Subjects have no access to the final payment page without completing every steps of the experiment.

Flexibility: It abstracts away many intricacies of server-client interaction. By

[4]Or in sessions of different treatments without explicitly and fully implementing a within-subject design.

[5]Indeed, during our initial pilots, a subject complained to us that she could not participate because one of her housemates was participating in the same experiment using a shared Internet connection.

adopting a modular design, the platform leaves the implementation of game-specific logics and user interface to experimenters, which allows the maximum degree of customisation. For example, using minimal server-side scripts, experimenters can configure game initialisation, player input processing, data collection, payment calculation, program player behaviour, etc. A set of these scripts are plugged into the server using an API such that an experiment becomes available once the scripts are uploaded. On the client side, experimenters are free to construct any user interface that messages with the server via a specified protocol. To reduce the cost of user interface implementation, the platform provides some common components such as countdown timer, inactivity detection, layout templates. Experimenters obtain highly customised experiments with reduced effort.

Subject experience: It strives to provide the best participation experience. This is not only crucial to the success of any online experiment but also influential in terms of drawing interest for future participation. For example, the platform offers chat room functionality so that participants can ask questions before the game. In the chat room, all responses by experimenter are public by default, however, experimenter can choose to communicate privately with individual participants and can filter out any inappropriate messages. During the game, the platform actively prevents dropouts caused by temporary network problems, and any temporarily disrupted connection is automatically restored upon detection. The aforementioned program player feature ensures that every session finishes even if all but one participant drop out due to problems beyond experimenter's control (e.g. power or connectivity failure, software crash, voluntary termination). Experimenter is able to configure whether the remaining participants are notified about dropouts. In any case, as long as the game starts, every active participant will complete the experiment without worrying about wasting any time or effort due to others' problems.

Technologies: It utilises a set of current yet proven technologies. Experiments

run entirely within browsers with default settings and without any extra plug-ins. The application servers are based on the non-blocking *Tornado* [Facebook Inc, 2009] server that is designed for real-time and scalable Internet systems. The data storage utilised *Redis* [Sanfilippo and Noordhuis, 2009] server to take advantage of its robust performance as well as its schema-less feature considering that different experiments may require different data formats. The client-server messaging uses *SockJS* [Majkowski, 2011] library to adopt WebSocket [Fette, 2011] communication in modern browsers with backward compatibilities for older browsers. In order to minimise dropouts due to networking errors, its messaging protocol is written with a "heartbeat + caching" mechanism so that connections can be restored automatically upon detection of disconnection within a configurable grace period. It also supports "program players" which automatically substitute disconnected human subjects. In comparison, technologies such as Java Applet and Comet, as adopted by TurkServer, are more error prone and less scalable among users. Development in Java is also more verbose than in Python, resulting in higher costs in creating experiments and extending features.

Independence: It is not tied to any particular source of subjects. Many online labour markets are still developing and there are other potential channels for recruiting subjects, thus it is advantageous for the platform to be able to work with subjects from any source. For example, Facebook is a promising possibility with the help of its universal login feature. Being independent is also necessary as the platform aims to operate its own subject pool. By comparison, TurkServer is closely integrated with AMT, and its usability and workflow is dictated by that market. A major limitation imposed by AMT is that only U.S-based entities (with U.S. address and bank account) can register as requesters, creating a barrier for non-U.S researchers.

5.2.3 Recommended Workflow and Usage

We present an optimised and tested workflow for running multiplayer online experiments to make a case for the methodological contribution of UbiquityLab. Although the current workflow derives from our experience of working with AMT, it is applicable to other sources of subjects in general.

Figure 5.1 gives an overview of the process of a typical experiment. The blocks coloured in green represent components of the platform, whereas the blocks in blue represent external processes. Dashed lines indicate optional routes that can be configured by experimenter. We suggest completing the steps prior to the waiting room a couple of weeks before the first session. All the steps starting from the waiting room till the end constitute a session and most of them do not require manual intervention. For best results, experimenter should monitor the progress of live sessions and act accordingly. We discuss details of this workflow as below.

First, most experiments should start with a qualification task. The type of experiments running on UbiquityLab is usually more cognitively demanding, and hence requires more commitment and discipline from subjects.

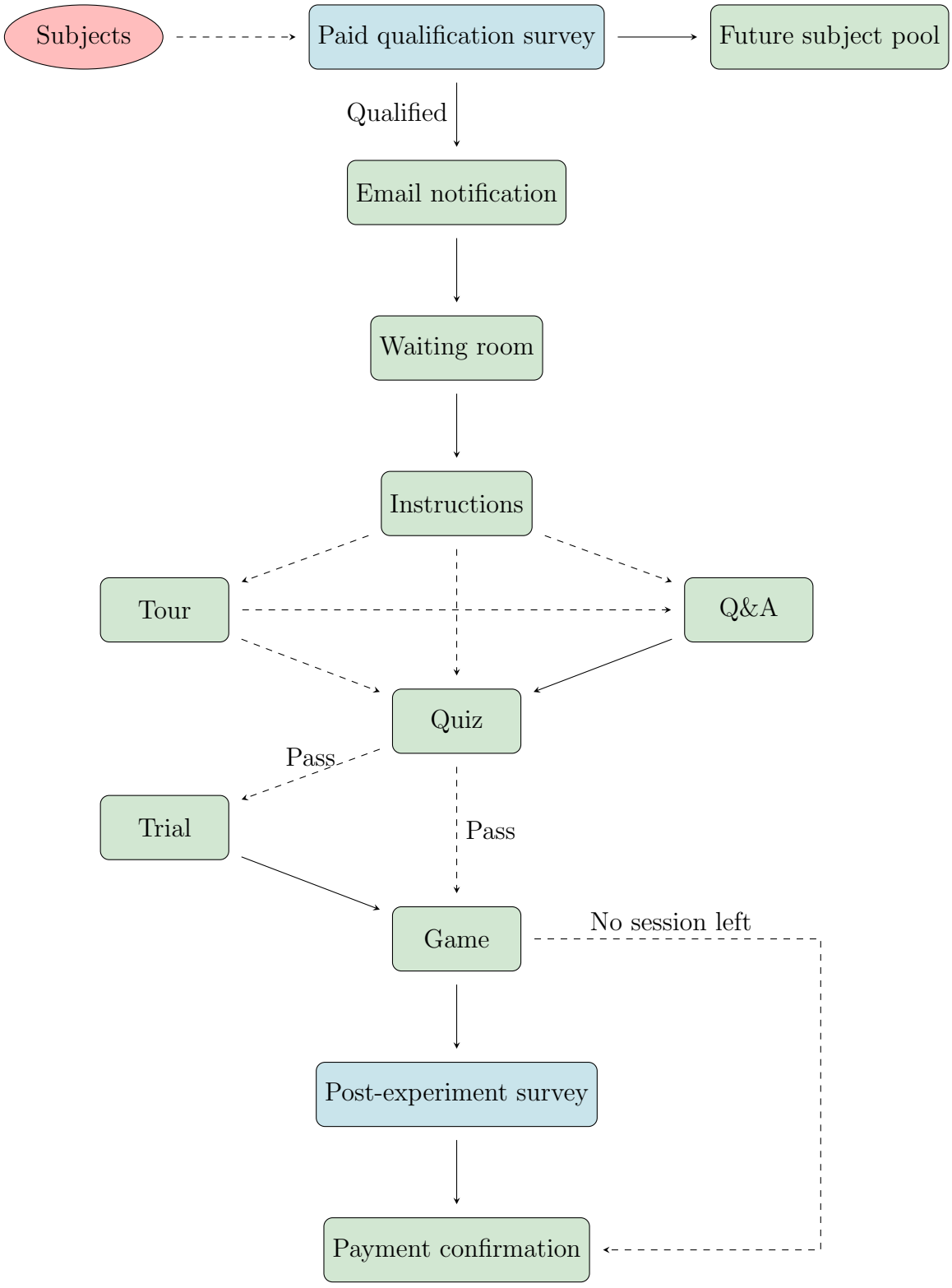


Figure 5.1: An optimised workflow of conducting online experiments using Ubiquity-Lab

In particular, more care need to be taken if subjects are recruited from AMT since most Turkers are accustomed to short (<15 minutes) and repetitive tasks. Indeed, a large number of HITs on AMT are simplistic and mechanical, and many Turkers primarily search for such HITs to maximise their rate of earning. Considering that Turkers are free to abandon tasks, it is hardly ideal to let such workers take part in experiments involving a fixed number of participants interacting with each other in real-time. Furthermore, some experiments require basic understanding of certain knowledge, thus it is necessary to ensure that subjects are qualified to undertake such experiments. We recommend using short survey-based HITs (e.g. using Qualtrics) for qualification tasks. The survey should contain questions that test specific knowledge or attributes needed for the actual experiment. For instance, in an experiment where the payoff function involves quadratic polynomials, we may include basic polynomial arithmetic calculation, and only those workers who answer them correctly within 3 tries qualify for later participation.

In addition to sourcing competent subjects, running qualification enables communication between experimenter and subjects, as AMT requesters can contact a worker through email only if the worker has completed a task for the requesters. Experimenter can then notify subjects about the time and the date of experimental sessions in advance, which is necessary for any experiment that requires simultaneous participation by multiple subjects. Furthermore, running qualification tasks is a viable way of building up subject pools. Even if a subject does not qualify for a particular experiment, the person may still be eligible to participate in others. Lastly but not least, paid qualification tasks accumulate reputation as a requester in the market, facilitating future research HITs.

Second, in communication with subjects, experimenter should be clear about the range of expected earning and the length of a session and these information should be repeated in the description of the experiment HIT. The earning usually consists

of a fixed fee p and a performance-based bonus i for incentivized game playing. We have found that $\frac{p}{i} \in [1, 3]$ offers a good calibration.

Furthermore, we recommend sending notifications to at least 3-4 times as many subject as required for each meta-session[6]. The number of required subjects per experimental session can be calibrated at $n = m + b$, where m is the number of subjects for starting the experimental game and b is the number of backups per session. Backups are necessary since it is not guaranteed that every subject will finish the instructions and pass the quiz, and we suggest setting $b \in [\frac{m}{4}, \frac{m}{3}]$. By default, the platform admits n subjects at a time only when there are $\geq n$ subjects active on the waiting room page.

One may ask what happens to the redundant subjects as a side-effect of this backup mechanism. UbiquityLab offers two modes of admission plus backup: In the "strict" mode, for every admitted group of n subjects, only the first m subjects of the group who reach the game page will participate in the game while the rest will be re-directed to receive the fixed fee p should they pass the quiz. This mode is suitable when experimenter prefer to maintain a balanced collection of fast and slow participants in a session and do not need a large number of subjects in a study. In the "relaxed" mode, as soon as $\geq n$ subjects arrive on the waiting room page, admission opens and remains open until all the experimental sessions are fulfilled or the waiting room is shut by the experimenter (whichever comes earlier). In addition, after passing the quiz, subjects queue on the game page and a game will start once there are enough subjects present. In this mode everyone will participate in a session unless either all the sessions are exhausted or there is not enough subjects remaining to start a session. The "relax" mode is suitable when the study requires a large number of subjects or the variance of time for completing the instructions and the quiz is high. In both modes, experimenter most likely will end up with redundant

[6]Each meta-session comprises multiple actual experimental sessions that are run in parallel.

subjects and more likely so when using the "strict" mode. As a standard practice in experimental economics, any redundant subject is disallowed to participate in future sessions of the same study after seeing the instructions.

Third, experimenter should monitor live session progress once subjects arrive at the waiting room page. The platform currently uses detailed loggings as a solution. Based on session progress, experimenter may choose to perform two actions. First, experimenter may shut down the waiting room to avoid the waste of late subjects in a meta-session with low turnout. This also prevents other subjects from waiting indefinitely and minimise their frustration. Upon shutdown, all the subjects on the waiting room page will be re-directed to a separate page where they will learn about the situation and will receive a small incentive (e.g. an extra \$0.30) if they complete a session next time. Second, experimenter may decide to terminate all the remaining games when there is not enough subjects left to start a game. This is especially useful when using the "relaxed" mode. After termination, the platform will redirect all the currently waiting subjects to the payment page to receive the fixed fee.

Prior efforts such as [Suri and Watts, 2011, Wang et al., 2012, Mason and Watts, 2012] try to justify repeated participation due to the lack of coordination and predictability of online subjects. Given our experiences, we believe that the combination of qualification, admission and backup system, live monitoring and control overcomes these problems to maintain sound practices.

Fourth, experimenter should collect and utilise different aspects of subject behavioural data throughout a session. For example, the platform logs how long a subject completes the instructions, how many mistakes a subject makes in each try of the quiz, and how long a subject waits for the start of the game. During the game in each stage, the platform logs whether a subject makes an input and how long it takes for making one. If additional interactive features are available, user input such as mouse clicks and mouse movement are recorded too. In addition, ex-

perimenter should always configure program player to prepare for the substitution of dropped out human players. The program player supports two types of game-playing behaviour. It can remain active and visible to other participants and behave according to some predefined decision rule (e.g. playing according to a Nash equilibrium). Alternatively, it can become passive and invisible after notifying other participants about the disconnection(s). The former option continues the game as if no participant drops out though the affected session should be abandoned and reported as such. The latter option may allow the session data still be used depending on the nature and the design of the experiment.

After the game, the platform redirects the participants to Qualtrics to fill in a questionnaire where experimenter collect demographic profile and auxiliary information (e.g. risk and trust attitudes). Participants are then redirected back to the platform to review a detailed summary of their earnings and are provided a unique confirmation code for claiming their payment through AMT. Thanks to the functionalities by Qualtrics, both redirections are automated and configurable, contributing to a smooth and unified user experience.

5.3 Technical Implementation

In this section we examine technical aspects of UbiquityLab. We will first show the architecture of the platform and focus on a few design decisions. We will then demonstrate the implementation of the most common type of experiments, synchronous games, by presenting its messaging protocol and its main server-side logic. Finally, we will showcase the modular API for customising synchronous games.

5.3.1 Platform Architecture

The platform consists of client and server components. Figure 5.2 gives a simplified structural overview.

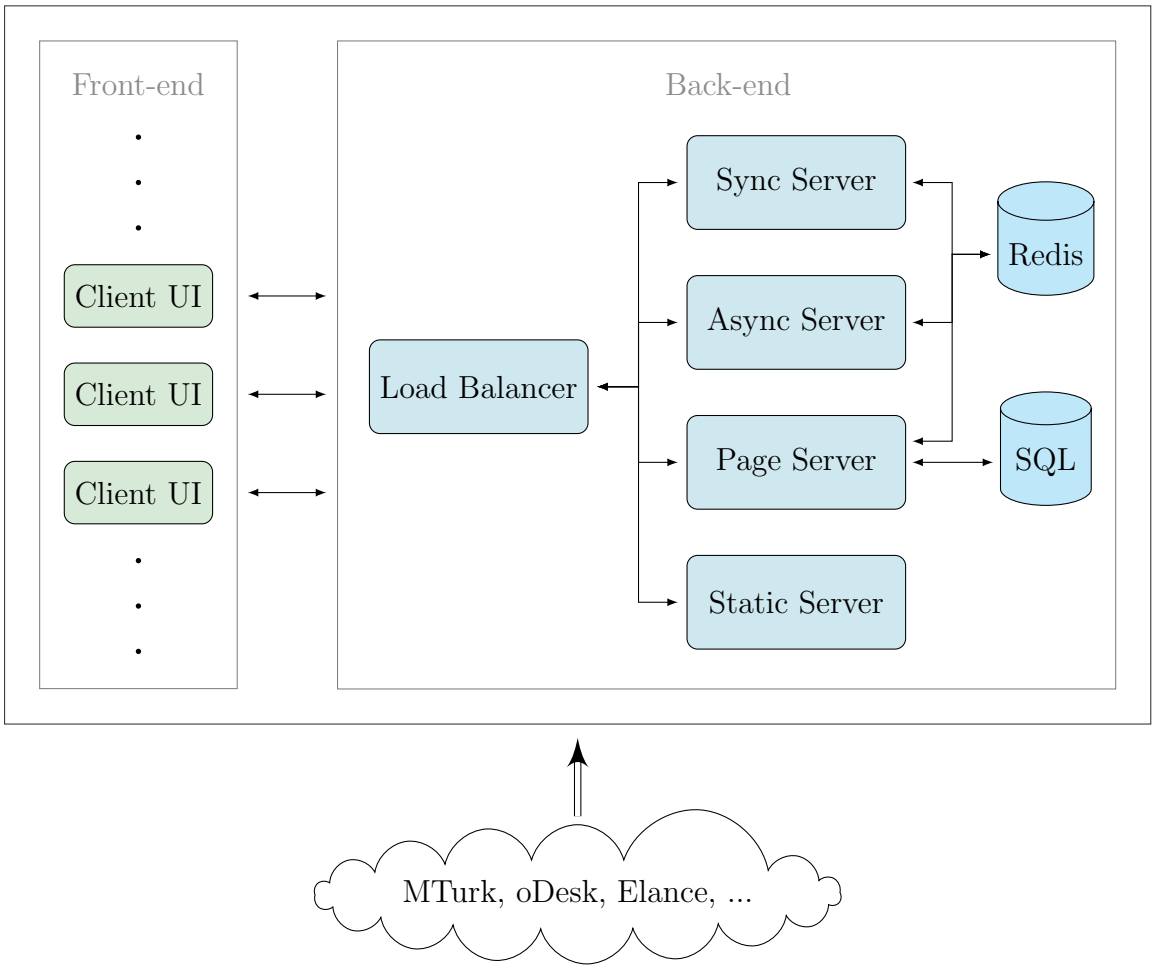


Figure 5.2: A simplified overview of the architecture of UbiquityLab

On the server side, *HAProxy* [Tarreau, 2001] load balancer regulates all inbound and outbound traffic by offloading SSL transport and reverse-proxy-ing for upstream servers. Functionally speaking, the load balancer handles two types of connections: short-lived HTTPs connections and long-lived SockJS connections. Short-lived connections deliver static web pages, while long-lived connections cater for persistent communication required by the waiting room and the games. SockJS is a JavaScript library that emulates WebSocket's bi-directional communication across browsers. It

uses fully duplex WebSocket transport by default whenever possible, and downgrades automatically to HTTP-based transports such as xdr-streaming or xhr-pooling if WebSocket fails. All the connections to UbiquityLab are encrypted to ensure the maximum support for SockJS in case that the client is behind an erroneous proxy.

Two application servers are created to handle synchronous and asynchronous experiments respectively, and they use JSON messages to communicate with the clients. Both servers are programmed using *Tornado*, which is an open source framework for building real-time server systems. The framework adopts non-blocking network I/O (e.g. `epoll` on Linux) so that it can maintain a large number of long-lived connections with much lower memory footprint than traditional multi-threaded servers[7]. The two game servers exclusively deal with SockJS connections for real-time interaction. A third application server (i.e. "page server") is written to manage all other aspects of experiments, such as experimental setup, workflow management, data extraction, etc. Notice that experimenters interact with this server only. Furthermore, a *nginx* [Sysoev, 2002] server dedicates to serve all static contents (i.e. HTML, CSS, images and JavaScript files) as almost all the pages of an experiment are created in advance rather than dynamically generated[8], and nginx is especially efficient in delivering static files.

For data storage, both game servers use *Redis* [Sanfilippo and Noordhuis, 2009] to record experiment data. Redis is an in-memory key-value store with persistence options. It accommodates a range of data structures such as hashes, lists, (sorted or unsorted) sets, etc. It is extremely fast and robust, and is schema-less as data are stored as strings. This strong performance suits the real-time nature of the platform, and is compatible with the use of Tornado servers. The freedom of storing strings instead of schematic data offers flexibility as different experiments may produce very

[7]Being single-threaded, Tornado servers require optimised implementation to block the I/O loop as short as possible to maximize scalability.

[8]This is also true for the game pages on which interactivity comes from JavaScript runtime and its communication with a game server.

different datasets. Both game servers and page server also utilise Redis to share states and track session progress. Furthermore, Redis enables communication from page server to game servers using its support for "Publish/Subscribe" messaging model so that game servers receive configuration and control from page server. On the other hand, a *SQLite* [Hipp, 2000] database serves exclusively page server to stores all other data such as experiment settings, session details, subject details, etc. SQLite is a lightweight yet effective choice, considering that most intensive read/write operations are handled by Redis.

As a whole system, a load balancer, 4 servers and 2 data stores can run either on a single multi-core server or distribute across multiple servers for scalability. So far, we have been operating the platform with ease using a single Linux virtual server that comes with 1GB memory and a shared 8-core processor. The total system-wide memory usage is consistently below 140 Megabytes and the CPU usage peaks around only 5% when running 6 parallel sessions of more than 100 participants. These benchmarks suggest the potential of running large-scale experiments involving thousands of participants.

On the client side, the platform enforces a JSON-based messaging protocol for communication between the game interface and a game server. Experimenter is free to implement most client-side functionalities specific to individual experiments.

5.3.2 Messaging Protocol and Server-side Logic

Synchronous games are the most common type of the experiments in experimental economics. It follows a round-based structure where a group of participants make decisions at fixed intervals. It is sometimes the preferred mode of experimentation as every decision maker faces the same time constraint and makes the same number of decisions. In some cases, researchers may even convert a naturally asynchronous activity into a synchronous game for the ease of experimental investigation.

There are three main building blocks of the implementation of synchronous games: *PlayGraph*, *ActionCache* and messaging protocol. Furthermore, a modular API is used to customise individual experiments on top of these infrastructural components.

PlayGraph is a graph data structure that defines the interactive relationships between participants in a game. In other words, each PlayGraph instance describes who plays against whom in a round of a game session. By default the relationships are mutual, i.e. the underlying graph is undirected[9]. It provides a number of helper functions such as retrieving connections for a given player, adding and removing players, connecting and disconnecting a pair of players, random shuffling of players, etc. In the simplest case, a PlayGraph instance can represent pairwise interactions with or without random re-matching, corresponding to partners or strangers condition in experimental design. On the other hand, it can model both static and dynamic connections in network-related experiments, that is, a network can be either exogenously set by experimenters [Gallo and Yan, 2013b] or endogenously formed by subjects [Gallo and Yan, 2013a]. Its implementation utilises a *Numpy* array to seek efficiency and functionalities. We include the full implementation of PlayGraph in Appendix C.1.

ActionCache is an in-memory storage of the complete history of actions for all the active game sessions. Player actions constantly update this centralised storage, and the game server erases session-specific history upon completion of the game. The game server also accesses the storage when it processes a round of actions. Efficient retrieval of these data is thus crucial to game server performance, and is implemented using an indexed and nested dictionary object. In a nutshell, we index all stored actions by player, stage and round respectively so that actions can be efficiently retrieved based on a given player, stage or round using the corresponding index inside a generator expression. We include the full implementation of ActionCache in Appendix C.2.

[9]It also can be configured to support unilateral relationships by using a directed graph.

A robust messaging protocol between server and clients is designed and implemented to accommodate different synchronous games, and all client-side implementation need to adhere to this protocol in order to communicate with server. Server acts on input at various stages of a game when it receives the expected messages by all the clients in the same session. In addition, a simple yet effective "heartbeat" mechanism is in place to detect potential networking issues. For example, an intermediary router may malfunction or a client may drop due to hardware failures without immediately breaking the logical connection to the server. The heartbeat mechanism makes server listen to a one-character heartbeat message sent by client every 5 seconds. If server misses three consecutive heartbeats for a client, it breaks the connection to activate re-connection procedure embedded on the client-side. Client then has up to three chances to reconnect automatically while server waits up to 20 seconds for reconnection before replacing client with a program player. If successful reconnected, client re-sends "Register" message and the cached last-sent message so that server can identify and restore its game states. Similarly, server sends its own cached last-sent message after client identifies itself. In both ends, server or client compares received message with the cached last-received message to avoid possible duplicates. Figure 5.3 illustrates the protocol with the omission of heartbeat messages.

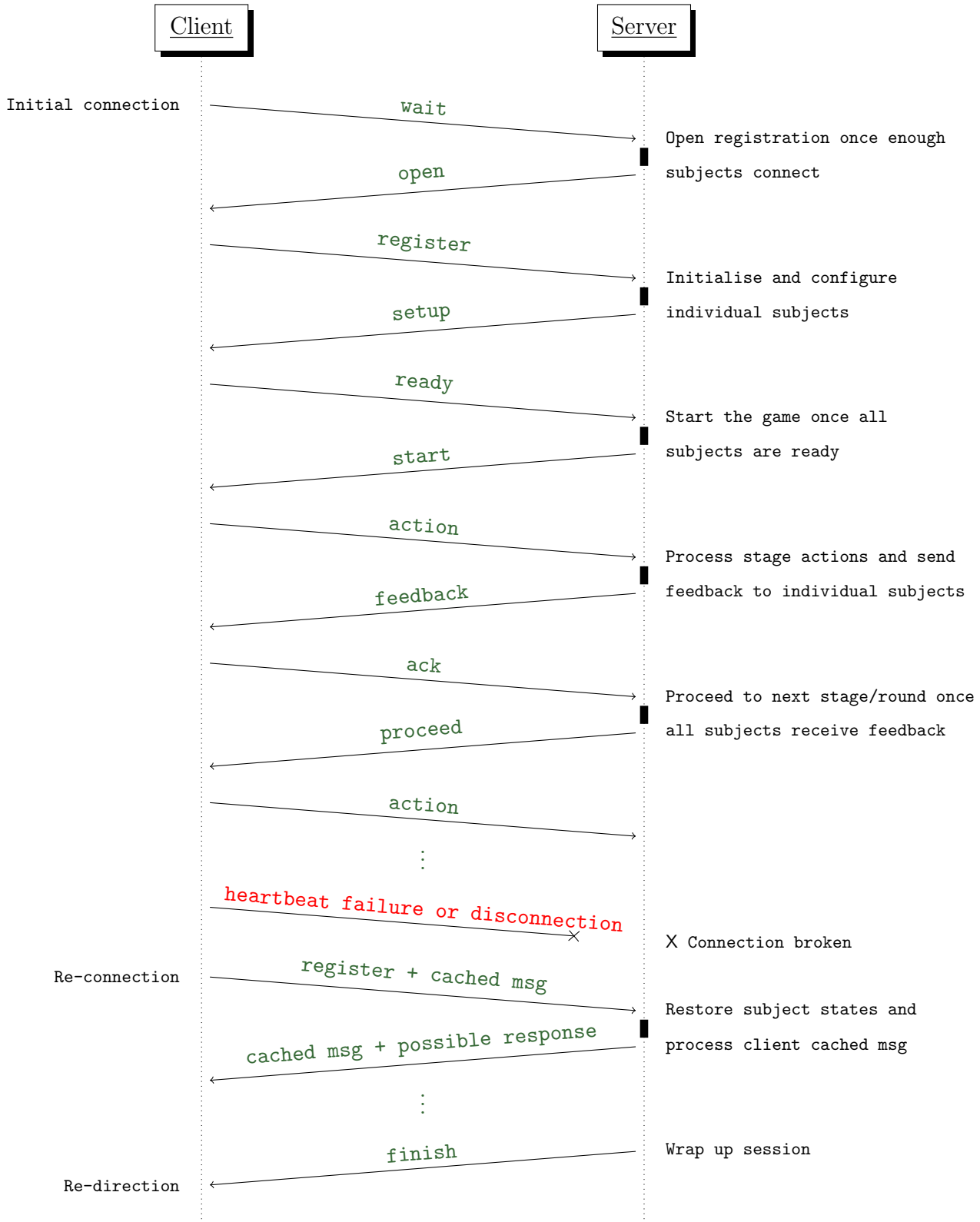


Figure 5.3: Client-server messaging protocol for synchronous games

Based on the messaging protocol, the following pseudo-code shows the core logic of synchronous game server.

```
def on_open(self, info):
    """
    Open connection and initialise states

    """
    # initialise a variety of per-connection variables
    ...

    # register a periodic callback on the Tornado's IOLoop to check
    # the health of the current connection. Close the connection
    # from server-side if three consecutive heartbeats missed
    ...

def on_message(self, msg):
    """
    React to different types of client message

    """
    # reset the counter of missed heartbeats
    ...

    try:
        # parse JSON message
        ...

        if the message is not the same as the last received:
            if msg_type == WAIT:
                # assign the subject to the waiting pool
                ...
            elif msg_type == REGISTER:
                # register subject to a game session
                ...
            elif msg_type == READY and self.registered:
                # start the game
                ...
            elif msg_type == ACTION and self.registered:
                # process stage actions
                ...
            elif msg_type == ACK and self.registered:
                # proceed to the next stage/round
                ...
            else:
                # unexpected/unknown message, close down connection
                ...
        else:
            # duplicate msg received from subject, ignore
    except json.JSONDecodeError as e:
        # JSON decode error, invalid message, log error
        ...
```

```
# code continues from the previous page
def on_close(self):
    """
    Handle expected and unexpected disconnections

    """
    # stop the heartbeat check
    ...

    # handle different disconnection scenarios
    if self.game_session is None:
        # possible cases: rogue/duplicate subjects, session does not exist
        # ignore
    elif not self.registered:
        # possible cases: disconnection while waiting, redundant subjects
        if self in waiting connections:
            # remove subject from the waiting pool of the current session
            ...
    elif self.registered and game session state == 'PENDING':
        # possible cases: disconnection before/after receiving READY msg
        if self.last_received == READY_MSG:
            # decrease the ACK counter by one for the session
            # reset self.last_received in case of re-connection
            ...
        # wait 10 secs for re-connection, after which de-register subject
        ...
    elif self.registered and game session state == 'STARTED':
        # possible cases: disconnection before/after the game is finished
        if game is not finished:
            # register a pending (up to 20 secs) takeover by program player
            ...
        else:
            # prepare in case of disconnection before receiving FINISH msg
            ...
```

5.3.3 Customisation via Modular API

As a platform for developing and running game theoretic experiments, UbiquityLab handles customised experiments with significantly reduced effort. It also allows live configuration such that experimenters can manage live experimental sessions. In this section, we demonstrate the the server-side modular API using which experimenters can customise various aspects of an experimental game.

Our solution is helped by the fact that Python is an interpreted dynamic programming language with a wide range of libraries and an easy learning curve. As

a popular general programming language, Python offers great support for almost any computational task, and is very accessible to beginners. Python also conveniently include built-in support for dynamic module loading. Given these features, we choose to adopt a modular design where each minimal Python module implements a few API functions to customise some aspect of the game. Using this approach, a complete game is defined by a set of Python modules and is activated once these modules are dynamically loaded on the game server. This design lends itself to a high degree of flexibility and breaks down an experimental implementation into small and well-defined modules for ease of development. We present this modular API for synchronous games as the following.

Setup: Games may be initialised to some specified state. For example, participants get assigned a fictitious ID upon joining which may or may not change during the game. Players may also be matched up according to preconfigured interactions. Such matching is generally defined using a PlayGraph instance. The `Setup` module executes once per participant and contains a single function:

```
def generate(node, play_graph):
    """
    Initialises a player given its position on the play graph
    Returns a dictionary object containing per-player settings

    node -- node ID assigned to the player on the play graph
    play_graph -- PlayGraph instance of the current session

    """
    pass
```

Action: In practice, synchronous games can be generalised as repeated single or multi-stage games. UbiquityLab supports arbitrary number of stages per round. At the end of each stage, `Action` module aggregates and centrally processes inputs by participants, and generates feedback that are distributed back to participants. Feedback may contain two types of information: results of the current stage or round and auxiliary information for the next stage or round. `Action` module contains two

functions:

```
def get_total_stages():
    """
    Returns the total number of stages in the base game (i.e. in a round)

    """
    pass

def process(rnd, stage, session, players, actions, play_graph, size, params):
    """
    Processes the actions for the given stage of the given round
    Returns a dictionary object containing feedback/results

    rnd -- current round
    stage -- current stage
    session -- current game session
    players -- a dictionary of player (connections) keyed by subject ID
    actions -- history of all players' actions
    play_graph -- the PlayGraph instance of the current session
    size -- number of players in the game
    params -- custom parameters specific to the game for result processing

    """
    if stage == 1:
        # generate stage 1 feedback/results
        pass
    elif stage == 2:
       # generate stage 2 feedback/results
       pass
    ...
    pass
```

Program: `Program` module ensures that every session carries on in the event of dropout. It is highly recommended to implement this module to avoid disruption to participants. The module contains two functions:

```
def init(subject, play_graph, params):
    """
    Initialises for a given dropped out subject

    subject -- subject ID, e.g. MTurk IDs
    play_graph -- PlayGraph instance of the current session
    params -- custom parameters specific to the game for program play

    """
    pass
```

```
# code continues from the previous page

def play(node, play_graph, params, actions, stage):
    """
    Simulates an action for the given player at the given stage
    Returns a dictionary object containing the action

    node -- node ID of the dropped player on the play graph
    play_graph -- PlayGraph instance of the current session
    params -- custom parameters specific to this game for program play
    actions -- history of all players' actions
    stage -- current stage

    """
    if stage == 1:
        # generate stage 1 program action
        pass
    elif stage == 2:
       # generate stage 2 program action
       pass
    ...
    pass
```

Data: The game server collects and records experimental data at the end of each stage of each round. All data is stored in .CSV format. Each experiment specifies a list of headers (i.e. column names) for each stage of each round, and returns a comma-separated string at the end of each stage. Experimenters can also initialise constant values at the start of the experiment through `Data` module. The module contains three functions:

```
def configure(session):
    """
    Initialises common/constant variables used by data collection

    session -- current game session

    """
    pass

def action_headers():
    """
    Returns a dictionary object containing the column names for each stage

    """
    pass
```

```
# code continues from the previous page

def record(session, subject, node, play_graph, rnd, stage, action, fdback, conn):
    """
    Records the action and resulted feedback/results for the given stage
    Returns a comma-separated data string according to the stage headers

    session -- current session
    subject -- subject ID
    node -- node ID on the play graph
    play_graph -- PlayGraph instance of the current session
    rnd -- current round
    stage -- current stage
    action -- current stage action
    fdback -- processed feedback/results
    conn -- subject connection instance

    """
    if stage == 1:
        # construct stage 1 data string
        pass
    elif stage == 2:
       # construct stage 2 data string
       pass
    ...
    pass
```

Payoff: `Payoff` module is responsible to derive payment details for respective participants. This process involves selecting a subset (or all) of the round results for each participant randomly or algorithmically, and return relevant details of the selected results to construct a statement of payment. To maximise transparency and credibility, the statement should make details such as selected rounds, round action(s), and round results available to the subjects on the payment confirmation page. `Payoff` module includes two functions:

```
def configure(length, size, params):
    """
    Initialises common/constant variables used by payoff calculation

    length -- length of the game in number of rounds
    size -- number of players in the game
    params -- custom parameters specific to this game for payoff calculation

    """
    pass
```

```
# code continues from the previous page

def settle(payoffs):
    """
    Derives the total payoff and its relevant details for the given round payoffs
    Returns a dictionary object containing the results for each selected round

    payoffs -- a list of per-round results ordered by round number

    """
    pass
```

5.4 Future Roadmap

We plan to further develop UbiquityLab into a full-fledged platform for online experimentation. This implies that UbiquityLab will manage and operate its own subject pool and payment service, rather than relying on a third-party online labour market. It also means that the platform will be open to other researchers and organisations and will aim to further promote online experiments as an alternative research methodology in experimental economics and beyond. We outline below several key aspects of the platform we would like to improve in the medium to long term.

Subject monitoring: We are developing a monitoring system that tracks and stores the entire journey of each participant, from the initial waiting room to the very last payment confirmation. At the moment, such data is collected and stored in the form of log files on our server. In contrast, the new subject monitoring system is accessible to experimenters via a dashboard-style page while the experimental sessions are live, so that experimenters can monitor participant behaviour and session progress in real-time without the use of log files. Examples of collected data may include: number of active subjects at each stage or page, statistics of subjects' performance at the Quiz, game session progress and timestamps, dropped out subjects, etc. All these data will be stored structurally in a database so that further analysis can be carried out to suggest improvements to the platform.

Subject management: As aforementioned, the platform will source subjects directly in the near future without relying on an intermediary. To do this, we will improve our subject management system in the following areas: 1. Easy registration/sign-up for new subjects; 2. Detailed statistics of subject participation; 3. Recruitment and notification for experiments. One particular consideration we take into account is that experimenters should be able to filter subjects who have participated in similar or related experiments since this may introduce undesirable effects or biases. At the moment, the platform records and store all such data but most operations have to be done by running scripts manually. Furthermore, a complete subject management component will allow demographically-specific studies by utilising the combination of self-reported demographics and participating IP addresses.

Scalability: The largest number of simultaneous sessions that UbiquityLab has hosted so far is 8 sessions in which each session consists of 17 subjects including backups. This is by no means a large amount of traffic to a website in production. It is our goal that the platform can handle thousands of live participants at once. A larger scale of operation will achieve two aims: 1. Running large-scale experiments that involves more subjects than what is physically feasible in labs; 2. Offering services to other researchers through a single platform. To scale up, we may need to distribute the game servers across multiple physical machines with upgraded computing power, and it may require some changes to the existing architecture of the platform.

Experiment development: Experimenters currently implement experiments by programming minimal server-side modules as well as client-side game interface. Depending on the experimental design, the complexity of implementation may vary. In the long run, we hope to further simplify implementation, especially for the client-side. Ultimately, the construction of game interface should be as graphical and interactive as possible. A set of common and standardised graphical components may be listed, and experimenters will simply drag and drop different components to compose an

interface for which the underlying HTML and CSS code will be automatically generated. We may reserve the flexibility of adding customised components through raw scripting of HTML, CSS and JavaScript.

Data analysis: Last but not least, we hope to provide analytic capabilities at UbiquityLab. These analytics will allow experimenters to quickly assess completed sessions before delving into the data for more thorough analysis. Such facility may prove to be especially helpful when piloting new experiments. It may also provide some basic data processing and analysis functionalities such as those available at *Qualtrics* [Qualtrics, 2013]. Given the choice of Python as our main server-side programming language, we enjoy a wide selection of libraries for scientific computing and data analysis. We plan to use *NumPy* [Oliphant, 2007] and *Pandas* [McKinney, 2011] for computation and *d3.js* [Bostock, 2011] for visualisation.

Chapter 6

Conclusion

As the amount of accessible data keeps growing, it is essential (and arguably more difficult) to make good use of available information to facilitate our decision-making in various settings. Reputation has been an effective mechanism to inform decision-making in both commercial and social interactions. However, we have yet to harness its full power and understand its complete function in the age of information abundance thanks to the Internet. This thesis aims to further the research of reputation in different contexts.

6.1 Summary

We make three contributions in this thesis. Theoretically, we propose a personalised reputation mechanism for online services that promotes truthful reporting from raters and resists common manipulations, and show that the mechanism is polynomial-time computable. Experimentally, we investigate the relative importance of reputational and social knowledge in dynamic cooperative network, and show that reputation is the main driver of promoting cooperation and social knowledge plays a complementary role in enhancing cooperation. Methodologically, we present an innovative online platform for real-time interactive experimentation and its accompanying workflow

for conducting multiplayer experiments while adhering to the standard practices of experimental economics wherever possible.

6.2 Future Directions

In addition to the discussion at the end of each main chapter, we discuss below some additional thoughts on the future directions of the research in this thesis.

Online Reputation Mechanisms: Our current mechanism does not take into account the number of available ratings for a target. Psychologically and informationally, the number of ratings may offer additional insight to evaluating a target's reputation. Furthermore, being able to reveal information about the distribution of ratings may make the mechanism more user-friendly. However, this may prove tricky for BTS-like algorithms. A possible workaround could be disclosing the distribution of only a subset of ratings available. An alternative approach would be to swap BTS with some other incentive compatible algorithm such as the *Peer Prediction* method and alike [Miller et al., 2005, Witkowski and Parkes, 2011, Witkowski and Parkes, 2012a], which do not have to conceal the distribution of ratings.

Moreover, we could aim to make the mechanism robust against more sophisticated collusions. One such collusion can be the following: a group of cheating agents form the majority of raters for a target so that they can manipulate which rating turns out to be surprisingly common according to the BTS criterion. Those agents who benefit from this colluded result will attain high truthfulness scores, and may be able to manipulate the target's reputation in the eyes of others if they build up their similarity scores by strategically rating other popular targets. A possible solution would be to periodically carry out graph theoretic analysis to detect such coalition of agents based on their rating patterns and remove the ratings made by the coalition. Last but not least, one could investigate the possibility of extending our mechanism

to other domains where the sets of entities involved are not disjoint but overlapping.

Reputation and Cooperation: Still within the context of cooperative dynamic networks, an interesting extension to our study is to add weights to the links between participants. The weight of a link in this case would indicate the degree of bonding or trust between two neighbours, and the payoffs from the resulting prisoner's dilemma game would adapt based on this weight. For instance, given two possible weights w^H, w^L ($w^H > w^L$) of a link, the payoffs for different outcomes of the prisoner dilemma game would be adjusted given the realised weight, such that $(C,C)^H > (C,C)^L$, $(D,C)^H > (D,C)^L$, $(D,D)^H < (D,D)^L$, and $(C,D)^H < (C,D)^L$. This setting corresponds to the intuitive observation that the magnitude of both gain and loss in a relationship closely depends on the depth of the relationship. It would be intriguing to see how subjects react to this heuristic in terms of both their link updating and cooperative behaviour.

In addition to our experimental results, [Cuesta et al., 2014] independently confirms that reputation is also an important driver in the enhancement of cooperation in dynamic networks. On the other hand, relatively little is known about how people utilise social knowledge (who is connected whom in the network) in strategic interaction in general. A potentially fruitful direction is to extend the usage of social knowledge to other network games so that we can investigate how people may utilise it and behave in other contexts. [Kosfeld,] offers a potential source of inspirations.

Platform for Online Experiments: Besides the new functionalities outlined at the end of Chapter 5, we ought to further examine and improve various methodological aspects of the platform. For example, how should we deal with participants who may sign in from different time zones. This is clearly not an issue with lab experiments, but quite possibly one with online experiments. The time difference may or may not have an effect on the experimental outcome, nevertheless, it may still raise questions from some experimentalists. Consider another example, in experiments requiring a

large number of participants per session, it is possible that some fast participants may have to wait for 10+ minutes before the game starts. Will the waiting have any effect on these participants? If so, how big will the effect be? Subtle issues like these require examination to ensure the rigour of the platform as a scientific research tool, and we should approach them practically using experimental data.

On another front, we may need to devise new methodological arrangements to facilitate large-scale online experiments (e.g. on a similar scale of [Gracia-Lázaro et al., 2012]). To coordinate hundreds of participants for a single session, we may look into the possibilities of partnering with other web platforms (such as Facebook, Zooniverse, etc). Furthermore, as the platform opens its services to other researchers, we need to come up with ways to manage the subject pool appropriately so that the same subjects do not participate frequently in similar studies[1].

[1]This is also an issue in the lab, but we hope to come up with a solution that benefits from the large number of available subjects online.

Appendix A

Supplementary Information of the Experiment

A.1 Subject Sample

The main experimental data contains the decisions of 364 subjects, and multiple participations are not allowed. We ran a total of 30 sessions starting at 11am EST between the 12th and the 21st of December, 2013. *UbiquityLab* allows us to run several sessions concurrently in a single "meta-session." We ran 9 meta-sessions consisting of 2 to 5 sessions each. In 2 out of 30 sessions subjects dropped out so we excluded them from the main analysis, but the results are unchanged if we include these sessions (see Section 7 for details). A single experimental session lasted on average 49.6 minutes. Subjects were paid a fixed $2 fee and a bonus based on their performance. The average earnings was $5.13. These earnings are above the average hourly earnings for a Turker on AMT.

We match the experimental data with the Questionnaire data. Table A.1 summarises the main socio-demographic characteristics of the participants and their pairwise correlations. All the participants are residents in the U.S.: 43% are female and

the average age is 32.6 years old. We asked subjects the standard interpersonal trust question from the World Values Survey (*WVS*) and found that about 52% believe that others can be trusted, which is higher than the average value from the *WVS* survey of the U.S. population. The level of trust is not correlated with gender, age or education.

Table A.1: Summary statistics and pairwise correlations for the main variables.

	n	Mean	s.d.	Gender	Age	Education	Trust	HL
1. Gender[1]	364	0.43	0.50	1.00				
2. Age	364	32.63	10.33	0.06	1.00			
3. Education[2]	364	3.35	1.35	-0.00	0.11*	1.00		
4. Trust[3]	364	0.52	0.50	-0.05	0.09	0.00	1.00	
5. HL[4]	333	7.54	1.74	-0.00	-0.01	0.02	-0.02	1.00

*$P < 0.05$. 1. Female = 1. 2. See the caption of Figure A.1b for codings; 3. Trust question with 0 = "Need to be very careful" and 1 = "Most people can be trusted"; 4. Holt and Laury's risk attitude test: 21 participants are excluded because they made at least one inconsistent choice (i.e. multiple switching points) and 10 participants are excluded because they had no switching point.

Figure A.1 illustrates in more detail the composition of the subject pool, which is more representative of the general population than the student populations that are typical of most laboratory studies. Figure A.1a shows that subjects belong to different age groups ranging from a minimum of 18 to a maximum of 70 years old. There is also a significant heterogeneity in the education level of participants. Figure A.1b shows that 6.9% of participants only have a high school diploma, 76.9% have attained some kind of college education, and 15.9% have a master, professional degree or Ph.D. As one would expect, there is a significant positive correlation between age

and education level.

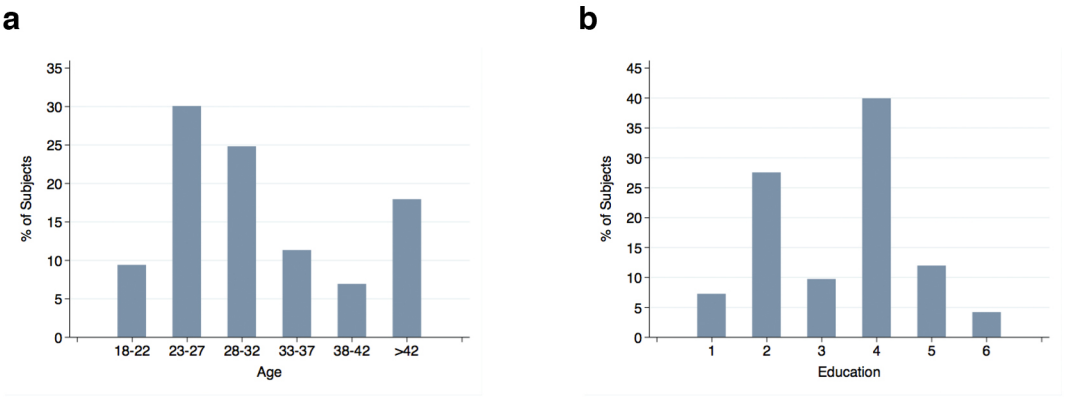


Figure A.1: Percentages of subjects of different age groups (left) and education levels (right) for the 364 participants. Classification of education levels: 1 = high school or less, 2 = some college, 3 = 2-year college, 4 = 4-year college, 5 = master, 6 = professional degree (e.g. MD) or PhD.

During the Questionnaire, subjects also took the Holt and Laury's risk attitude test [Holt and Laury, 2002]: there are 10 scenarios and in each scenario a subject has to pick between a safe and a risky lottery. In the early scenarios the safe lottery gives a higher expected payment, while in the late scenarios it is the opposite. A risk-neutral participant would switch from the safe to the risky lottery at scenario 5. The mean switching point is 7.5 so the participants are on average risk-averse. There are 10 (3%) participants who switched at or before scenario 4 and therefore are risk-seeking, and 40 (12%) participants who are risk-neutral. There is no significant correlation of risk aversion with gender, age, education or trust.

A.2 Subject Participation

We report additional data on subjects' participation of the experiments.

On average, subjects take 18.9 minutes ($SD = 3.4$) to complete the Instructions and the Quiz. There are slight variations in the Instructions and the Quiz across treatments, so the average time spent on them are 18.1, 20.4, 17.1 and 20.0 minutes

for treatments B, N, R and RN respectively.

In the Quiz there are 5 questions in the B and the R treatments, and 6 questions in the N and the RN treatments. The extra question in the N and RN treatments tests subjects' understanding of the interactive features of the network figure in the interface, which are available only in those two treatments. Subjects have a maximum of 3 tries to pass the Quiz, and in case they fail the third try they are not allowed to continue to the experiment. Moreover, after each unsuccessful attempt, we randomly re-generate the numbers in the failed question(s) for the subsequent try to ensure it is very unlikely that subjects pass the Quiz by randomly guessing the answer through trial and error. Overall, subjects take on average 1.5 ($SD = 0.7$) tries to answer correctly all the questions. The mean numbers of tries for treatments B, N, R and RN are 1.5, 1.6, 1.4 and 1.5 respectively, and there is no significant difference across treatments. 64.0% of subjects pass the Quiz in their first try, 24.7% pass with two tries, and 11.3% need three tries. For those subjects who do not pass the Quiz in their first try, they make on average 1.4 ($SD = 0.6$) mistakes in their first try and 1.3 ($SD = 0.5$) mistakes in their subsequent try (if required).

On average, subjects spend 20.9 ($SD = 3.6$) minutes to play the experimental game. The mean time for subjects to submit their choices in the first stage is 16.3 ($SD = 8.2$) seconds, the mean time in the second stage is 11.3 ($SD = 6.8$) seconds, and the mean time in the third stage is 9.1 ($SD = 4.0$) seconds. During the N and RN treatments, we implement real-time tracking of subjects' mouse movements on the network figure of the interface, in terms of both the number of times a subject hovers over a node with the mouse pointer to highlight that node and its neighbours, and the number of times a subject clicks and drags the nodes to re-arrange the network layout. Figure A.2 below shows the evolution of both measurements over the course of the game for both the N and RN treatments. On average, subjects hover over 4.3 ($SD = 1.5$) and 4.9 ($SD = 1.3$) nodes in each round of the RN and

N treatments respectively. Every instance of node dragging is also recorded as an instance of hovering over a node, so we measure the usage of node dragging per round in terms of the percentage of times that hovering over a node involved node dragging as well, as shown by the red line in Figure A.2. Subjects drag a node to re-arrange the network 10.5% of times they hover over it in the N treatment, and 11.8% of times in the RN treatment. The relatively lower usage of the node dragging feature may indicate that our customised force layout algorithm is effective at displaying the network and subjects do not need to re-arrange its layout manually.

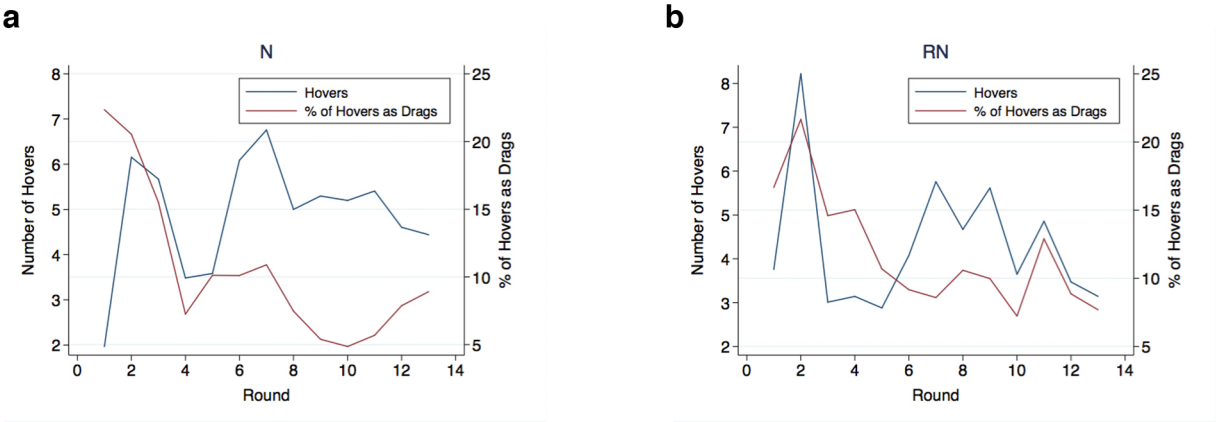


Figure A.2: Usage of the interactive features of the network figure in treatments (a) N and (b) RN. The blue line shows the evolution of the usage of hovering over a node during the game, while the red line shows the evolution of the usage of node dragging (as a percentage of instances of hovering over a node).

Finally, we examine subjects' decisions in the first round of the game across treatments. Treatments R and RN begin with qualitatively higher cooperation levels, and the effect is marginally significant for R compared to both treatments N and B (R vs. N, $P = 0.055$; R vs. B, $P = 0.041$). As a result, subjects earn a higher payoff on average in the first round in R than in N and B (R vs. N, $P = 0.047$; R vs. B, $P = 0.035$). A potential explanation is that subjects are more cooperative because they are aware that their action will be common knowledge to the group, and this effect is slightly more pronounced in the R treatment than in the RN one because

the lack of global network information makes the presence of reputational information more salient. There are no significant differences across treatments in network density and network clustering in the first round.

A.3 Robustness Checks

We check that our results are robust to the inclusion of the sessions with dropped out subjects.

There are a total of 2 sessions (out of 30) in which subject(s) drop out during the game. In treatment R, four subjects subsequently dropped out during the game in rounds 1, 2, 5 and 9. In treatment RN, one subject dropped out in round 3. UbiquityLab actively prevents drop-outs caused by temporary network problems, and any temporarily disrupted connection is automatically restored upon detection. However, we cannot circumvent issues such as permanent loss of power or Internet connection, crash of subject's browser or operating system, and voluntary termination. In the event of a drop-out, UbiquityLab automatically pauses the game at the end of the affected stage, notifies remaining subjects about the drop-out(s), and resumes the game after removing the subject who dropped out.

In this section, we re-run the analyses in the main text after including the data collected in these two sessions with drop-outs, and show that our results are robust to the inclusion of this additional incomplete data.

At the aggregate level, Figure S9 shows that the evolution of the cooperation level, average payoffs, density and clustering of the network in the different treatments is essentially unchanged compared to Figure 1 in the main text. All the differences in these metrics between treatments remain statistically significant after including the sessions with drop-outs. The difference in the level of cooperation between the R and B treatments, and the differences in clustering between R and N and between

R and B become significant with the inclusion of the new data. This is because R is the treatment with the session with 4 drop-outs, which increases the values of both metrics due to the significantly smaller size of the network.

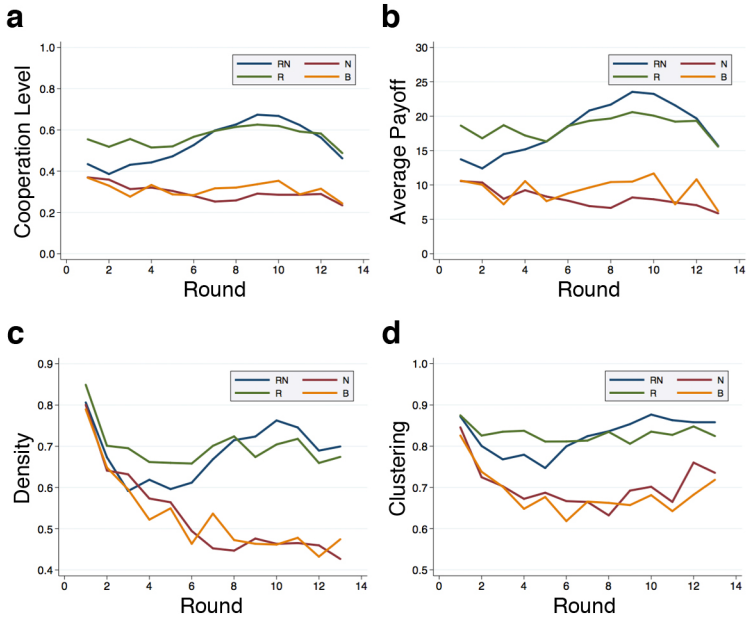


Figure A.3: Cooperation level (a), average payoff (b), density (c), and clustering (d) over 13 rounds of play for treatments B (yellow), N (red), R (green) and RN (blue) after including "drop-out" sessions. Analyses are based on aggregated data after round 5 in each session. These are the results for Rank-sum tests comparing the differences across treatments, we only include significant ($P < 0.05$) and marginally significant ($P < 0.1$) findings: (a) Cooperation level: RN vs. B, $P = 0.008$; RN vs. N, $P = 0.015$; R vs. B, $P = 0.028$; R vs. N, $P = 0.037$. (b) Average payoff: RN vs. B, $P = 0.011$; RN vs. N, $P = 0.021$; R vs. B, $P = 0.049$; R vs. N, $P = 0.049$. (c) Density: RN vs. B, $P = 0.001$; RN vs. N, $P = 0.008$; R vs. B, $P = 0.015$; R vs. N, $P = 0.028$. (d) Global clustering: RN vs. B, $P = 0.004$; RN vs. N, $P = 0.006$; R vs. B, $P = 0.037$; R vs. N, $P = 0.037$.

At the community level, all the results in the main analysis are unchanged. For the largest community $C1$, the RN treatment achieves a significantly higher level of cooperation, density and clustering compared to the N and B treatments ($P = 0.021, P = 0.015, P = 0.011$ for cooperation, density and clustering compared to N; and $P = 0.028, P = 0.011, P = 0.011$ for cooperation, density and clustering compared to B). As in the main analysis, there are no differences across treatments in terms of the cooperation level, density and clustering of the $C2$ community. RN

remains the only treatments where $C1$ is more cooperative than $C2$ ($P = 0.009$). Due to the smaller size of the session in R with 4 drop-outs, the cooperation level of community $C1$ becomes marginally higher than that of $C1$ in B ($P = 0.049$), and the level of clustering of $C1$ becomes higher than that in N ($P = 0.037$).

At the subject level, the results of the regression analysis are unchanged. The complete regression table are available upon request.

A.4 Experimental Instructions (for treatment B)

General rules[1] **1/7**

There are 3 parts to the Experiment:

- Instructions
- Experiment
- Final Questionnaire

You will earn $2 for sure only if you complete the Experiment and the Final Questionnaire. In addition, you can earn Experimental Points (EP) that will be converted into real earning in dollars. We expect the average total earning to be within the $4 - 7 range, but your actual earnings may vary considerably depending on your performance.

The expected duration of the Experiment is about 45 minutes, and you need to fully dedicate your time to this Experiment for the next 45 minutes. If you exit at any point before completion, you will not receive any earnings.

The aim of this Experiment is to study how individuals make decisions in certain contexts. You will make decisions that will affect the amount of points you earn and the amount of points other Turkers earn.

[1]Each section heading corresponds to a web page of the Instructions.

All your decisions will remain completely confidential. We will not disclose your Turker ID or any other information that may allow others to identify you. Each Turker will be assigned a one-letter ID that is kept the same throughout the Experiment. ID assignments are random and carry no meaning.

You will be asked to take a Quiz later to ensure that you understand the Instructions. If you cannot pass the Quiz within 3 tries, you will not be able to participate in the Experiment.

The expected average time before you reach the Quiz is 15 minutes, and it is important that you read through the Instructions carefully. Note that each participant is shown exactly the same Instructions.

Please click the 'Next' link below to continue.

The game 2/7

You and 12 other Turkers will play several rounds of a "game". Each round is the same and consists of 3 stages.

In Stages 1 and 2, you can form and cut links with any of the other Turkers, which we will explain shortly. If you are linked with another Turker then that Turker is a "neighbor".

In Stage 3, you choose either action A or action B, and the choice of action A or B applies to all your neighbors. Action **A** is color-coded in **green** and action **B** is color-coded in **blue**. The colors are only a visual aid to distinguish the actions and have no meaning. You get points for the action you choose and the action each of the other Turkers chooses, in the following way:

You get 0 points if you are not linked with another Turker regardless of your choice of action.

The number of points you get depends on the actions you and your neighbor choose, according to the table below:

		Neighbor	
		A	**B**
You	**A**	3	–5
	B	5	–3

This is what the table says:

- If you choose **A** and the neighbor chooses **A**, you get 3 points
- If you choose **A** and the neighbor chooses **B**, you get -5 points
- If you choose **B** and the neighbor chooses **A**, you get 5 points
- If you choose **B** and the neighbor chooses **B**, you get -3 points

At the end of each round you will see a summary of the number of points you get with each of the other Turkers.

You will play 13 rounds of this game for sure. After round 13, there is a 50% chance that the game will terminate in the following round. In other words, in every round after round 13 the computer will "flip a coin" and if the outcome is "Heads" then the game will terminate.

Please click the 'Next' link below to continue to a detailed description of Stage 1.

Stage 1 - Link decisions 3/7

In Stage 1, you decide which Turkers you want to link with.
We will explain each part of the interface separately and you will see the whole interface in the Tour later.

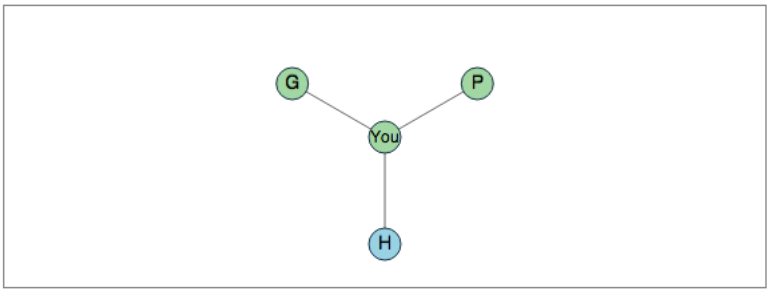
Top-right of the screen The first row shows the current stage and the current round you are in. You see a countdown timer that indicates the time you have left to make your decisions. In every round, you have 70 seconds to complete Stage 1.

The second row reminds you of the actions you have chosen in last 5 rounds. For example, the sequence below means that you chose action A in last round (the right-most green square), action B two rounds ago (blue square in the 2nd slot counting from the right), action B three rounds ago (blue square in the 3rd slot counting from the right), etc.

Note that if you do not have any neighbor in a round, then you cannot choose an action in that round, and your round action is denoted by a gray box .

There is also a "Table of Points" link: you can move your mouse pointer over it to show the table of points you saw previously. Try it now!

Top-left of the screen This figure visualizes your current neighbors.



The neighbors you have at the start of each round are the same as the neighbors you

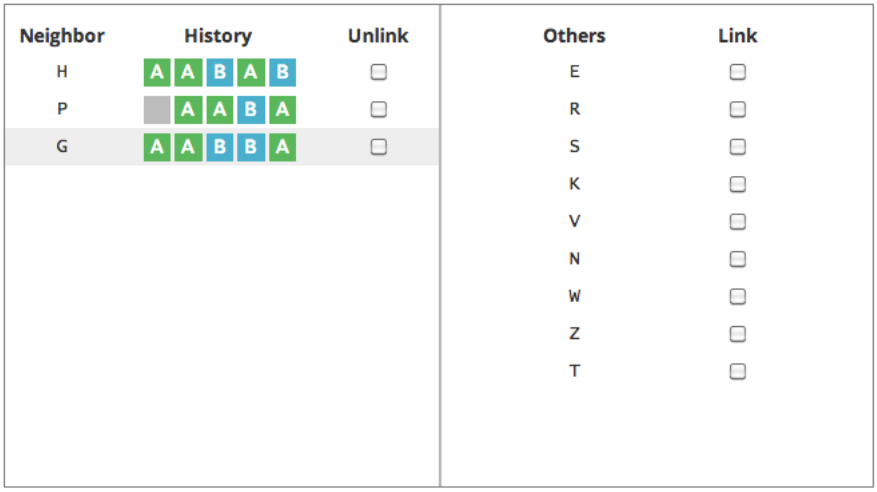
had at the end of the previous round. The links are reciprocal so if you are linked to Turker G then Turker G is linked to you.

Each circle denotes a Turker with an ID. The "You" circle is always positioned at the center. For example, above you are linked with H, P, and G.

The colors of the circles denote the action chosen by each Turker in the previous round. For example, in the above figure Turker P chose action **A** and Turker H chose action **B** in the previous round.

Note that no links exist between Turkers in Stage 1 of Round 1, so only the "You" circle will be displayed.

Middle-left of the screen This table allows you to make decisions in Stage 1.



You can cut a link to any neighbor by ticking the corresponding box under the "Unlink" column. You can also propose a link to any other Turker by ticking the corresponding box under the "Link" column. There is no limit to how many boxes you can tick, and you can also choose not to cut or propose any link by leaving all the boxes unticked.

The table has 5 columns, from left to right:

Neighbor: lists the IDs of Turkers who are currently linked with you.

History: lists last 5 actions chosen by each of your neighbors.

Unlink: allows you to cut your link to any of your neighbors. For example, ticking the box in the first row will cut your link to Turker H.

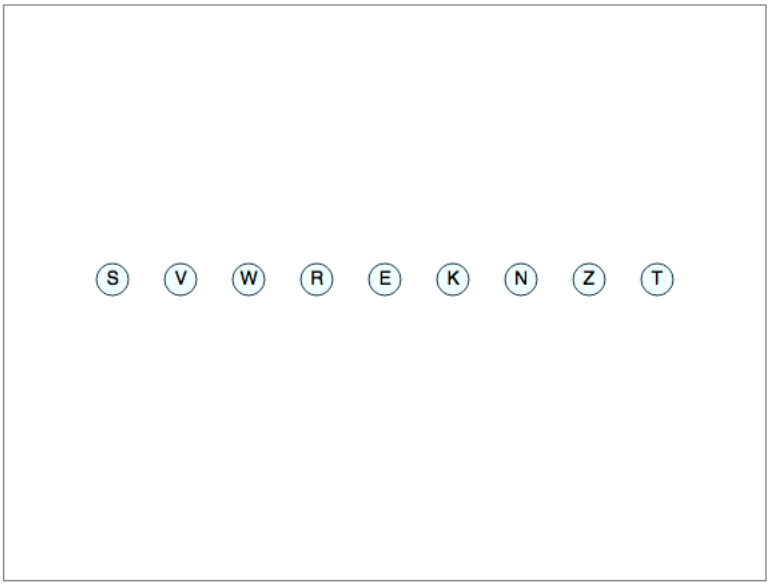
Others: lists the IDs of Turkers who are not currently linked with you.

Link: allows you to propose a link to any of the Turkers you are not linked with. For example, ticking the box in the first row proposes a link to Turker E.

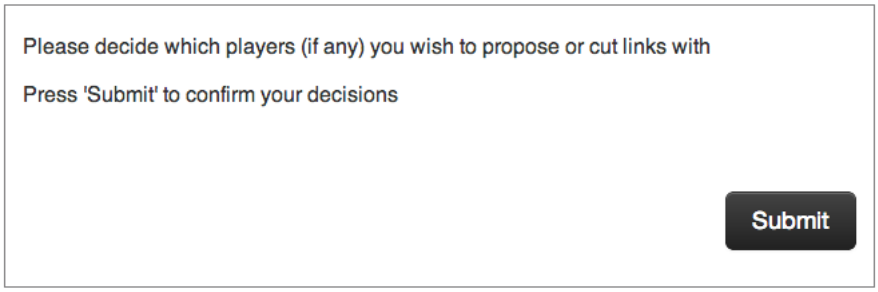
The color-codings of the other Turkers' actions are the same as before. The green box A denotes the choice of action **A**. The blue box B denotes the choice of action **B**. The gray box denotes that the Turker did not have any neighbor in that round, and therefore the Turker did not choose an action. For example, no action was chosen by Turker P five rounds ago.

Note that in Round 1 all the Turkers will be listed under "Others" because no links exist between Turkers.

Bottom-left of the screen This figure shows you the Turkers you are not currently linked with. Each circle denotes a Turker with an ID.



Bottom-right of the screen The text reminds you about the decisions you need to make in Stage 1. You can propose new links and cut existing links, or you can choose not to propose or cut any links. In either case, you need to click the "Submit" button to confirm your decisions.



Please click the 'Next' link below to continue to a detailed description of Stage 2.

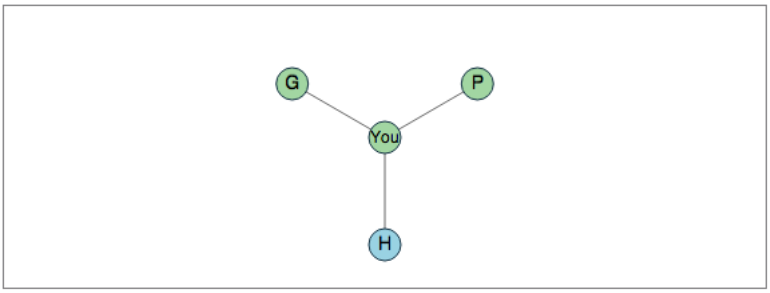
Stage 2 - Yes/No responses 4/7

In Stage 2, you decide whether to accept or reject link proposals from other Turkers.

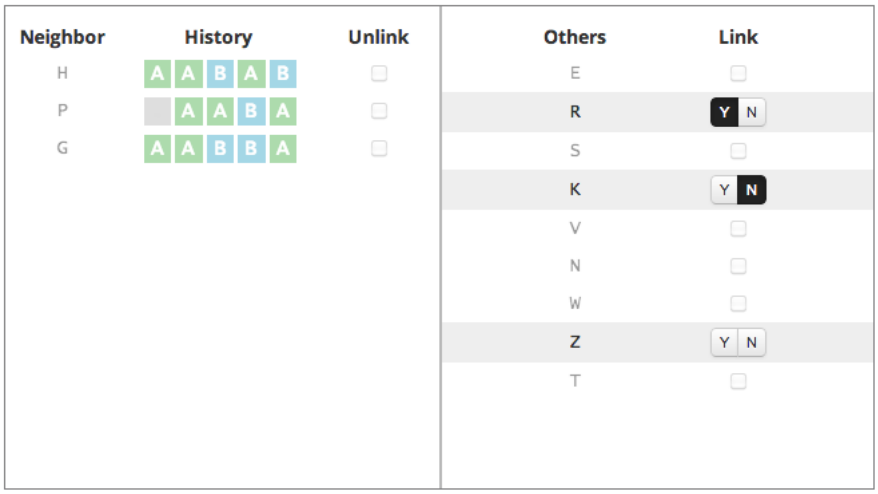
Top-right of the screen In each round, you have 45 seconds to complete Stage 2.

The second row is the same as in Stage 1: your actions in last 5 rounds and the "Table of Points" link.

Top-left of the screen This is the same as in Stage 1: a figure that shows your current neighbors. Note that the figure is not updated yet with yours and others' link decisions in Stage 1. The colors of the circles denote the action chosen by each Turker in the previous round.



Middle-left of the screen This table allows you to make decisions in Stage 2.



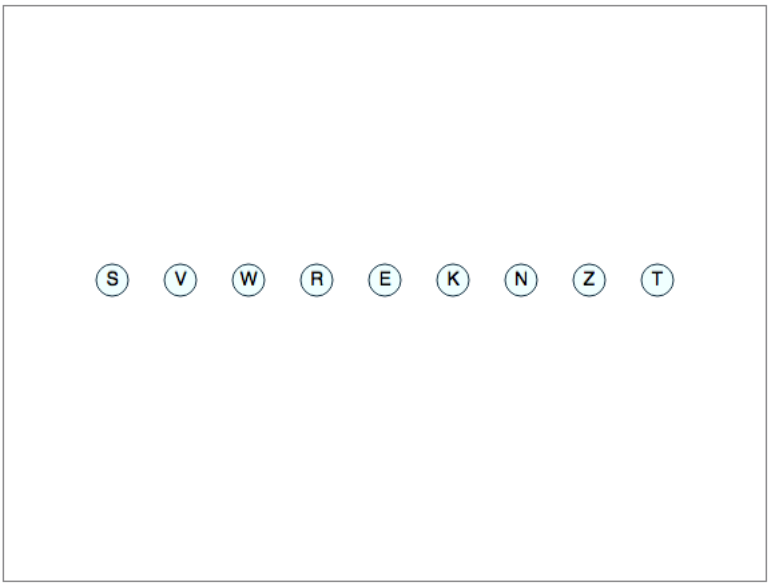
The first three columns from the left ("Neighbor", "History", "Unlink") are the same as in Stage 1.

The "Others" column lists the Turkers you are not linked with.

Now look at Turkers R, K and Z: under the "Link" column a "Y/N" ("Y" for Yes, "N" for No) button appears. This means that R, K and Z have sent you a link proposal. For each proposal, you need to click on "Y" to accept it or "N" to reject it. In the example above, you have chosen to accept the proposal from R and reject the proposal from K, but you have not responded yet to the proposal from Z.

Note that your choice to accept or reject a proposal is highlighted in black.

Bottom-left of the screen This is the same as in Stage 1: a figure that shows the Turkers you are not linked with. Note that the figure is not updated yet with yours and others' link decisions in Stage 1.

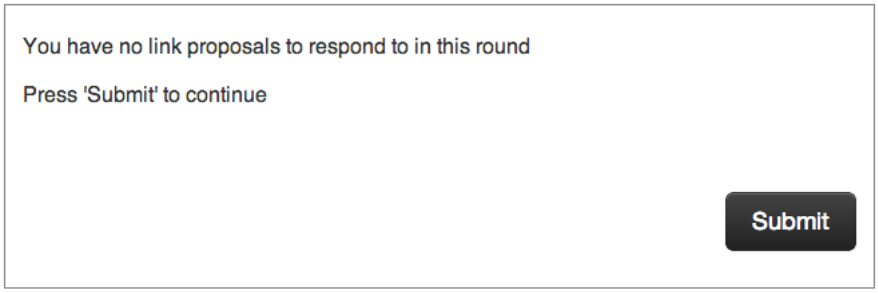


Bottom-right of the screen The first 3 rows remind you of your responses.

Before you press "Submit", you can change your responses and the text in the second and third rows will be updated accordingly.

You should click the "Submit" button after you have responded to all the proposals. If you do not respond to every proposal by the end of the stage, the computer will automatically reject any proposal that you have not responded to.

Note that it is also possible that no Turker proposes to link with you in Stage 1. In such case you will see the message below and you will need to wait for the other Turkers to complete Stage 2.



Please click the 'Next' link below to continue to a detailed description of Stage 3.

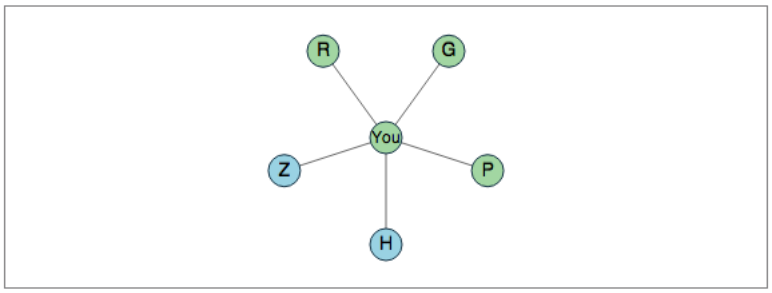
Stage 3 - Action choice 5/7

In Stage 3, you choose either action A or action B.

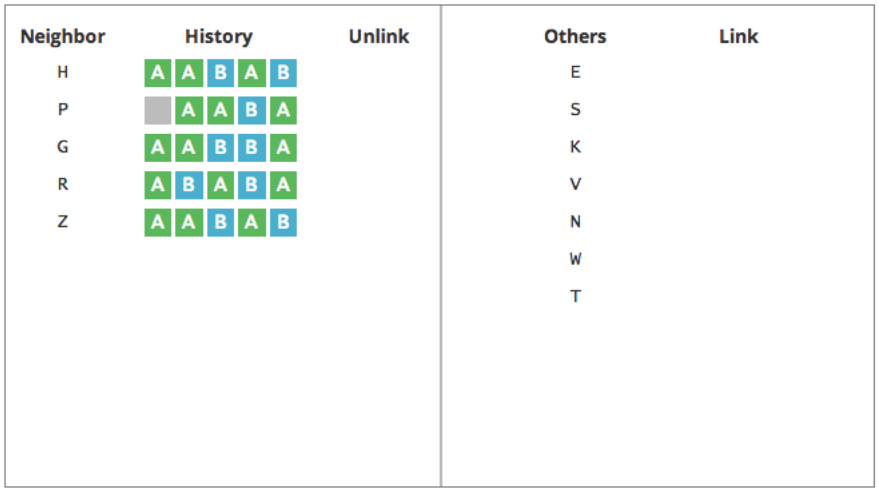
Top-right of the screen In each round, you have 25 seconds to complete Stage 3.

The second row shows your actions in last 5 rounds. Please move your mouse pointer over the "Table of Points" link to remind yourself about how you get points.

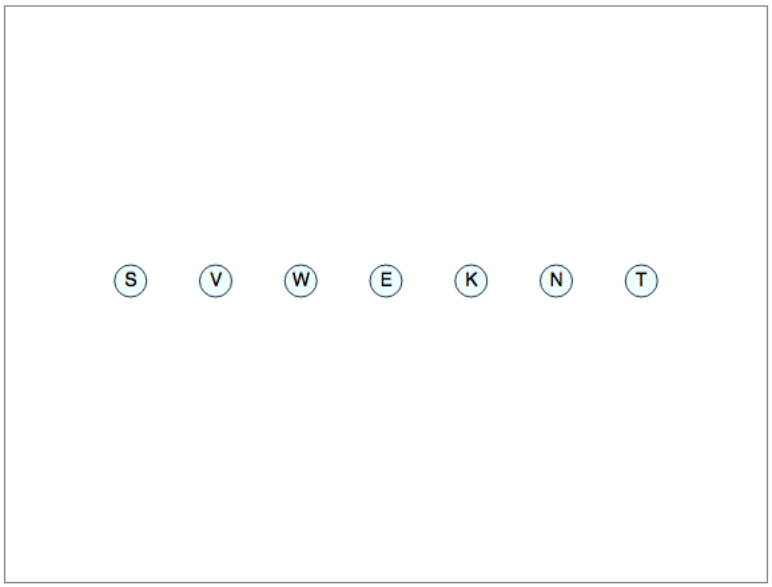
Top-left of the screen This figure shows your current neighbors after Stage 1 and 2. Note that the figure is now updated with yours and the others' decisions in the previous stages. The colors of the circles denote the action chosen by each Turker in the previous round.



Middle-right of the screen This table shows your neighbors, the Turkers you are not linked with, and your neighbors' actions in last 5 rounds. Note that the table is updated with yours and the others' decisions in the previous stages.



Bottom-left of the screen This figure shows the Turkers you are not linked with. Note that the figure is now updated with yours and others' decisions in the previous stages.

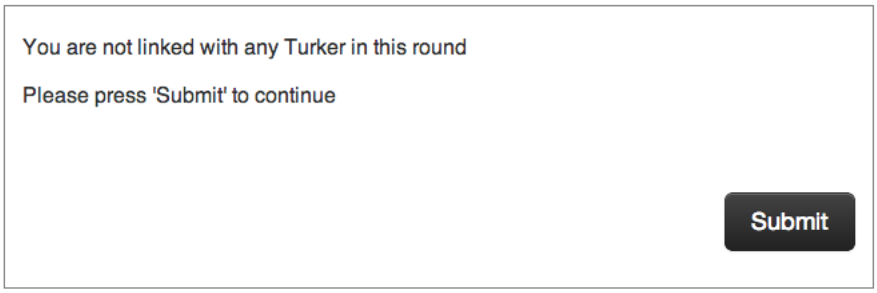


Bottom-right of the screen This is where you choose your action in Stage 3 by clicking either the A or B button.

Once an action is chosen, the "Invalid" text will change to indicate your choice. Note that if you do not click any button then the computer will choose action B by default.

You click the "Submit" button to confirm your choice.

Note that it is possible that you are not linked with any Turker (i.e. you have no neighbor) as the result of everyone's earlier decisions. In such case, you will instead see the message below, and you should click the 'Submit' button to continue.



Please click the 'Next' link below to continue.

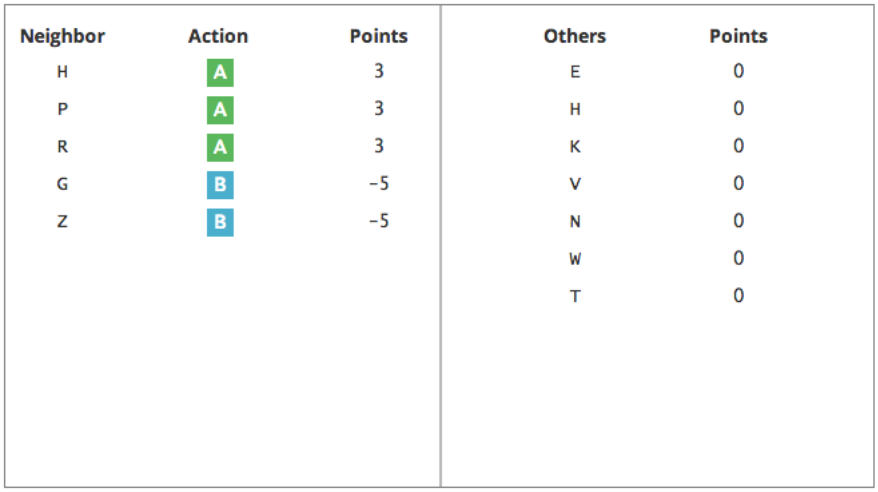
Round results 6/7

After Stage 3, you will see a summary of the points you got with each of the other Turkers in the round.

Top-right of the screen In each round, you have 15 seconds to see the results.

The second row reminds you of the action you have just chosen.

Middle-right of the screen This table summarizes the results of the current round.



Neighbor	Action	Points
H	A	3
P	A	3
R	A	3
G	B	-5
Z	B	-5

Others	Points
E	0
H	0
K	0
V	0
N	0
W	0
T	0

There are 5 columns, from left to right:

Neighbor: lists the IDs of Turkers who are linked with you.

Action: lists the actions chosen by your neighbors.

Points: lists the points you got with each of your neighbors.

Others: lists the IDs of Turkers who are not linked with you.

Points: lists the points you got with each of the Turkers who are not linked with you, which are always 0.

Please click the 'Next'link below to continue.

Your bonus **7/7**

At the end of the Experiment, we will randomly select 6 rounds for payment. In each of these 6 rounds, we will randomly pick 2 other Turkers. Note that a Turker who was not linked with you can be picked, and in that case you will get 0 points for the game with that Turker.

To determine your bonus, we sum the number of points you got with each of the picked Turkers in each of the 6 rounds. The exchange rate from Experimental Points (EP) to US dollars is:

$$5 \text{ EP} = 1 \text{ dollar}$$

For example, the table below shows the payment details of an imaginary Turker.

Selected round	Selected turker	Your action	Turker's action	Your points
2	S	B	A	5
2	H	B	A	5
5	W	A	B	-5
5	P	unlinked	unlinked	0
8	W	A	A	3
8	R	A	B	-5
9	N	B	B	-3
9	E	B	A	5
11	Z	A	A	3
11	K	A	A	3
13	P	unlinked	unlinked	0
13	R	B	A	5
Total				16

This imaginary Turker's total earnings are:

$$\text{Total earnings} = 200 + 1 \times \frac{16}{0.05} = 200 + 320 = 520 \text{ cents}$$

Note that the total earnings include the $2 fixed fee and a $3.20 performance bonus.

In the unlikely event that we lose the connection to a Turker during the Experiment, the one-letter ID of that Turker will be removed in all subsequent rounds of the game

and you will be notified in the round when this happens. In the process of determining your bonus, we will not select any disconnected Turker after the round in which we lose the connection to that Turker.

It is possible to win an additional Bonus payment in the Final Questionnaire. We will explain this during the Final Questionnaire.

Please click the 'Next' link below to continue to the Tour.

Appendix B

Other Experiment Screenshots

We showcase the experimental interface of [Gallo and Yan, 2013b], which was conducted in June 2013. A draft of [Gallo and Yan, 2013b] is available upon request.

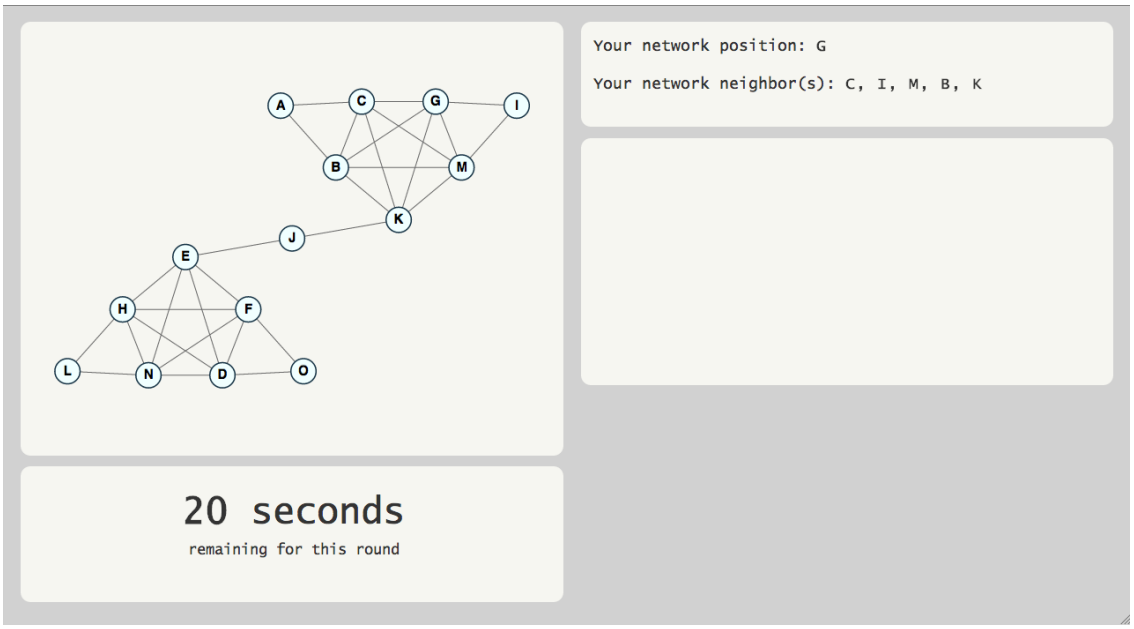


Figure B.1: At the start of each round, subject observes her random assignment to a node

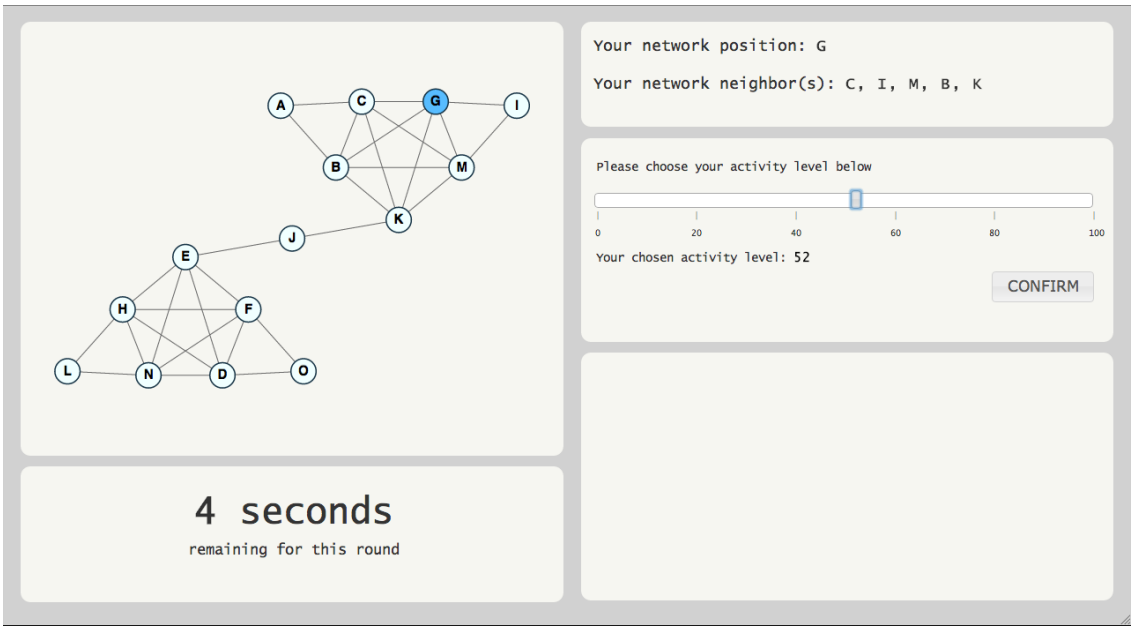


Figure B.2: After 5 seconds into each round, subject is able to make a choice by placing the bar on the scale

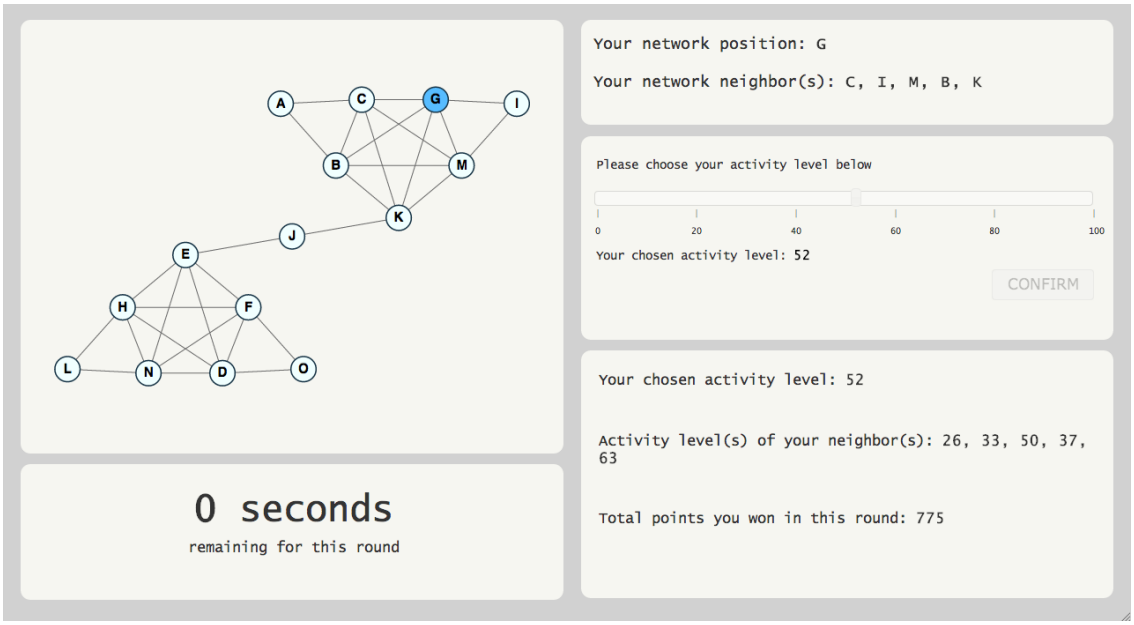


Figure B.3: After everyone inputs a choice, subject sees the result of the round

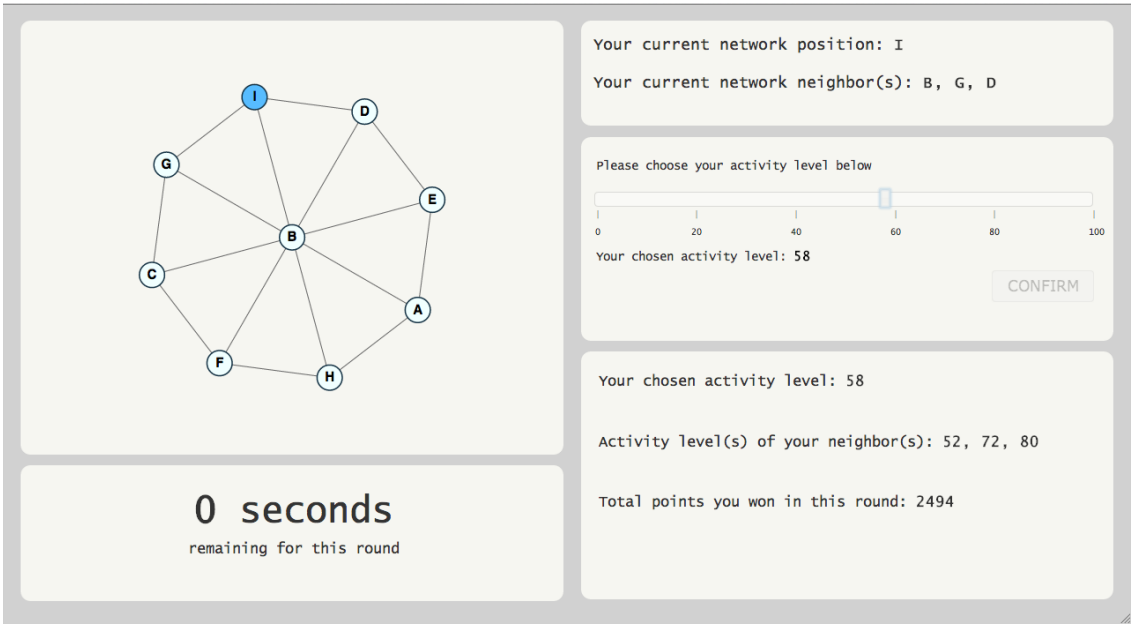


Figure B.4: Same experiment in a different network treatment

Appendix C

Implementation of Core Components for Synchronous Games

C.1 PlayGraph

```
from numpy import zeros, array_str, where
from random import shuffle
from itertools import compress

class PlayGraph(object):

    def __init__(self, dimension, directed):
        self.positions = []
        self.dimension = dimension
        self.graph = zeros((dimension, dimension), dtype=bool)
        self.directed = directed

    def create_graph(self, graph_strings):
        if len(graph_strings) == self.dimension:
            # graph_strings is an array of strings
            for idx, row_string in enumerate(graph_strings):
                # convert into int array
                ints = map(int, row_string)
                self.graph[idx] = ints
            return True
        else:
            return False
```

```
    # code continues from the previous page

    def get_details(self):
        details = {}
        details['positions'] = self.positions
        details['dimension'] = self.dimension
        details['graph'] = array_str(self.graph)
        return details

    def get_size(self):
        return self.dimension

    def get_graph(self):
        return self.graph

    def add_player(self, player):
        for idx, participant in enumerate(self.positions):
            if participant is None:
                self.positions[idx] = player
                return idx + 1
        if len(self.positions) < self.dimension:
            self.positions.append(player)
            return len(self.positions)
        else:
            return False

    def remove_player(self, player, live):
        try:
            idx = self.positions.index(player)
            if live:
                self.graph[idx, :] = False
                self.graph[:, idx] = False
            else:
                self.positions[idx] = None

            return True
        except ValueError as e:
            return False

    def shuffle_players(self):
        shuffle(self.positions)

    def get_players(self):
        return self.positions

    def connect(self, node1, node2):
        if not self.directed:
            self.graph[node1 - 1][node2 - 1] = 1
            self.graph[node2 - 1][node1 - 1] = 1
        else:
            self.graph[node1 - 1][node2 - 1] = 1
```

```
    # code continues from the previous page

    def disconnect(self, node1, node2):
        if not self.directed:
            self.graph[node1 - 1][node2 - 1] = 0
            self.graph[node2 - 1][node1 - 1] = 0
        else:
            self.graph[node1 - 1][node2 - 1] = 0

    def set_row_by_node(self, node, row):
        self.graph[node - 1] = row

    def get_row_by_node(self, node):
        return self.graph[node - 1]

    def get_player_by_node(self, node):
        return self.positions[node - 1]

    def get_node_by_player(self, player):
        return self.positions.index(player) + 1

    def get_connected_nodes(self, node):
        if self.directed:
            return where(self.graph[node - 1])[0] + 1
        else:
            return where(self.graph[node - 1] & \
            self.graph.transpose()[node - 1])[0] + 1

    def get_connected_players(self, node):
        if self.directed:
            return compress(self.positions, self.graph[node - 1])
        else:
            return (player for player, connection in zip(self.positions, \
            enumerate(self.graph[node - 1])) if connection[1] and \
            self.graph[connection[0]][node - 1])
```

C.2 ActionCache

```
from collections import defaultdict

class ActionCache(object):
    def __init__(self):
        self.action_cache = defaultdict(lambda: dict())
        self.player_index = defaultdict(list)
        self.round_index = defaultdict(list)
        self.stage_index = defaultdict(list)

    def add_action(self, session, curr_rnd, curr_stg, player, opp_group, action):
        """
        Add an action by a player towards a given opponents group at a given
        stage and round

        """
        if curr_stg == 1:
            self.player_index[(session, player)].append(curr_rnd)
            self.round_index[(session, curr_rnd)].append(player)
        self.stage_index[(session, player, curr_rnd)].append(curr_stg)
        self.action_cache[(session, curr_rnd, curr_stg, player)][opp_group] = \
            action
    def get_player_latest_action(self, session, player):
        """
        Return the latest stage action by a given player of the game

        """
        latest_round = self.player_index[(session, player)][-1]
        latest_stage = self.stage_index[(session, player, latest_round)][-1]
        return self.action_cache[(session, latest_round, latest_stage, player)]

    def get_player_stage_actions(self, session, player, stage):
        """
        Return the stage action history (in ascending order of round) of a
        given player at a given stage of the game

        """
        return (self.action_cache[(session, rnd, stage, player)] for rnd in \
            sorted(self.player_index[(session, player)]))

    def get_round_stage_actions(self, session, rnd, stage):
        """
        Return the stage action history (in ascending order of player) of
        all players at a given stage of a given round in the game

        """
        return (self.action_cache[(session, rnd, stage, player)] for player in \
            sorted(self.round_index[(session, rnd)]))
```

```
    # code continues from the previous page

    def clear_session(self, session):
        """
        Erase all action data for a given session

        """
        for c_key in (k for k in self.action_cache.keys() if k[0] == session):
            del self.action_cache[c_key]
        for p_key in (k for k in self.player_index.keys() if k[0] == session):
            del self.player_index[p_key]
        for r_key in (k for k in self.round_index.keys() if k[0] == session):
            del self.round_index[r_key]
        for s_key in (k for k in self.stage_index.keys() if k[0] == session):
            del self.stage_index[s_key]
```

Bibliography

[Ackerloff, 1970] Ackerloff, G. (1970). The market for lemons: Quality uncertainty and the market mechanism. *Quarterly Journal of Economics*, 84(3):488–500.

[Adler and De Alfaro, 2007] Adler, B. and De Alfaro, L. (2007). A content-driven reputation system for the wikipedia. In *Proceedings Of the 16th International Conference on World Wide Web*.

[Alexander, 1987] Alexander, R. D. (1987). *The biology of moral systems*. Transaction Publishers.

[Altman and Tennenholtz, 2010] Altman, A. and Tennenholtz, M. (2010). An axiomatic approach to personalized ranking systems. *Journal of the ACM (JACM)*, 57(4):26.

[Amir et al., 2012] Amir, O., Rand, D., and Gal, Y. (2012). Economic games on the internet: The effect of $1 stakes. *PloS ONE*, 7(2):e31461.

[Anderson et al., 2002] Anderson, D. P., Cobb, J., Korpela, E., Lebofsky, M., and Werthimer, D. (2002). Seti@ home: an experiment in public-resource computing. *Communications of the ACM*, 45(11):56–61.

[Andreoni and Croson, 2008] Andreoni, J. and Croson, R. (2008). *Handbook of Experimental Economics Results*, volume 1, chapter Partners versus strangers: Random rematching in public goods experiments, pages 776–783. Elsevier.

[Antonellis et al., 2008] Antonellis, I., Garcia-Molina, H., and Chang, C. (2008). Simrank++: query rewriting through link analysis of the clickgraph. In *Proceedings of the 17th International Conference on World Wide Web*, volume 1.

[Bell et al., 2013] Bell, S., Upchurch, P., Snavely, N., and Bala, K. (2013). Opensurfaces: A richly annotated catalog of surface appearance. In *SIGGRAPH Conf. Proc.*, volume 32.

[Berinsky et al., 2012] Berinsky, A., Huber, G., Lenz, G., et al. (2012). Evaluating online labor markets for experimental research: Amazon. com's mechanical turk. *Political Analysis*.

[Blondel et al., 2008] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

[Bonacich, 1972] Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113–120.

[Bostock, 2011] Bostock, M. (2011). `http://d3js.org`.

[Bostock et al., 2011] Bostock, M., Ogievetsky, V., and Heer, J. (2011). D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309.

[Buhrmester et al., 2011] Buhrmester, M., Kwang, T., and Gosling, S. (2011). Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5.

[Carpenter et al., 2005] Carpenter, J. P., Harrison, G. W., and List, J. A. (2005). *Field experiments in economics*, volume 10 of *Research in Experimental Economics*. Elsevier JAI.

[Carvalho and Larson, 2011] Carvalho, A. and Larson, K. (2011). A truth serum for sharing rewards. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, volume 11.

[Cassar, 2007] Cassar, A. (2007). Coordination and cooperation in local, random and small world networks: Experimental evidence. *Games and Economic Behavior*, 58(2):209–230.

[Charness et al., 2012] Charness, G., Gneezy, U., and Kuhn, M. A. (2012). Experimental methods: Between-subject and within-subject design. *Journal of Economic Behavior & Organization*, 81(1):1–8.

[Chen et al., 2010] Chen, L., Nayak, R., and Xu, Y. (2010). Improving matching process in social network. In *Proceedings of IEEE International Conference on Data Mining Workshops*.

[Chen et al., 2012] Chen, X., Schick, A., Doebeli, M., Blachford, A., and Wang, L. (2012). Reputation-based conditional interaction supports cooperation in well-mixed prisoner's dilemmas. *PloS one*, 7(5):e36260.

[Cheng and Friedman, 2005] Cheng, A. and Friedman, E. (2005). Sybilproof reputation mechanisms. In *Proceedings of ACM SIGCOMM Workshop on Economics of Peer-to-peer Systems*.

[Coenen et al., 2013] Coenen, A., Markant, D., Martin, J., and McDonnell, J. (2013). Using mechanical turk and psiturk for dynamic web experiments. In *Proceedings of the annual meeting of the cognitive science society*.

[Cong et al., 2012] Cong, R., Wu, B., Qiu, Y., and Wang, L. (2012). Evolution of cooperation driven by reputation-based migration. *PloS one*, 7(5):e35776.

[Cooper et al., 2010] Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., Popović, Z., et al. (2010). Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760.

[Cuesta et al., 2014] Cuesta, J. A., Gracia-Lázaro, C., Ferrer, A., Moreno, Y., and Sánchez, A. (2014). Reputation drives cooperative behaviour and network formation in human groups. *In Submission*.

[Dawes, 1989] Dawes, R. M. (1989). Statistical criteria for establishing a truly false consensus effect. *Journal of Experimental Social Psychology*, 25(1):1–17.

[Desrosiers and Karypis, 2010] Desrosiers, C. and Karypis, G. (2010). A novel approach to compute similarities and its application to item recommendation. *PRICAI 2010*.

[Dixit and Nalebuff, 2010] Dixit, A. K. and Nalebuff, B. J. (2010). *The Art of Strategy: A Game Theorist's Guide to Success in Business and Life*. W. W. Norton & Company.

[Economist, 2012] Economist, T. (2012). The roar of the crowd. *The Economist*.

[Enemark et al., 2014] Enemark, D., McCubbins, M. D., and Weller, N. (2014). Knowledge and networks: An experimental test of how network knowledge affects coordination. *Social Networks*, 36:122–133.

[Engelmann and Fischbacher, 2009] Engelmann, D. and Fischbacher, U. (2009). Indirect reciprocity and strategic reputation building in an experimental helping game. *Games and Economic Behavior*, 67(2):399–407.

[Facebook Inc, 2009] Facebook Inc (2009). `http://www.tornadoweb.org`.

[Fehl et al., 2011] Fehl, K., van der Post, D. J., and Semmann, D. (2011). Co-evolution of behaviour and social network structure promotes human cooperation. *Ecology letters*, 14(6):546–551.

[Fehr and Fischbacher, 2004] Fehr, E. and Fischbacher, U. (2004). Social norms and human cooperation. *Trends in Cognitive Sciences*, 8(4):185–190.

[Fette, 2011] Fette, I. (2011). `http://websocket.org`.

[Freeman, 1977] Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.

[Friedman, 1994] Friedman, D. (1994). *Experimental methods: A primer for economists*. Cambridge University Press.

[Friedman, 1971] Friedman, J. W. (1971). A non-cooperative equilibrium for supergames. *The Review of Economic Studies*, pages 1–12.

[Friedman, 1985] Friedman, J. W. (1985). Cooperative equilibria in finite horizon noncooperative supergames. *Journal of Economic Theory*, 35(2):390–398.

[Fu et al., 2008] Fu, F., Hauert, C., Nowak, M., and Wang, L. (2008). Reputation-based partner choice promotes cooperation in social networks. *Physical Review E*, 78(2):026117.

[Fudenberg and Maskin, 1986] Fudenberg, D. and Maskin, E. (1986). The folk theorem in repeated games with discounting or with incomplete information. *Econometrica*, pages 533–554.

[Gallo and Yan, 2013a] Gallo, E. and Yan (2013a). Synergistic effects of reputational and social knowledge on cooperation. *Submitted*.

[Gallo and Yan, 2013b] Gallo, E. and Yan, C. (2013b). Nash and the degree heuristic in network games: An online experiment. *Working paper*.

[Gracia-Lázaro et al., 2012] Gracia-Lázaro, C., Ferrer, A., Ruiz, G., Tarancón, A., Cuesta, J., Sánchez, A., and Moreno, Y. (2012). Heterogeneous networks do not promote cooperation when humans play a prisoner's dilemma. *Proceedings of the National Academy of Sciences*, 109(32):12922–12926.

[Grujić et al., 2012] Grujić, J., Eke, B., Cabrales, A., Cuesta, J. A., and Sánchez, A. (2012). Three is a crowd in iterated prisoner's dilemmas: experimental evidence on reciprocal behavior. *Scientific reports*, 2.

[Grujić et al., 2010] Grujić, J., Fosco, C., Araujo, L., Cuesta, J., and Sánchez, A. (2010). Social experiments in the mesoscale: Humans playing a spatial prisoner's dilemma. *PloS one*, 5(11):e13749.

[Grujić et al., 2014] Grujić, J., Gracia-Lázaro, C., Milinski, M., Semmann, D., Traulsen, A., Cuesta, J. A., Moreno, Y., and Sánchez, A. (2014). A comparative analysis of spatial prisoner's dilemma experiments: Conditional cooperation and payoff irrelevance. *Scientific reports*, 4.

[Hamilton, 1964a] Hamilton, W. D. (1964a). The genetical evolution of social behaviour. i. *Journal of theoretical biology*, 7(1):1–16.

[Hamilton, 1964b] Hamilton, W. D. (1964b). The genetical evolution of social behaviour. ii. *Journal of theoretical biology*, 7(1):1–16.

[Hertwig and Ortmann, 2001] Hertwig, R. and Ortmann, A. (2001). Experimental practices in economics: A methodological challenge for psychologists? *Behavioral and Brain Sciences*, 24(03):383–403.

[Heymann and Garcia-Molina, 2011] Heymann, P. and Garcia-Molina, H. (2011). Turkalytics: analytics for human computation. In *Proceedings of the 20th international conference on World wide web*, pages 477–486. ACM.

[Hipp, 2000] Hipp, D. (2000). `http://www.sqlite.org/`.

[Hoffman et al., 2009] Hoffman, K., Zage, D., and Nita-Rotaru, C. (2009). A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, 42(1):1.

[Holt and Laury, 2002] Holt, C. A. and Laury, S. K. (2002). Risk aversion and incentive effects. *American economic review*, 92(5):1644–1655.

[Hopcroft and Sheldon, 2007] Hopcroft, J. and Sheldon, D. (2007). Manipulation-resistant reputations using hitting time. In *Algorithms and Models for the Web-Graph*, pages 68–81. Springer.

[Horton et al., 2011] Horton, J., Rand, D., and Zeckhauser, R. (2011). The online laboratory: Conducting experiments in a real labor market. *Experimental Economics*, 14(3):399–425.

[Howie et al., 2011] Howie, P., Wang, Y., and Tsai, J. (2011). Predicting new product adoption using bayesian truth serum. *Journal of Medical Marketing: Device, Diagnostic and Pharmacetical Marketing*, 11(1).

[Hu et al., 2006] Hu, N., Pavlou, P., and Zhang, J. (2006). Can online reviews reveal a product's true quality? In *Proceedings of the 7th ACM Conference on Electronic Commerce*.

[Ipeirotis, 2010a] Ipeirotis, P. (2010a). Demographics of mechanical turk. *Center for Digital Economy Research, NYU Stern School of Business, Working paper*.

[Ipeirotis, 2010b] Ipeirotis, P. G. (2010b). Analyzing the amazon mechanical turk marketplace. *XRDS*, 17(2):16–21.

[Ismail and Jøsang, 2002] Ismail, R. and Jøsang, A. (2002). The beta reputation system. In *Proceedings of the 15th Bled Conference on E-Commerce*.

[Jamali and Ester, 2010] Jamali, M. and Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*.

[Jeh and Widom, 2002] Jeh, G. and Widom, J. (2002). Simrank: a measure of structural-context similarity. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[Jia et al., 2010] Jia, X., Liu, H., Zou, L., He, J., Du, X., and Cai, Y. (2010). Local methods for estimating simrank score. In *Proceedings of APWEB'10*.

[Jøsang et al., 2007] Jøsang, A., Ismail, R., and Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644.

[Jsang and Ismail, 2002] Jsang, A. and Ismail, R. (2002). The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*.

[Jurca and Faltings, 2009] Jurca, R. and Faltings, B. (2009). Mechanisms for making crowds truthful. *Journal of Artificial Intelligence Research*, 34(1).

[Kamvar et al., 2003] Kamvar, S., Schlosser, M., and Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*.

[Khatib et al., 2011] Khatib, F., DiMaio, F., Cooper, S., Kazmierczyk, M., Gilski, M., Krzywda, S., Zabranska, H., Pichova, I., Thompson, J., Popović, Z., et al. (2011). Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature structural & molecular biology*, 18(10):1175–1177.

[Kirchkamp and Nagel, 2007] Kirchkamp, O. and Nagel, R. (2007). Naive learning and cooperation in network experiments. *Games and Economic Behavior*, 58(2):269–292.

[Kosfeld,] Kosfeld, M. Economic networks in the laboratory: A survey. *Review of Network Economics*, 3(1).

[Kreps et al., 1982] Kreps, D., Milgrom, P., Roberts, J., and Wilson, R. (1982). Rational cooperation in the finitely repeated prisoners' dilemma. *Journal of Economic theory*, 27(2):245–252.

[Lambert and Shoham, 2009] Lambert, N. and Shoham, Y. (2009). Eliciting truthful answers to multiple-choice questions. In *Proceedings of the 10th ACM Conference on Electronic Commerce*.

[Larson et al., 2002] Larson, S. M., Snow, C. D., Shirts, M. R., and Pande, V. S. (2002). Folding@ home and genome@ home: Using distributed computing to tackle previously intractable problems in computational biology. *Computational Genomics*.

[Le Boudec et al., 2004] Le Boudec, J.-Y., Buchegger, S., et al. (2004). A robust reputation system for peer-to-peer and mobile ad-hoc networks. In *P2PEcon 2004*.

[Leskovec et al., 2009] Leskovec, J., Backstrom, L., and Kleinberg, J. (2009). Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506. ACM.

[Li et al., 2009] Li, P., Cai, Y., Liu, H., He, J., and Du, X. (2009). Exploiting the block structure of link graph for efficient similarity computation. *Advances in Knowledge Discovery and Data Mining*.

[Li et al., 2010] Li, P., Liu, H., Xu, J., Jun, Y., and Du, H. (2010). Fast single-pair simrank computation. In *Proceeding of the 2010 SIAM International Conference on Data Mining*.

[Lieberman et al., 2005] Lieberman, E., Hauert, C., and Nowak, M. A. (2005). Evolutionary dynamics on graphs. *Nature*, 433(7023):312–316.

[Little et al., 2009] Little, G., Chilton, L., Goldman, M., and Miller, R. (2009). Turkit: tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 29–30. ACM.

[Lizorkin et al., 2010] Lizorkin, D., Velikhov, P., Grinev, M., and Turdakov, D. (2010). Accuracy estimate and optimization techniques for simrank computation. *The VLDB Journal*, 19(1).

[Ma et al., 2011] Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. (2011). Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*.

[Mailath and Samuelson, 2006] Mailath, G. J. and Samuelson, L. (2006). *Repeated Games and Reputations: Long-Run Relationships*. Oxford University Press, USA.

[Majkowski, 2011] Majkowski, M. (2011). `http://sockjs.org`.

[Mao et al., 2012] Mao, A., Chen, Y., Gajos, K. Z., Parkes, D., Procaccia, A. D., and Zhang, H. (2012). Turkserver: Enabling synchronous and longitudinal online experiments. In *Proceedings of HCOMP 12*.

[Marks and Miller, 1987] Marks, G. and Miller, N. (1987). Ten years of research on the false-consensus effect: an empirical and theoretical review. *Psychological Bulletin*, 102(1):72.

[Marti and Garcia-Molina, 2006] Marti, S. and Garcia-Molina, H. (2006). Taxonomy of trust: Categorizing p2p reputation systems. *Computer Networks*, 50(4):472–484.

[Mason and Suri, 2012] Mason, W. and Suri, S. (2012). Conducting behavioral research on Amazon Mechanical Turk. *Behavior research methods*, 44(1):1–23.

[Mason and Watts, 2012] Mason, W. and Watts, D. J. (2012). Collaborative learning in networks. *Proceedings of the National Academy of Sciences*, 109(3):764–769.

[McKinney, 2011] McKinney, W. (2011). `http://pandas.pydata.org`.

[Milinski et al., 2001] Milinski, M., Semmann, D., Bakker, T., and Krambeck, H. (2001). Cooperation through indirect reciprocity: image scoring or standing strategy? *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1484):2495–2501.

[Milinski et al., 2002] Milinski, M., Semmann, D., and Krambeck, H.-J. (2002). Reputation helps solve the tragedy of the commons. *Nature*, 415(6870):424–426.

[Miller et al., 2005] Miller, N., Resnick, P., and Zeckhauser, R. (2005). Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9).

[Nash, 1950] Nash, J. F. (1950). Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49.

[Nowak and Sigmund, 1998] Nowak, M. and Sigmund, K. (1998). Evolution of indirect reciprocity by image scoring. *Nature*, 393(6685):573–577.

[Nowak, 2006] Nowak, M. A. (2006). Five rules for the evolution of cooperation. *science*, 314(5805):1560–1563.

[Nowak and May, 1992] Nowak, M. A. and May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829.

[Ntoulas et al., 2004] Ntoulas, A., Cho, J., and Olston, C. (2004). What's new on the web?: the evolution of the web from a search engine perspective. In *Proceedings of the 13th international conference on World Wide Web*, pages 1–12. ACM.

[Ohtsuki et al., 2006] Ohtsuki, H., Hauert, C., Lieberman, E., and Nowak, M. A. (2006). A simple rule for the evolution of cooperation on graphs and social networks. *Nature*, 441(7092):502–505.

[Oliphant, 2007] Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science and Engineering*, 9:10–20.

[Pacheco et al., 2006] Pacheco, J. M., Traulsen, A., and Nowak, M. A. (2006). Active linking in evolutionary games. *Journal of theoretical biology*, 243(3):437–443.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project.

[Paolacci et al., 2010] Paolacci, G., Chandler, J., and Ipeirotis, P. (2010). Running experiments on amazon mechanical turk. *Judgment and Decision Making*, 5(5):411–419.

[Perc and Szolnoki, 2010] Perc, M. and Szolnoki, A. (2010). Coevolutionary games a mini review. *BioSystems*, 99(2):109–125.

[Prelec, 2004] Prelec, D. (2004). A bayesian truth serum for subjective data. *Science*, 306(5695).

[Qualtrics, 2013] Qualtrics (2013). `http://www.qualtrics.com`.

[Radanovic and Faltings, 2013] Radanovic, G. and Faltings, B. (2013). A robust bayesian truth serum for non-binary signals. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*.

[Rand, 2011] Rand, D. (2011). The promise of mechanical turk: How online labor markets can help theorists run behavioral experiments. *Journal of theoretical biology*.

[Rand et al., 2011] Rand, D., Arbesman, S., and Christakis, N. (2011). Dynamic social networks promote cooperation in experiments with humans. *Proceedings of the National Academy of Sciences*, 108(48):19193–19198.

[Rand et al., 2012] Rand, D. G., Greene, J. D., and Nowak, M. A. (2012). Spontaneous giving and calculated greed. *Nature*, 489(7416):427–430.

[Rand and Nowak, 2013] Rand, D. G. and Nowak, M. A. (2013). Human cooperation. *Trends in Cognitive Sciences*, 17(8):413.

[Rand et al., 2013] Rand, D. G., Tarnita, C. E., Ohtsuki, H., and Nowak, M. A. (2013). Evolution of fairness in the one-shot anonymous ultimatum game. *Proceedings of the National Academy of Sciences*, 110(7):2581–2586.

[Reips, 2000a] Reips, U.-D. (2000a). 13 theory and techniques of conducting web experiments. *Online social sciences*, page 243.

[Reips, 2000b] Reips, U.-D. (2000b). The web experiment method: Advantages, disadvantages, and solutions. *Psychological experiments on the Internet*, pages 89–117.

[Reips, 2002] Reips, U.-D. (2002). Standards for internet-based experimenting. *Experimental Psychology*, 49(4):243–256.

[Ross et al., 2010] Ross, J., Irani, L., Silberman, M., Zaldivar, A., and Tomlinson, B. (2010). Who are the crowdworkers?: shifting demographics in mechanical turk. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, pages 2863–2872. ACM.

[Saavedra et al., 2009] Saavedra, S., Reed-Tsochas, F., and Uzzi, B. (2009). A simple model of bipartite cooperation for ecological and organizational networks. *Nature*, 457(7228):463–466.

[Sanfilippo and Noordhuis, 2009] Sanfilippo, S. and Noordhuis, P. (2009). `http://redis.io`.

[Santos and Pacheco, 2005] Santos, F. and Pacheco, J. (2005). Scale-free networks provide a unifying framework for the emergence of cooperation. *Physical Review Letters*, 95(9):98104.

[Santos and Pacheco, 2006] Santos, F. C. and Pacheco, J. M. (2006). A new route to the evolution of cooperation. *Journal of Evolutionary Biology*, 19(3):726–733.

[Santos et al., 2006] Santos, F. C., Pacheco, J. M., and Lenaerts, T. (2006). Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *Proceedings of the National Academy of Sciences of the United States of America*, 103(9):3490–3494.

[Santos et al., 2008] Santos, F. C., Santos, M. D., and Pacheco, J. M. (2008). Social diversity promotes the emergence of cooperation in public goods games. *Nature*, 454(7201):213–216.

[Savage, 1971] Savage, L. J. (1971). Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801.

[Schram, 2005] Schram, A. (2005). Artificiality: The tension between internal and external validity in economic experiments. *Journal of Economic Methodology*, 12(2):225–237.

[Segal, 2011] Segal, D. (2011). A rave, a pan, or just a fake. *The New York Times*.

[Sharma and Cosley, 2013] Sharma, A. and Cosley, D. (2013). Do social explanations work?: studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1133–1144. International World Wide Web Conferences.

[Shaw et al., 2011] Shaw, A., Horton, J., and Chen, D. (2011). Designing incentives for inexpert human raters. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*.

[Shirado et al., 2013] Shirado, H., Fu, F., Fowler, J. H., and Christakis, N. A. (2013). Quality versus quantity of social ties in experimental cooperative networks. *Nature communications*, 4.

[Smith, 1982] Smith, V. L. (1982). Microeconomic systems as an experimental science. *American Economic Review*, 72(5):923–955.

[Streitfeld, 2012] Streitfeld, D. (2012). For $2 a star, an online retailer gets 5-star product reviews. *The New York Times*.

[Suri and Watts, 2011] Suri, S. and Watts, D. (2011). Cooperation and contagion in web-based, networked public goods experiments. *PLoS ONE*, 6(3):e16836.

[Sysoev, 2002] Sysoev, I. (2002). `http://nginx.org`.

[Tarreau, 2001] Tarreau, W. (2001). `http://haproxy.1wt.eu`.

[Taylor et al., 2007] Taylor, P. D., Day, T., and Wild, G. (2007). Evolution of cooperation in a finite homogeneous graph. *Nature*, 447(7143):469–472.

[Teacy et al., 2006] Teacy, W. L., Patel, J., Jennings, N. R., and Luck, M. (2006). Travos: Trust and reputation in the context of inaccurate information sources. In *Proceedings of Autonomous Agents and Multi-Agent Systems*, volume 12, pages 183–198.

[Traulsen and Nowak, 2006] Traulsen, A. and Nowak, M. A. (2006). Evolution of cooperation by multilevel selection. *Proceedings of the National Academy of Sciences*, 103(29):10952–10955.

[Traulsen et al., 2010] Traulsen, A., Semmann, D., Sommerfeld, R., Krambeck, H., and Milinski, M. (2010). Human strategy updating in evolutionary games. *Proceedings of the National Academy of Sciences*, 107(7):2962–2966.

[Trivers, 1971] Trivers, R. L. (1971). The evolution of reciprocal altruism. *Quarterly review of biology*, pages 35–57.

[Von Ahn, 2006] Von Ahn, L. (2006). Games with a purpose. *Computer*, 39(6):92–94.

[Von Ahn and Dabbish, 2004] Von Ahn, L. and Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM.

[Walsh and Sirer, 2005] Walsh, K. and Sirer, E. G. (2005). Fighting peer-to-peer spam and decoys with object reputation. In *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 138–143. ACM.

[Wang et al., 2012] Wang, J., Suri, S., and Watts, D. (2012). Cooperation and assortativity with dynamic partner updating. *Proceedings of the National Academy of Sciences*, 109(36):14363–14368.

[Watts and Strogatz, 1998] Watts, D. and Strogatz, S. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

[Weiss, 2009] Weiss, R. (2009). Optimally aggregating elicited expertise: a proposed application of the bayesian truth serum for policy analysis. Master's thesis, MIT.

[Whitby et al., 2005] Whitby, A., Jøsang, A., and Indulska, J. (2005). Filtering out unfair ratings in bayesian reputation systems. *The Icfain Journal of Management Research*.

[Wilson, 1975] Wilson, D. S. (1975). A theory of group selection. *Proceedings of the national academy of sciences*, 72(1):143–146.

[Witkowski and Parkes, 2011] Witkowski, J. and Parkes, D. C. (2011). Peer prediction with private beliefs. In *Proceedings of the 1st Workshop on Social Computing and User Generated Content*.

[Witkowski and Parkes, 2012a] Witkowski, J. and Parkes, D. C. (2012a). Peer prediction without a common prior. In *Proceedings of the 13th ACM Conference on Electronic Commerce*.

[Witkowski and Parkes, 2012b] Witkowski, J. and Parkes, D. C. (2012b). A robust bayesian truth serum for small populations. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*.

[Xi et al., 2005] Xi, W., Fox, E., Fan, W., Zhang, B., Chen, Z., Yan, J., and Zhuang, D. (2005). Simfusion: measuring similarity using unified relationship matrix. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*.

[Xu et al., 2007] Xu, X., Yuruk, N., Feng, Z., and Schweiger, T. (2007). Scan: A structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[Yang et al., 2012] Yang, X., Steck, H., and Liu, Y. (2012). Circle-based recommendation in online social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*.

[Yelp Inc, 2012] Yelp Inc (2012). SEC S-1/A filing. U.S. Securities and Exchange Commission.

[Yu and Singh, 2003] Yu, B. and Singh, M. P. (2003). Detecting deception in reputation management. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 73–80. ACM.