University of California
Los Angeles

Networks of Strategic Agents: Social Norms, Incentives and Learning

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Jie Xu

2015

Abstract of the Dissertation

Networks of Strategic Agents: Social Norms, Incentives and Learning

by

Jie Xu

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2015

Professor Mihaela van der Schaar, Chair

Much of society is organized in networks: autonomous communication networks, social networks, economic networks. However, to enable the efficient and robust operation of networks several key challenges need to be overcome: the interacting agents (people, devices, software, companies, etc.) are strategic, heterogeneous and have incomplete information about the other agents. This dissertation develops systematic solutions to address these challenges.

The first part of this dissertation studies how to incentivize self-interested agents to take socially optimal actions. In many service exchange networks, agents connect to other agents to request services (e.g. favors, goods, information etc.); however, since agents who provide service gain no (immediate) benefit but only incur costs, they have an incentive to withhold their service. This dissertation designs and analyzes incentives mechanisms that rely on various types of social reciprocation, including exchange of fiat money and rating systems. The analysis builds on the theory of repeated and stochastic games with imperfect monitoring, but requires significant innovations to address the unique characteristics and requirements of online communities and networks: the anonymity and heterogeneity of agents, informational constraints (for both agents and the network manager), real-time constraints, network topology constraints, etc.

The second part of this dissertation studies how agents learn in networks. In many networks, agents need to learn how to cooperate with each other to achieve a common goal.

This dissertation designs the first multi-agent learning algorithm that is able to achieve cooperation without requiring any explicit message exchange with other agents and to provide performance guarantees, including characterizing the speed of convergence.

A final part of the dissertation aims to address the problem of adverse selection in networks. The goal is to design and analyze reputation-based social norms that aim to eliminate agents of low qualities from participating in networks and communities. For this, a system of reputation in which agents reputation is determined based on their productivity when working alone or with others. If the agents reputation at the time of their evaluation (determined by the social norm) is higher than a quality/productivity level (determined by the social norm) they can remain in the network; otherwise they are expelled. The dissertation designs and analyzes social norms that maximize the productivity of the society.

The dissertation of Jie Xu is approved.

Ali H. Sayed

Paulo Tabuada

Ichiro Obara

Mihaela van der Schaar, Committee Chair

University of California, Los Angeles

2015

To my family

Table of Contents

List of Figures

List of Tables

Acknowledgments

First of all, I would like to thank my advisor, Prof. Mihaela van der Schaar, without whom this dissertation does not exist. Throughout the past five years at UCLA, she has been giving me invaluable guidance with her vision of research and passion for high-quality work. I am much indebted for her patience and tolerance of my mistakes and her generous support on research and beyond. Words are far away from expressing my appreciation.

I would like to thank other committee members, Prof. Ali H. Sayed, Prof. Paolo Tabuada and Prof. Ichiro Obara for their time and efforts in evaluating my work. I would like to express my special thanks to Prof. William Zame. I have been very fortunate to work with and receive invaluable advice and continuous support from a great economist and mathematician.

This dissertation is a result of multiple collaborations with many amazing researchers and friends around the world. I am grateful to Dr. Deepak S. Turaga and Dr. Daby M. Sow for being my mentors during my internship in IBM T. J. Waston Research Center. I thank Dr. Yiannis Andreopoulos, Dr. Nicholas Mastronarde, Dr. Cong Shen, Dr. Jiangchuan Liu, Dr. Cyrus Shahabi, Dr. Ugur Demiryurek, Dr. William Hsu, Dr. Gregory Pottie and their students with whom the collaboration became much fun and productive.

I also would like to thank my colleagues and friends at UCLA: Dr. Jaeok Park, Dr. Shaolei Ren, Dr. Yu Zhang, Khoa Phan, Dr. Cem Tekin, Dr. Luca Canzian, Dr. Yuanzhang Xiao, Siming Song, Jianyu Wang, Linqi Song, Simpson Zhang, Darcy Song, Kartik Ahuja, Onur Atan, Basar Akbay and Ahmed Alaa. I learned and benefited a lot from the discussions and collaborations with them and my PhD life became much more colorful because of them.

Finally, I wish to express my deepest appreciation to my family for their love, support and encouragement. My parents are always a telephone call away even though they are on the other side of the world. I am incredibly grateful to my beloved wife, Shen Dong, with whom hard working becomes meaningful and joyful.

Vita

2008	B.E. (Electronics Engineering), Tsinghua University.
2010	M.S. (Electronics Engineering), Tsinghua University.
2010–present	Research Assistant, Electrical Engineering Department, UCLA.
2014	Research Intern, IBM T. J. Watson Research Center, Yorktown Heights, New York.

Publications

Jie Xu, C. Tekin and M. van der Schaar, "Distributed Multi-Agent Online Learning with Global Feedback," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp 2225-2238, Feb. 2015.

Jie Xu and M. van der Schaar, "Efficient Working and Shirking in Networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 4, pp. 651-662, Jan. 2015.

Jie Xu, M. van der Schaar, J. Liu and H. Li, "Forecasting Popularity of Videos using Social Media," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 2, pp. 330-343, Nov. 2014.

Jie Xu, Y. Song, and M. van der Schaar, "Sharing in Networks of Strategic Agents," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 717-731, Apr. 2014.

Jie Xu, Y. Andreopoulos, Y. Xiao and M. van der Schaar, "Non-stationary Resource Allocation Policies for Delay-constrained Video Streaming: Application to Video over Internet-of-Things-enabled Networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 782-794, Apr. 2014.

M. van der Schaar, Jie Xu, W. Zame (author names listed alphabetically), "Efficient Online Exchange via Fiat Money" *Economic Theory*, vol. 54, no. 2, pp. 211-248, Oct. 2013.

Jie Xu and M. van der Schaar, "Token System Design for Autonomic Wireless Relay Networks," *IEEE Transactions on Communications*, vol. 61, no. 7, pp. 2924-2935, Jun. 2013.

Jie Xu and M. van der Schaar, "Social Norm Design for Information Exchange Systems with Limited Observations," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 11, pp. 2126-2135, Dec. 2012.

W. Zame, Jie Xu and M. van der Schaar, "Cooperative Multi-Agent Learning and Coordination for Cognitive Radio Networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 3, pp. 464-477, Dec. 2013.

W. Zame, Jie Xu and M. van der Schaar, "Winning the Lottery: Learning Perfect Coordination with Minimal Feedback," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 846-857, Apr. 2013.

S. Zhou, Jie Xu, and Z. Niu, "Interference-Aware Relay Selection Scheme for Two-Hop Relay Networks with Multiple Source-Destination Pairs," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2327-2338, Jan. 2013.

Jie Xu, D. Deng, U. Demiryurek, C. Shahabi, and M. van der Schaar, "Mining the Situation: Spatiotemporal Traffic Prediction with Big Data," *IEEE Journal of Selected Topics in Signal*

Processing, to appear.

L. Song, W. Hsu, Jie Xu and M. van der Schaar, "Using Contextual Learning to Improve Diagnostic Accuracy: Application in Breast Cancer Screening," *IEEE Journal of Biomedical and Health Informatics*, to appear.

C. Shen, Jie Xu and M. van der Schaar, "Silence is Gold, Strategic Interference Mitigation Using Tokens in Heterogeneous Small Cell Networks," *IEEE Journal on Selected Areas in Communications*, to appear.

Jie Xu, M. van der Schaar, J. Liu and H. Li, "Timely Popularity Forecasting based on Social Networks," *IEEE INFOCOM*, 2015.

Jie Xu, D. Sow, D. Turaga and M. van der Schaar, "Online Transfer Learning for Differential Diagnosis Determination," *AAAI Workshop on the World Wide Web and Public Health Intelligence (W3PHI)*, 2015.

Jie Xu, D. Deng, U. Demiryurek, C. Shahabi, M. van der Schaar, "Context-Aware Online Spatiotemporal Traffic Prediction," *IEEE ICDM Workshop on Spatial and Spatio-Temporal Data Mining*, 2015.

Jie Xu, S. Zhang, M. van der Schaar, "Network Evolution with Incomplete Information and Learning," *Allerton Conference on Control, Communication and Computing*, 2014.

Jie Xu, J. Y. Xu, L. Song, G. Pottie and M. van der Schaar, "Context-Driven Online Learning for Activity Classification in Wireless Health," *IEEE Global Communications Conference (GLOBECOM)*, 2014.

C. Shen, Jie Xu, and M. van der Schaar, "Silence is Gold: Strategic Small Cell Inter-

ference Management Using Tokens," *IEEE Global Communications Conference (GLOBECOM)*, 2014.

Jie Xu, Y. Song and M. van der Schaar, "Incentivizing Information Sharing in Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

Jie Xu, C. Tekin and M. van der Schaar, "Learning Optimal Classifier Chains for Real-time Big Data Mining," *Allerton Conference on Control, Communication and Computing*, 2013.

N. Mastrondarde, V. Patel, Jie Xu, and M. van der Schaar, "Learning Relaying Strategies in Cellular D2D Networks with Token-Based Incentives," *IEEE GLOBECOM Workshop on Emerging Technologies for LTE-Advanced and Beyong-4G*, 2013.

W. Zame, Jie Xu, and M. van der Schaar, "Learning Perfect Coordination with Minimal Feedback in Wireless Multi-Access Communications," *IEEE Global Communications Conference (GLOBECOM)*, 2013.

Jie Xu, Y. Zhang, and M. van der Schaar, "Rating Systems for Enhanced Cyber-Security Investments," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

Jie Xu and M. van der Schaar, "Incentive Design for Heterogeneous User-Generated Content Networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 4, Mar. 2014.

S. Won, I. Cho, K. Sudusinghe, Jie Xu, Y. Zhang, M. van der Schaar and S. S. Bhattacharyya, "A Design Methodology for Distributed Adaptive Stream Mining Systems," *International Conference on Computational Science*, 2013.

Jie Xu and M. van der Schaar, "Sustaining Cooperation in Social Exchange Networks with

Incomplete Global Information," *IEEE Annual Conference on Decision and Control (CDC)*, 2012.

Jie Xu, W. Zame, and M. van der Schaar, "Token Economy for Online Exchange Systems," *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.

Jie Xu, W. Zame and M. van der Schaar, "Token-based Incentive Protocol Design for Online Exchange Systems," *International Conference on Game Theory for Networks (GameNets)*, 2012.

Jie Xu and M. van der Schaar, "Designing Incentives for Wireless Relay Networks using Tokens," *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2012.

Jie Xu, Y. Wu, Z. Niu, "Exploiting Multiuser Diversity in OFDMA Wireless Mesh Networks by Fractional Spatial Reuse," *IEEE International Conference on Communications (ICC)*, 2010.

Jie Xu, S. Zhou, Z. Niu, "Interference-Aware Relay Selection for Multiple Source-Destination Cooperative Networks," *Asia-Pacific Conference on Communications (APCC)*, 2009.

CHAPTER 1

Introduction

Much of society is organized in networks and the interacting agents in many of these networks are strategic - they are self-interested and have learning abilities. Examples range from autonomous communication networks in engineering to social networks and economic networks. To enable the efficient and robust operation of such networks, several key challenges need to be overcome: the interacting agents (people, devices, software, companies, etc.) are strategic, heterogenous and have incomplete information about the other agents. This dissertation develops systematic solutions to address these challenges.

The dissertation consists of three parts. In the first part (Chapter 2 and Chapter 3), I answer the question of how to incentivize self-interested agents to take socially optimal actions. In the second part (Chapter 4 and Chapter 5), I develop efficient learning algorithms for agents to learn in networks, thereby improving the network performance. In the final part (Chapter 6), I solve the adverse selection problem of networks by designing appropriate social norm mechanisms. In what follows, I provide the motivation and summaries for each parts of this dissertation.

1.1 Part I: Incentives

1.1.1 Chapter 2: Efficient Online Exchange via Fiat Money

In many online systems, individuals provide services for each other; the recipient of the service obtains a benefit but the provider of the service incurs a cost. If benefit exceeds cost, provision of the service increases social welfare and should therefore be encouraged – but the individuals providing the service gain no (immediate) benefit from providing the service and

hence have an incentive to withhold service. Hence there is scope for designing a protocol that improves welfare by encouraging exchange. To operate successfully within the confines of the online environment, such a protocol should be distributed, robust, and consistent with individual incentives. Chapter 2 proposes and analyzes protocols that rely solely on the exchange of *fiat money* or *tokens*. The analysis has much in common with work on search models of money but the requirements of the environment also lead to many differences from previous analyses – and some surprises; in particular, existence of equilibrium becomes a thorny problem and the optimal quantity of money is different.

1.1.2 Chapter 3: Sharing in Networks of Strategic Agents

In Chapter 3, I study the incentive problem in environments where individuals interact subject to topological constraints. I design *distributed* rating protocols which exploit the ongoing nature of the agents' interactions to assign ratings and through them, determine future rewards and punishments: agents that have behaved as directed enjoy high ratings – and hence greater future access to the information/goods of others; agents that have not behaved as directed enjoy low ratings – and hence less future access to the information/goods of others. Unlike existing rating protocols, the proposed protocol operates in a distributed manner and takes into consideration the underlying interconnectivity of agents as well as their heterogeneity. I prove that in many networks, the price of anarchy (PoA) obtained by adopting the proposed rating protocols is 1, that is, the optimal social welfare is attained. In networks where PoA is larger than 1, I show that the proposed rating protocol significantly outperforms existing incentive mechanisms. Last but not least, the proposed rating protocols can also operate efficiently in dynamic networks, where new agents enter the network over time.

1.2 Part II: Learning

1.2.1 Chapter 4: Distributed Multi-Agent Online Learning

In Chapter 4, I develop online learning algorithms which enable the agents to cooperatively learn how to maximize the overall reward in scenarios where only noisy global feedback is available without exchanging any information among themselves. I prove that our algorithms' learning regrets - the losses incurred by the algorithms due to uncertainty - are logarithmically increasing in time and thus the time average reward converges to the optimal average reward. Moreover, I also illustrate how the regret depends on the size of the action space, and I show that this relationship is influenced by the informativeness of the reward structure with regard to each agent's individual action. When the overall reward is fully informative, regret is shown to be linear in the total number of actions of all the agents. When the reward function is not informative, regret is linear in the number of joint actions. Our analytic and numerical results show that the proposed learning algorithms significantly outperform existing online learning solutions in terms of regret and learning speed. I illustrate how our theoretical framework can be used in practice by applying it to online Big Data mining using distributed classifiers.

1.2.2 Chapter 5: Content Popularity Forecasting using Social Media

Chapter 5 presents a contextual learning algorithm applied to a content popularity forecasting problem using social media. The propose algorithm, called Social-Forecast, explicitly considers the dynamically changing and evolving propagation patterns of videos in social media when making popularity forecasts, thereby being situation and context aware. Social-Forecast aims to maximize the forecast reward, which is defined as a tradeoff between the popularity prediction accuracy and the timeliness with which a prediction is issued. The forecasting is performed online and requires no training phase or a priori knowledge. I analytically bound the prediction performance loss of Social-Forecast as compared to that obtained by an omniscient oracle and prove that the bound is sublinear in the number of

video arrivals, thereby guaranteeing its short-term performance as well as its asymptotic convergence to the optimal performance. In addition, I conduct extensive experiments using real-world data traces collected from the videos shared in RenRen, one of the largest online social networks in China. These experiments show that our proposed method outperforms existing view-based approaches for popularity prediction (which are not context-aware) by more than 30% in terms of prediction rewards.

1.3 Part III: Social Norms

1.3.1 Chapter 6: The Design and Dynamics of Up-or-Out Evaluation

"Up-or-out" evaluation is common in many professions. New hires are given a trial period in which to establish their value. If their performance makes the cut, they move up to a permanent position; otherwise, they move out. Such evaluation systems allow organizations to control the number and quality of their members. In Chapter 6, I model the population dynamics of organizations under up-or-out evaluation. Agents of varying quality arrive and exit stochastically. They produce output individually and collaboratively; the rate depends on their individual and collective quality, subject to noise. An evaluation system is characterized by the time-to-evaluation and the minimum performance level; agents who do not meet the minimum level are eliminated. I prove that an evaluation system has at least one steady state. Given a desired organization size, I show how to design evaluation systems to maximize total productivity. The optimal time-to-evaluation is set by the noise level, with higher noise requiring longer trial periods so that robust differences in quality can emerge. I also prove that more intense collaboration decreases average quality, as it screens individual quality from assessment. I illustrate these results through simulation, and show how they extend to models that include cumulative advantage.

CHAPTER 2

Efficient Online Exchange via Fiat Money

This chapter is motivated by the problem of online exchange of files (or data or services). In typical systems that serve this purpose – Napster, now defunct, is the most familiar example but there are many in current operation, including Gnutella and Kazaa (file sharing), Seti@home (computational assistance), Slashdot and Yahoo Answers (answers to queries) – a single interaction involves an agent who wants a file (or data or service) and an agent who can provide it. The former benefits from obtaining the file but the latter bears the (often non-trivial) cost of providing it and so has an incentive to free-ride.[1] Assuming that benefit exceeds the cost, provision of the service increases social welfare and should therefore be encouraged – but how?

This problem is a particular instance of trade in the absence of a double coincidence of wants, which has motivated a large literature on search models of money. Indeed, we shall formalize our problem in the same terms, and the "solution" we develop is for a (benevolent) designer to institute a system that relies on *fiat money* or *tokens* (we use the terms interchangeably), to introduce a quantity of tokens into the system and to recommend strategies to the participants for requesting and providing service. Because our agents are self-interested, the designer must recommend strategies that constitute an equilibrium – but our environment also imposes other constraints on the designer: the system must be anonymous and distributed, must take account of the fact that agents meet only electronically (and not face-to-face), that files and tokens are indivisible, that the designer cannot know the precise parameters of the population and, perhaps most importantly, that the designer

[1] Empirical studies show that this free-riding problem can be quite severe: in the Gnutella system for instance, almost 70% of users share no files at all (Adar & Hubeman, 2000) [AH00].

cannot constrain the number of tokens that agents hold.[2,3]

This chapter asks how much a designer can accomplish, given these constraints, by judicious choice of the *protocol* – the quantity of tokens and the recommended strategies. To answer this question we characterize equilibrium protocols and among these, the ones that are robust to small perturbations of the population parameters (the designer's slight misperceptions of these parameters); we prove that robust equilibrium protocols exist; we provide bounds for the efficiency of robust equilibrium protocols; we show that the "optimal quantity of money" in our setting is different than in other settings considered in the literature; we provide an *effective* procedure for choosing a robust equilibrium protocol whose efficiency is at least good (if not optimal); and we provide numerical simulations to illustrate some of the theorems and also to demonstrate that *design matters*: a great deal of efficiency may be lost if the designer chooses the "wrong" protocol.

As in the familiar search models of money, our environment is populated by a continuum of agents each of whom is initially endowed with a unique file that can be duplicated and provided to others.[4] In each period, a fraction of the population is matched; one member of each match – the *client* – must decide whether to request service (provision of a file or forwarding of a packet) and the other – the *server* – must decide whether to provide the service (if requested). The client who receives the service derives a benefit, the server who provides the service incurs a cost. To simplify the analysis we assume here that, except for the uniqueness of the files they possess, all agents are identical, and that all files are equally valuable to receive and equally costly to provide. (We discuss extensions in the Conclusion.)

[2]It might be useful to note that none of the systems mentioned above involve a central authority or central monitoring agency. Napster, for instance, merely maintained many partial lists (distributed across many servers) of music files available and contact information for subscribers who had these files; users seeking files could simply search these lists and then contact the file-holder directly. In practice, the absence of a central agency is crucial, since it could not handle the volume of traffic that would be generated and would be exceedingly vulnerable to attack. Hence a distributed system is a *sine qua non*.

[3]The reader might wonder how agents who do not meet in person can exchange tokens at all, since they can only exchange electronic files, and electronic files would seem to be easily duplicated. In fact, however, there are practicable, secure and private procedures for online token exchange, utilizing hardware or software or both; see Buttyan & Hubaux (2001) [BH01], Vishnumurthy, Chandrakumar & Sirer (2003) [VCS03] and Ciuffoletti (2010) [Ciu10] for instance. Similar procedures can also serve as escrow accounts to assure that service that is promised is actually provided and that payment that is promised is actually made.

[4]In the real systems we have in mind, the population is in the tens of thousands or hundreds of thousands so a continuum model seems a reasonable approximation.

We assume benefit exceeds cost, so that social welfare is increased when the service is provided, but that cost is strictly positive, so that the server has a disincentive to provide it. The designer supplies a supply of tokens and recommends strategies (circumstances under which service should be requested or provided); together these constitute a *protocol*. We assume that the price of service is fixed at one token; this restriction seems natural in our environment and is made in much of the search literature; see also below and the Conclusion. We differ from much of the literature in three ways suggested by the motivation discussed. First, we do not impose an exogenous upper bound on money holdings: agents can store as much money as they wish. Second, we require that the protocol should induce an equilibrium (i.e., that the recommended strategies are best replies in the (unique) steady-state distribution) that is *robust* to small perturbations of the population parameters. Third, we allow the designer to control *both* the money supply *and* the price. As we shall show, each of these has significant implications.

Leaving aside degenerate protocols in which there is no trade, *all* robust equilibrium protocols are Markov (not history dependent), *symmetric* (the population plays a single pure strategy) and have a particularly simple form: clients request service whenever their token holding is above zero; servers provide service when their token holding is at or below a *threshold* K and do not provide service when the token holding is above K. We prove that robust equilibria exist but the absence of an exogenous upper holdings makes the proof surprisingly hard. (See Section 6.) Having shown that robust equilibria exist we turn to our original question: which equilibrium protocols are the most efficient? We have shown that we can restrict attention to threshold strategies; among protocols that employ the threshold K the one that would be most efficient *if agents were compliant* has token supply $K/2$. However, these protocols need not be equilibria and the most efficient protocols may have token supplies different from $K/2$; $K/2$ *need not be the optimal quantity of money*. We go on to provide estimates for efficiency of various protocols and an effective procedure for the designer to choose a "good" – if not optimal – protocol. Simulations illustrate these results and some related points.

2.1 Related Works

Following the seminal work of Kiyotaki & Wright (1989) [KW89], there is a large literature on search models of money which has contributed enormously to our understanding of money in various environments. A portion of this literature – e.g. Camera & Corbae (1999) [CC99], Berentsen (2002) [Ber02] – allows agents to accumulate more than one unit of money, while maintaining the assumption of Kiyotaki & Wright (1989) [KW89] that there is an exogenously given upper bound on money holdings; this is precluded in our environment. A different portion of this literature – e.g. Cavalcanti & Wallace (1999A, 1999B) [CW99a] [CW99b], Berentsen, Camera & Waller (2007) [BCW07], Zhu & Mannaer (2009) [ZM09] and Hu, Kennan & Wallace (2009) [HKW09] – assumes that agents have and can condition on (complete or partial) knowledge of the money holdings of (some of) their counter-parties in each match, which is again precluded in our environment. A particularly striking paper in this literature is Kocherlakota (2002) [Koc02], which shows that any individually rational outcome can be supported in equilibrium provided money is infinitely divisible and the common discount factor is above some minimum threshold. However, Kocherlakota (2002) [Koc02] also assumes that agents have a good deal of information about the money holdings of their counterparties. To quote the abstract: "The one-money theorem says that the allocation is achievable using only one money if that money is divisible and money holdings are observable. The two-money theorem says that the allocation is achievable using two divisible monies, even if money holdings are concealable." To elaborate: the one-money theorem assumes that agents must display their *true* money holdings; the two-money theorem assumes that agents can display *less* money than they actually have but cannot display *more* money than they actually have. In both cases, agents have (complete or partial) knowledge about the money holdings of their counter-parties and can condition on it. In our work, agents have *no* knowledge about the money holdings of their counter-parties and so *cannot* condition on it.

Our work is closest to Zhou (1999) [ZM09] and Berentsen (2000) [Ber00]. Zhou (1999) assumes money is divisible and the supply of money is given endogenously but the price

is determined endogenously.[5] Berentsen (2000) assumes money is indivisible and the price is given exogenously but the money supply is determined endogenously. In our work, both the money supply *and* the price are chosen exogenously by the designer. Of course, from an economic point of view, all that really matters is the ratio M/p of the money supply M to the price p; fixing either the money supply or the price amounts simply to choosing a normalization. So it would be more accurate to say that in both Zhou (1999) and Berentsen (2000) the ratio M/p is determined *endogenously*: fixing M, as Zhou (1999) does, is just choosing a normalization; fixing p, as Berentsen (2000) does, is just choosing a different normalization. In our work, we have chosen a particular price normalization $p = 1$ but the money supply, and hence the ratio M/p, is determined *exogenously* by the designer. Our designer has more control and that control is important because if the designer did not control M/p the designer could not be sure of designing an optimal equilibrium protocol. Indeed, if the designer did not control M/p it would seem to make no sense to even talk about designing protocols, much less optimal equilibrium protocols. We also note that neither Zhou (1999) nor Berentsen (2000) prove that equilibrium exist; they both provide sufficient conditions but those conditions are stringent and endogenous – they are not conditions on the primitives of the model (benefit-cost ratio and discount factor).

This work also connects to an Electrical Engineering and Computer Science literature that discusses token exchanges in online communities. Some of that literature assumes that agents are compliant, rather than self-interested, and does not treat incentives and equilibrium Chandrakumar, Sirer & Vishnumurthy (2003) [VCS03], Buttyan & Hubaux (2003) [BH03]; some of that literature makes use of very different models than the one offered here Jarvis & Tan (2006) [TJ06] and Figueiredo, Shapiro & Towsley (2004) [FST04]; and some of the literature is not formal and rigorous, offering simulations rather than theorems Mohr & Pai (2006) [PM06]. The papers closest to ours are probably Friedman, Halpern and Kash (2006, 2007) [FHK06] [KFH07], which treat somewhat different models. However, these papers seem puzzling in many dimensions and many of the proofs seem mysterious (at least to us).

Another literature to which this work connects is the game-theoretic literature on anony-

[5] We say "the" price because Zhou (1999) considers only single-price equilibria.

mous interactions. In a context in which interactions were publicly observable, full cooperation (i.e., provision of service) could be achieved at equilibrium by the use of trigger strategies, which deny service in the future to any agent who refuses service in the present. As Kandori (1992) [Kan92] and Ellison (2000) [Ell00] have pointed out, in some contexts, cooperation can be supported even without public observability if agents deny service in the future to *all* agents whenever they have observed an agent who refuses service in the present; in this equilibrium any failure to provide service results in a contagion, producing wider and wider ripples of defection, until no agent provides service. However contagion is not likely to sustain cooperation in the systems of interest to us, because the population is so large (typically comprising tens of thousands or even hundreds of thousands of agents) that an agent is unlikely, in a reasonable time frame, to meet any other agent whose network of past associations overlap with his. (When the population is literally a continuum, no agent *ever* meets any other agent whose network of past associations overlap with his.) A more relevant literature, of which Kandori (1992) is again the seminal work, uses reputation and social norms as devices as a means of incentivizing cooperation. The work that is closest to ours is Park, van der Schaar & Zhang (2010) [ZPS14], which asks which reputation-based systems can be supported in equilibrium and which of these achieve the greatest social efficiency. Because provision of service in their model depends on the reputations of *both* client and server, some central authority must keep track of and verify reputations; hence these systems are not distributed in the sense we use here.

2.2 Model

The population consists of a continuum (mass 1) of infinitely lived agents. Each agent can provide a resource (e.g, a data file, audio file, video file, service) that is of benefit to others but is costly to produce (uploading a file uses bandwidth and time). The benefit of receiving this resource is b and the cost of producing it is c; we assume $b > c > 0$.[6] Agents care about current and future benefits/costs and discount future benefits/costs at the constant

[6]If $b \leq c$ there is no social value to providing service; if $c \leq 0$ agents will always be willing to provide service.

rate $\beta \in (0,1)$. Agents are risk neutral so seek to maximize the discounted present value of a stream of benefits and costs.

Time is discrete. In each time period, a fraction $\rho \leq 1/2$ of the population is randomly chosen to be a *client* and matched with a randomly chosen *server*; the fraction $1 - 2\rho$ are unmatched.[7] (No agent is both a client and a server in the same period.) When a client and server are matched, the client chooses whether or not to request service, the server chooses whether or not to provide service (e.g., transfer the file) if requested.

The parameters b, c, β, ρ completely describe the environment. Because the units of benefit b and cost c are arbitrary (and tokens have no intrinsic value), only the benefit-cost ratio $r = b/c$ is actually relevant. We consider variations in the benefit-cost ratio r and the discount factor β, but view the matching rate ρ as immutable.

2.2.1 Tokens and Strategies

In a single interaction between a server and a client, the server has no incentive to provide services to the client. The mechanism we study for creating incentives to provide service involves the exchange of *tokens*. Tokens are indivisible, have no intrinsic value, cannot be counterfeited, and can be stored and transferred without loss. Each agent can hold an arbitrary non-negative finite number of tokens, but cannot hold a negative number of tokens and cannot borrow. We emphasize that our tokens are purely electronic objects and are transferred electronically.

The designer creates incentives for the agents to provide or share resources by providing a supply of *tokens* and recommending *strategies* (behavior) for agents when they are clients and servers. At the moment, we allow for strategies that depend on histories but we show that optimal strategies (best responses) depend only on current token holdings.

An *event* describes the particulars of a match at a particular time: whether the agent was chosen to be a client or a server or neither, whether the agent was matched with someone who

[7] We assume that the matching procedure is such that the Law of Large Numbers holds exactly; Duffie & Sun (1997) [DS07], Alós-Ferrer (1999) [Alo99], Podczeck (2010) [Pod10] and Podczeck & Puzzello (2012) [PP12] construct such matching procedures.

was willing to serve or to buy, whether the agent received a benefit and surrendered a token or provided service and acquired a token or neither, and the change in the token holding. Write ϵ_t for an event at time t. A *history of length* T specifies an initial token holding m and a finite sequence of events $h = (m; \epsilon_0, \epsilon_1, \epsilon_{T-1})$. Write H_T for the set of histories of length T, $H = \bigcup_T H_T$ for the set of finite histories. An *infinite history* specifies an initial token holding m and an infinite sequence of events $h = (m; \epsilon_0, \epsilon_1, \ldots)$. We insist that finite/infinite histories be feasible in the sense that net token holdings are never negative (i.e., a request for service by an agent holding 0 tokens will not be honored). Given a finite or infinite history h, write $d(h,t)$ for the *change* in token holding at time t and $d^+(h,t), d^-(h,t)$ for the positive and negative parts of $d(h,t)$. Note that $d(h,t) = +1$ if the agent serves, $d(h,t) = -1$ if the agent buys, $d(h,t) = 0$ otherwise. Note also that the token holding at the end of the finite history h is

$$N(h) = m + \sum_{t=0}^{T-1} d(h,t)$$

A *strategy* is a pair $(\sigma, \tau) : H \to \{0,1\}$; τ is the *client strategy* and σ is the *server strategy*. Following the history h, $\tau(h) = 1$ means the client requests service and $\tau(h) = 0$ means the client does not request service; $\sigma(h) = 1$ means the server provides service, $\sigma(h) = 0$ means the server does not provide service. (Note that we require individual agents to follow pure strategies, but we will eventually allow for the possibility that different agents follow different pure strategies, so the population strategy might be mixed.) If service is requested and provided, a single token is transferred from client to server, so the client's holding of tokens decreases by 1 and the server's holding of tokens increases by 1. Tacitly, we assume that a token is transferred if and only if service is provided; like the transfer of tokens itself, this can be accomplished electronically in a completely distributed way.

2.2.2 Steady State Payoffs, Values and Optimal Strategies

Because we consider a continuum population, assume that agents are matched randomly and can observe only their own histories, the relevant state of the system from the point of view of a single agent can be completely summarized by the fraction μ of agents who do not

request service when they are clients and the fraction ν of agents who do not provide service when they are servers. If the population is in a steady state then μ, ν do not change over time. Given μ, ν, a strategy (τ, σ) determines in the obvious way a probability distribution $P(\tau, \sigma|\mu, \nu)$ over infinite histories H. We define the *discounted expected utility* to an agent whose initial token holding is m and who follows the strategy (τ, σ) to be

$$Eu(m, \tau, \sigma|\mu, \nu) = \sum_{h \in H} P(\tau, \sigma|\mu, \nu)(h) \sum_{t=0}^{\infty} \beta^t \left[d^+(h,t)b - d^-(h,t)c \right]$$

(Here and below, when some of the variables $\beta, b, c, \mu, \nu, \tau, \sigma$ are clearly understood we frequently omit all or some of them; this should not cause confusion.)

Given μ, ν, τ, σ and an initial token holding m we define the *value* to be

$$V(m, \mu, \nu, \tau, \sigma) = \sup_{(\tau, \sigma)} Eu(m, \tau, \sigma|\mu, \nu)$$

Discounting implies that the supremum – which is taken over *all* strategy profiles – exists and is at most $b/(1-\beta)$.

Given μ, ν the strategy (τ, σ) is *optimal* or *a best response* for an initial token holding of m if

$$Eu(m, \tau, \sigma|\mu, \nu) \geq Eu(m, \tau', \sigma'|\mu, \nu)$$

for all alternative strategies τ', σ'. Because agents discount the future at the constant rate β, the strategy (τ, σ) is optimal if and only it has the *one-shot deviation property*; that is, there does not exist a finite history h and a profitable deviation (τ', σ') that differs from (τ, σ) following the history h and nowhere else. A familiar and straightforward diagonalization argument establishes that optimal strategies exist and achieve the value; we record this fact below, omitting the proof.

Proposition 1. *For each μ, ν and each initial token holding m there is an optimal strategy τ, σ and*

$$Eu(m, \tau, \sigma|\mu, \nu) = V(m, \mu, \nu, \tau, \sigma)$$

2.2.3 Optimal Strategies

We want to characterize optimal strategies, but before we do, there is a degeneracy that must be addressed. If $\mu = 1$ then no one ever requests service so the choice of whether to provide service is irrelevant; if $\nu = 1$ then no one ever provides service so the choice of whether to request service is irrelevant. In what follows, we sometimes ignore or avoid these degenerate cases, but this should not lead to any confusion.

Fix β, b, c, μ, ν; let (τ, σ) be optimal for the initial token holding m. Note that the continuation of (τ, σ) must also be optimal following every history that begins with m. If h is such a history and the token holding at h is n then (τ, σ) induces a strategy (τ^h, σ^h) from an initial token holding n that simply transposes what follows h back to time 0, and this strategy must be optimal for the initial token holding of n. Conversely, any strategy that is optimal for the initial token holding of n must also be optimal following h. It follows that optimal strategies (τ, σ) (whose existence is guaranteed by Proposition 1) *depend only on the current token holding* but are otherwise independent of history; we frequently say such strategies are *Markov* – but note that they are Markov in *individual* token holdings. Write $\Sigma(\mu, \nu, \beta)$ for the set of optimal strategies.

Theorem 1. *For all b, c, β, μ, ν with $\nu < 1$, every optimal strategy (τ, σ) has the property that $\tau(n) = 1$ for every $n \geq 1$; i.e. "always request service when possible".*[8]

In view of Theorem 1, we suppress client strategies τ entirely, assuming that clients always request service whenever possible. We abuse notation and continue to write $\Sigma(\mu, \nu, \beta)$ for the set of optimal strategies.

We now show that optimal (server) strategies also have a simple form. Say that the (server) strategy σ is a *threshold strategy (with threshold K)* if

$$\begin{aligned} \sigma(n) &= 1 \quad \text{if} \quad n \leq K \\ \sigma(n) &= 0 \quad \text{if} \quad n > K \end{aligned} \tag{2.1}$$

[8] *Because a request for service will not be honored when an agent holds 0 tokens, it is irrelevant whether $\tau(0) = 0$ or $\tau(0) = 1$.*

We write σ_K for the threshold strategy with threshold K and

$$\Sigma = \{\sigma_K : 0 \leq K < \infty\}$$

for the set of threshold strategies.

Theorem 2. *For each μ, ν, b, c, β with $\mu < 1$ the set of optimal (server) strategies consists of either a single threshold strategy or two threshold strategies with adjacent thresholds.*

(The assumptions in Theorems 1 and 2 that $\nu < 1$ and $\mu < 1$ avoid the degeneracies previously noted.)

2.2.4 Protocols

The designer chooses a *per capita* supply of tokens $\alpha \in (0, \infty)$ and recommends a strategy to each agent; we allow for the possibility that the designer recommends different strategies to different agents. Because self-interested agents will always play a best response, the designer will recommend only strategies in Σ; in view of anonymity, it does not matter which agents are recommended to play each strategy, but rather only the fraction of agents recommended to play each strategy. Hence we can identify a recommendation with a *mixed threshold strategy*, which is a probability distribution on Σ; with the obvious abuse of notation, we view γ as a function $\gamma : \mathbb{N}_+ \to [0, 1]$ such that

$$\begin{aligned} \gamma(K) &\geq 0 \text{ for each } K \geq 0 \\ \sum_{K=0}^{\infty} \gamma(K) &= 1 \end{aligned}$$

Write $\Delta(\Sigma)$ for the set of mixed threshold strategies. As usual, we identify the threshold strategy σ_K with the mixed strategy that puts mass 1 on σ_K. Assuming that the designer only recommends best responses (because other recommendations would not be followed), we interpret an element $\gamma \in \Delta(\Sigma)$ as a recommendation that the fraction $\gamma(K)$ play the threshold strategy σ_K.

A *protocol* is a pair $\Pi = (\alpha, \gamma)$ consisting of a per-capita supply of tokens $\alpha \in (0, \infty)$ and a mixed strategy recommendation $\gamma \in \Delta(\Sigma)$.

2.2.5 Invariant Distributions

If the designer chooses the protocol $\Pi = (\alpha, \gamma)$ and agents follow the recommendation γ, we can easily describe the evolution of the *token distribution* (the distribution of token holdings). The token distribution must satisfy the two feasibility conditions:

$$\sum_{k=0}^{\infty} \eta(k) = 1 \tag{2.2}$$

$$\sum_{k=0}^{\infty} k\eta(k) = \alpha \tag{2.3}$$

Write

$$\mu = \eta(0), \quad \nu = \sum_{\sigma(k)=0} \eta(k)$$

Evidently, with respect to this token distribution, μ is the fraction of agents who have no tokens, hence cannot pay for service, and ν is the fraction of agents who do not serve (assuming they follow the protocol).

To determine the token distribution next period, it is convenient to think backwards and ask how an agent could come to have k tokens in the next period. There are three possibilities; the agent could have

- $k-1$ tokens in the current period, be chosen as a server, meet a client who can pay for service, and provide service (hence acquire a token);
- $k+1$ tokens in the current period, be chosen as a client, meet a server who provides service, and buy service (hence expend a token);
- k tokens in the current period but neither provide service nor buy service (hence neither acquire nor expend a token).

Given a recommendation γ it is convenient to define $\sigma^\gamma : \mathbb{N}_+ \to [0,1]$ by

$$\sigma^\gamma(n) = \sum_{K=0}^{\infty} \gamma(K)\sigma_K(n)$$

Assuming that the Law of Large Numbers holds exactly in our continuum framework and that all agents follow the recommendation γ, $\sigma^\gamma(n)$ is the fraction of agents in the population

who serve when they have n tokens, so σ^γ is the population strategy. Keeping in mind that token holdings cannot be negative, it is easy to see that the token distribution next period will be

$$\begin{aligned}\eta_+(k) &= \eta(k-1)[\rho(1-\mu)\sigma^\gamma(k-1)] \\ &\quad + \eta(k+1)[\rho(1-\nu)] \\ &\quad + \eta(k)[1-\rho(1-\mu)\sigma^\gamma(k)-\rho(1-\nu)] \end{aligned} \tag{2.4}$$

where we use the convention $\eta(-1) = 0$.

Given the protocol $\Pi = (\alpha, \gamma)$, the (feasible) token distribution η is *invariant* if $\eta_+ = \eta$; that is, η is stationary when agents comply with the recommendation γ. Invariant distributions always exist and are unique.

Theorem 3. *For each protocol $\Pi = (\alpha, \gamma)$ there is a unique invariant distribution η^Π, which is completely determined by the feasibility conditions (2.2) and (2.3) and the recursion relationship*

$$\begin{aligned}\eta^\Pi(k) &= \eta^\Pi(k-1)[\rho(1-\mu)\sigma^\gamma(k-1)] \\ &\quad + \eta^\Pi(k+1)[\rho(1-\nu)] \\ &\quad + \eta^\Pi(k)[1-\rho(1-\mu)\sigma^\gamma(k)-\rho(1-\nu)] \end{aligned} \tag{2.5}$$

2.2.6 Definition of Equilibrium and Robust Equilibrium

Assuming agents are rational and self-interested, they will comply with a given protocol if and only if compliance is individually optimal; that is, no agent can benefit by deviating from the protocol. To formalize this, fix a protocol $\Pi = (\alpha, \gamma)$, and let η^Π be the unique invariant distribution. Write

$$\mu^\Pi = \eta^\Pi(0)\ ,\ \nu^\Pi = \sum_{\sigma(k)=0} \eta^\Pi(k)$$

for the fraction of agents who have no tokens and the fraction of agents who do not serve (in the invariant distribution induced by Π), respectively. We say $\Pi = (\alpha, \gamma)$ is an *equilibrium*

protocol if σ_K is an optimal strategy (given given μ^Π, η^Π) whenever $\gamma(K) > 0$. That is, γ puts positive weight only on threshold strategies that are optimal, given the invariant distribution that Π itself induces.

Using the one step deviation principle, we can provide a useful alternative description of equilibrium in terms of the value function V. As noted before, because optimal strategies exist and are Markov, we may unambiguously write V_k for the value following *any* history at which the agent has k tokens. (The value function depends on the population data μ, ν and on the environmental parameters b, c, β; but there should be no confusion in suppressing those here.)

Fix any Markov strategy σ. In order for σ to be optimal, it is necessary and sufficient that it achieves the value V_ℓ following *every* token holding ℓ. Expressed in terms of *current* token holdings and *future* values, and taking into account how behavior in a given period affects the token holding in the next period, this means that σ is optimal if and only if it satisfies the following system of equations:

$$\begin{aligned} V_0 &= \rho\sigma(0)[(1-\mu)(-c+\beta V_1) + \mu\beta V_0] \\ &\quad +\rho[1-\sigma(0)]\beta V_0 + (1-2\rho)\beta V_0 \end{aligned}$$

$$\begin{aligned} V_k &= \rho[(1-\nu)(b+\beta V_{k-1}) + \nu\beta V_k] \\ &\quad + \rho\sigma(k)[(1-\mu)(-c+\beta V_{k+1}) + \mu\beta V_k] \\ &\quad +\rho[1-\sigma(k)]\beta V_k + (1-2\rho)\beta V_k \\ &\quad \text{for each } k > 0 \end{aligned} \tag{2.6}$$

Applying this observation to the threshold strategy σ_K and carrying out the requisite algebra, we conclude that σ_K is optimal if and only if

$$-c + \beta V_{k+1} \geq \beta V_k \text{ if } k \leq K \tag{2.7}$$

$$-c + \beta V_{k+1} \leq \beta V_k \text{ if } k > K \tag{2.8}$$

(If it seems strange that α, γ do not appear in these inequalities, remember that the value depends on the invariant distribution η^Π, which in turn depends on α and on γ.)

Given a benefit/cost ratio $r > 1$ and a discount factor $\beta < 1$, write $EQ(r, \beta)$ for the set of protocols Π that constitute an equilibrium when the benefit/cost ratio is r and the discount factor is β. Conversely, given a protocol Π write $E(\Pi)$ for the set $\{(r, \beta)\}$ of pairs of benefit/cost ratios r and discount factors β such that Π is an equilibrium protocol when the benefit/cost ratio is r and discount factor is β. Note that EQ, E are correspondences (which might have empty values) and are inverse to each other.

Given r, β we say that Π is a *robust equilibrium* if (r, β) belongs to the interior of $E(\Pi)$; i.e., there is some $\varepsilon > 0$ such that $\Pi \in EQ(r', \beta')$ whenever $|r' - r| < \varepsilon$ and $|\beta' - \beta| < \varepsilon$. Write $EQR(r, \beta)$ for the set of robust equilibrium protocols for the benefit/cost ratio r and discount factor β and $ER(\Pi)$ for the set $\{(r, \beta)\}$ of pairs of benefit/cost ratios for which Π is a robust equilibrium. Note that EQR, ER are correspondences (which might have empty values) and are inverse to each other.

2.3 Equilibrium and Robust Equilibrium

We first describe the nature of equilibrium and robust equilibrium and then use that description to show that robust equilibria exist. The crucial fact about equilibrium is that the strategy part of an equilibrium protocol can involve mixing over at most two thresholds and that these thresholds must be adjacent; the crucial fact about robust equilibrium is that the strategy cannot involve strict mixing at all but must rather be a pure strategy.

Theorem 4. *For each benefit/cost ratio $r > 1$ and discount factor $\beta < 1$ the set $EQ(r, \beta)$ is either empty or consists of protocols that involve only (possibly degenerate) mixtures of two threshold strategies with adjacent thresholds.*

Theorem 5. *If $\Pi = (\alpha, \sigma)$ is a robust equilibrium then σ is a pure threshold strategy.*

The existence of equilibrium or robust equilibrium does not seem at all obvious (and our proof is not simple). For both intuition and technical convenience, it is convenient to work "backwards": rather than beginning with population parameters r, β and looking for protocols Π that constitute an equilibrium for those parameters, we begin with a protocol Π

and look for population parameters r, β for which Π constitutes an equilibrium. That is, we do not study the correspondences $EQ(r, \beta)$ and $EQR(r, \beta)$ directly, but rather the inverse correspondences $E(\Pi)$ and $ER(\Pi)$. This is easier for several reasons, one of which is that the latter correspondences *always* have non-empty values.

To give an intuitive understanding of the difficulty and how we overcome it, fix a protocol $\Pi = (\alpha, \sigma)$ and let η^Π be the invariant distribution. Because we will eventually want to find a robust equilibrium, we assume σ is a threshold strategy: $\sigma = \sigma_K$. To look for population parameters r, β for which Π is an equilibrium, let us fix r and let β vary. (We could fix β and let r vary, or vary both β, r simultaneously, but the intuition is most easily conveyed by fixing r and letting β vary.) As we have already noted, the invariant distribution η^Π, and hence μ^Π, ν^Π, depend only on Π and so do not change as β varies. Given the invariant distribution, if β is close to 0, an agent has little incentive to acquire tokens; however the incentive to acquire tokens increases as $\beta \to 1$. It can be shown that there is a smallest discount factor $\beta_L(\Pi)$ with the property that an agent whose discount factor is at least $\beta_L(\Pi)$ will be willing to continue providing service until he has acquired K tokens. This is not enough, because σ_K will only be incentive compatible if the agent is also willing to *stop* providing service *after* he has acquired K tokens. However, it can also be shown that there is a largest discount factor $\beta_H(\Pi)$ for which the agent is willing to stop providing service after he has acquired K tokens, and that $\beta_L(\Pi) < \beta_H(\Pi)$. (Recall that r, Π are fixed.) For every discount factor β in the closed interval $[\beta_L(\Pi), \beta_H(\Pi)]$, the protocol Π is an equilibrium when the population parameters are r, β; that is, $(r, \beta) \in E(\Pi)$. From this it can be shown that for every discount factor β in the interval $(\beta_L(\Pi), \beta_H(\Pi))$, the protocol Π is a robust equilibrium when the population parameters are r, β; that is, $(r, \beta) \in ER(\Pi)$. Similarly, we can hold β fixed and let r vary from 1 to ∞, construct the corresponding intervals $[r_L(\Pi), r_H(\Pi)]$ with $r_L(\Pi) < r_H(\Pi)$ and then show that for every benefit/cost ratio r in the open interval $(r_L(\Pi), r_H(\Pi))$ the protocol Π is a robust equilibrium when the population parameters are r, β; that is, $(r, \beta) \in ER(\Pi)$. This is the content of Theorem 6 below.

Applying this procedure for every protocol yields a family $\{ER(\Pi)\}$ of non-empty open sets of parameters r, β for which robust equilibria exist. However our work is not done

because we do not know whether a robust equilibrium exists for *given* population parameters r, β. To see that it does, we show that $\{ER(\Pi)\}$ *covers* a big enough set of population parameters. In particular, for each $r > 1$ there is a $\beta^* < 1$ such that $\{ER(\Pi)\}$ covers the set $\{\Pi(r, \beta) : \beta > \beta^*\}$; this means that for each $r > 1$ and $\beta > \beta^*$ there is a protocol Π that constitutes a robust equilibrium for the population parameters r, β. Similarly, for each $\beta > 0$ there is a $r^* > 0$ such that if $r > r^*$ there is a protocol that constitutes a robust equilibrium for the population parameters β, r. The proof is not easy; to do so, we first establish (Theorem 6) some special properties of protocols of the form $\Pi_K = (K/2, \sigma_K)$; we then apply these special properties (Theorem 7) to obtain the desired result.

It is natural to ask why our proof seems (and is) so much more complicated than existence proofs in the literature, such as in Berentsen (2002) [Ber02]. The answer is that the literature establishes the existence of equilibrium only under the assumption that there is an *exogenous upper bound* K^* on the number of tokens any agent can hold. As discussed above, this assumption makes it relatively easy to show that equilibrium exists: Fix the benefit/cost ratio $r > 1$ and an arbitrary $\alpha > 0$ and consider the protocol (α, σ_{K^*}). As above, an agent whose discount factor β is at least $\beta_L(\alpha, \sigma_{K^*})$ will provide service until he has acquired K^* tokens; under the assumption that K^* is an upper bound on the number of tokens any agent can hold, the agent will *stop* providing service *after* he has acquired K^* tokens because, by assumption, he *cannot hold more than* K^* *tokens* so providing service incurs a present cost with no future benefit. Hence (α, σ_{K^*}) is an equilibrium protocol for *every* $\beta \geq \beta_L(\Pi)$ and is a robust equilibrium protocol for every $\beta > \beta_L(\alpha, \sigma_{K^*})$. Thus, *any* protocol can be supported in equilibrium so long as agents are sufficiently patient. As we have noted in the Introduction, assuming an exogenous upper bound on token holdings does not seem realistic in the environments we consider.

Theorem 6. *Fix a protocol* $\Pi = (\alpha, \sigma_K)$.

(i) For each benefit/cost ratio $r > 1$*, the set* $\{\beta : \Pi \in EQ(r, \beta)\}$ *is a non-degenerate closed interval* $[\beta_L(\Pi), \beta_H(\Pi)]$ *whose endpoints are continuous functions of* r.

(ii) For each discount factor $\beta < 1$*, the set* $\{r : \Pi \in EQ(r, \beta)\}$ *is a non-degenerate closed*

interval $[r_l(\Pi), r_H(\Pi)]$ *whose endpoints are continuous functions of* β.

These results are illustrated for $\alpha = 1/4$ in in Figures 2.1 and 2.2. (Figure 2.1 may give the impression that the intervals for successive values of K do not overlap, but as Figure 2.2 illustrates, they actually *do* overlap; the overlap is masked by the granularity of the Figure. However, as we have already said, we do not assert that overlapping of intervals for successive values of K is a general property.)

For the special protocols $\Pi_K = (K/2, \sigma_K)$, in which the supply of tokens is exactly half the selling threshold we prove in Theorem 7 below that the intervals corresponding to successive values of the threshold overlap but are not nested. This is exactly what we need to guarantee that *(non-degenerate) equilibria always exist* provided that β, r are sufficiently large. Theorem 10 provides estimates on how big β, r must be.)

Theorem 7. *Robust equilibria exist whenever* β, r *are sufficiently large. More precisely:*

(i) For each fixed threshold K *and benefit-cost ratio* $r > 1$*, successive* β*-intervals overlap but are not nested:*

$$\beta_L(\Pi_{K-1}) < \beta_L(\Pi_K) < \beta_H(\Pi_{K-1}) < \beta_H(\Pi_K)$$

Moreover

$$\lim_{K\to\infty} \beta_L(\Pi_K) = 1$$

In particular, there is some $\beta^* < 1$ *such that* $EQR(r, \beta) \neq \emptyset$ *for all* $\beta > \beta^*$.

(ii) For each fixed threshold K *and discount factor* $\beta < 1$*, successive* r*-intervals overlap but are not nested:*

$$r_L(\Pi_{K-1}) < r_L(\Pi_K) < r_H(\Pi_{K-1}) < r_H(\Pi_K)$$

Moreover

$$\lim_{K\to\infty} r_L(\Pi_K) = \infty$$

In particular, there is some $r^* > 1$ *such that* $EQR(r, \beta) \neq \emptyset$ *for all* $r > r^*$.

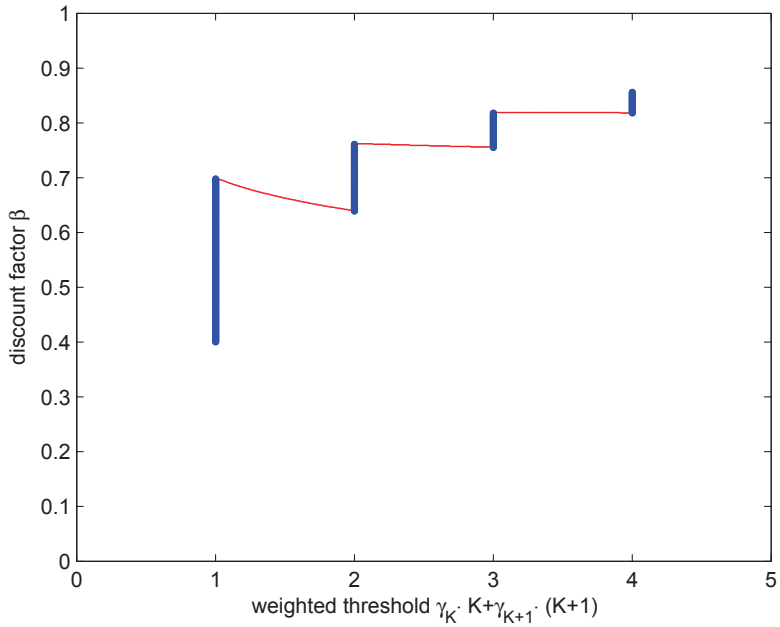


Figure 2.1: Pure and Mixed Equilibrium: $\alpha = 1/4$
(blue (thick) - pure equlibrium; red (thin) - mixed equilibrium)

It follows from Theorem 7 that, as $K \to \infty$, the left-hand end-points $\beta_L(\Pi_K) \to 1$, so *a fortiori* the lengths of β-intervals shrink to 0. It is natural to guess that the lengths of these intervals shrink *monotonically* to 0, and simulations suggest that this guess is correct, but we have neither a proof nor a good intuition that this is actually true. We also guess that the lengths of r-intervals shrink monotonically, but again we have neither a proof nor a good intuition that this is actually true.

2.4 Efficiency

If agents were compliant (rather than self-interested), the designer could simply instruct them to provide service at every meeting and they would comply, so the per capita social gain in each period would be $\rho(b-c)$. If agents follow the protocol $\Pi = (\alpha, \sigma_K)$ then service will be provided only in those meetings where the client can buy service and the server is willing to provide service, so the per capita social gain in each period will be $\rho(b-c)(1-\mu^\Pi)(1-\nu^\Pi)$.

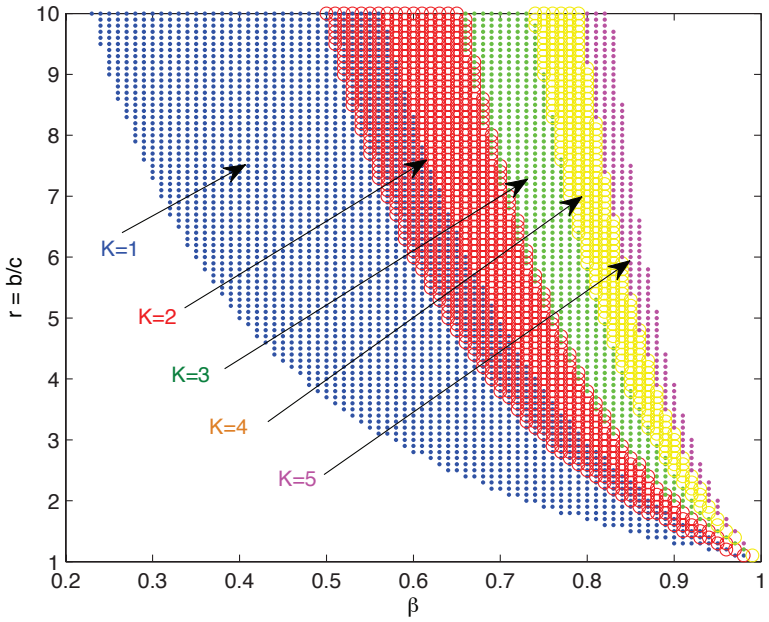


Figure 2.2: Threshold Equilibrium: $\alpha = 1/4$

Hence we define the *efficiency* of the protocol Π to be

$$\text{Eff}(\Pi) = (1 - \mu^{\Pi})(1 - \nu^{\Pi})$$

In general it seems hard to determine the efficiency of a given protocol or to compare the efficiency of different protocols. However, we can provide efficiency bounds for protocols that utilize a given threshold strategy σ_K and compute the precise efficiency of the protocols Π_K.[9]

Theorem 8. *For each $\alpha \in (0, \infty)$, each threshold K and all values of the population parameters we have:*

(i) $Eff(\alpha, \sigma_K) \leq 1 - \frac{1}{2\lceil\alpha\rceil+1}$

(ii) $Eff(\alpha, \sigma_K) \leq Eff(\Pi_K)$

(iii) $Eff(\Pi_K) = \left(1 - \frac{1}{K+1}\right)^2 = \left(\frac{K}{K+1}\right)^2$

[9]Berentsen (2002) [Ber02] derives similar results in a different model, with Poisson arrival rates.

Two implications of Theorem 8 are immediate. The first is that, in order that a (threshold) protocol achieve efficiency near 1 it is necessary that it provide a large number of tokens *and also* that it prescribe a high selling threshold. Put differently: to yield full efficiency in the limit it is *not* enough to increase the number of tokens without bound or to increase the threshold without bound – *both* must be increased without bound. The second is that the protocols Π_K that provide $K/2$ tokens per capita are the most efficient protocols that utilize a given threshold strategy σ_K.

We caution the reader, however, that the protocols Π_K *need not be* equilibrium *protocols*, and it is (robust) equilibrium protocols that we seek. However, it follows immediately from Theorem 7 that whenever agents are sufficiently patient or the benefit-cost ratio is sufficiently large (or both), then *some* protocol Π_K is an equilibrium for large K, and hence that nearly efficient equilibrium protocols always exist.

Theorem 9.

(i) for each fixed discount factor $\beta < 1$

$$\liminf_{r\to\infty} \sup\{\mathit{Eff}(\Pi_K) : \Pi_K \in EQR(\beta, r)\} = 1$$

(ii) for each fixed benefit-cost ratio $r > 1$

$$\liminf_{\beta\to 1} \sup\{\mathit{Eff}(\Pi_K) : \Pi_K \in EQR(\beta, r)\} = 1$$

In words: as agents become arbitrarily patient or the benefit/cost ratio becomes arbitrarily large, it is possible to choose *robust equilibrium protocols* that achieve efficiency arbitrarily close to first best. Some intuition might be useful. Consider the protocols Π_K and the corresponding invariant distributions. As K increases, the fraction of agents who cannot purchase service and the fraction of agents who will do not provide both decrease – so efficiency increases. However, if r, β are fixed and K increases then the protocols Π_K will eventually cease to be equilibrium protocols so equilibrium efficiency is bounded. On the other hand, if we fix r and let $\beta \to 1$ or fix β and let $r \to \infty$ then the thresholds K for which the protocols Π_K *are* equilibrium protocols blow up, and hence efficiency tends to 1. Put

differently: *high discount factors or high benefit/cost ratios make the use of high thresholds consistent with equilibrium.*

Theorem 9 provides asymptotic efficiency results; the following result presents an explicit lower bound (in terms of the population parameters r, β) for the efficiency obtainable by a robust equilibrium protocol.

Theorem 10. *Given the benefit/cost ratio $r > 1$ and the discount factor $\beta < 1$, define*[10]

$$\begin{aligned} K^L &= \max\left\{\log_{\frac{\rho\beta}{2(1-\beta)+2\rho\beta}}\left(\frac{1}{1+r}\right) - 1\ , 0\right\} \\ K^H &= \log_{\frac{\rho\beta}{1-\beta+\rho\beta}}\left(\frac{1}{2r}\right) \end{aligned}$$

Then:

(i) all the thresholds K for which Π_K is a robust equilibrium protocol lie in the interval $[K^L, K^H]$;

(ii) the efficiency of the optimal robust equilibrium protocol is at least $\left(1 - \frac{1}{K^L+1}\right)^2 = \left(\frac{K^L}{K^L+1}\right)^2$.

Theorem 10 yields a lower bound on efficiency because the optimal robust equilibrium protocol is at least as efficient as any protocol Π_K that is a robust equilibrium, but does not yield an upper bound on efficiency because the optimal robust equilibrium protocol might be more efficient than any protocol Π_K that is a robust equilibrium.

Theorem 10 also yields the designer an effective procedure for finding a robust equilibrium whose efficiency is *good*, if not optimal, since all that is necessary is to check protocols Π_K with thresholds K in the (finite) interval $[K^L, K^H]$. Moreover, it is not necessary to conduct an exhaustive search. Rather the designer can begin by checking the protocol Π_K, where K is the midpoint of the interval $[K^L, K^H]$. If $M_{\sigma_K}(K-1) \geq c/\beta$ and $M_{\sigma_K}(K) \leq c/\beta$, then Π_K is an equilibrium protocol and the search can stop. If $M_{\sigma_K}(K-1) < c/\beta$, then for all $K' > K$, $M_{\sigma_{K'}}(K'-1) < c/\beta$ (because $\beta^L(\Pi_{K'}) > \beta^L(\Pi_K)$). Therefore threshold protocols for which $K' > K$ cannot be an equilibrium and the designer can restrict search

[10] *Note that both the basis of the logarithms and the arguments are less than 1.*

to the left half interval $[K^L, K]$. If $M_{\sigma_K}(K) > c/\beta$, then for all $K' < K$, $M_{\sigma_{K'}}(K') > c/\beta$ (because $\beta^H(\Pi_{K'}) > \beta^H(\Pi_K)$). Therefore threshold protocols for which $K' < K$ cannot be an equilibrium and the designer can restrict search to the right half interval $[K, K^H]$. Continuing to bisect in this way, the designer can find an equilibrium threshold protocol in at most $\log_2(K^H - K^L)$ iterations.

2.4.1 The Optimal Quantity of Money

The question naturally arises: "Which equilibrium protocols are most efficient?" Because all robust equilibrium protocols are threshold protocols, this amounts to asking for which values of α, K is (α, σ_K) the most efficient equilibrium protocol. If we focus on α we are asking a familiar question: "What is the optimal quantity of money?" Kiyotaki & Wright (1989) [KW89] constrain agents to hold no more than 1 unit of money and show that the optimal quantity of money is 1/2. Berentsen (2002) [Ber02] relaxes the constraint on money holdings to K and shows that (with certain assumptions) the optimal quantity of money is $K/2$. However, this conclusion is an artifact of the constraint that agents can hold no more than K units of money. In our framework, which does not place an exogenous constraint on money holdings, $K/2$ *may not be – and often will not be – the optimal quantity of money.* Figure 2.3 illustrates this point in a simulation, but it is in fact quite a robust phenomenon.

To see what this is so, fix $r \geq 1$ and $K \geq 1$. Theorem 7 guarantees that there is an open interval of discount factors for which Π_K is an equilibrium and an open interval of discount factors for which Π_{K+1} is an equilibrium and these intervals overlap:

$$\beta_L(\Pi_K) < \beta_L(\Pi_{K+1}) < \beta_H(\Pi_K) < \beta_H(\Pi_{K+1})$$

Consider a discount factor β with $\beta_L(\Pi_K) < \beta < \beta_L(\Pi_{K+1})$. By construction, $\Pi_K = (K/2, \sigma_K)$ is an equilibrium and $\Pi_{K+1} = ((K+1)/2, \sigma_{K+1})$ is not, so Π_K is the most efficient equilibrium protocol *among the protocols* $\Pi_{K'}$. However, these are not the only protocols: if we seek the most efficient among *all* equilibrium protocols we must also consider protocols $(\alpha, \sigma_{K'})$ for values of α *other than* $\alpha = K'/2$. However, for discount factors $\beta < \beta_L(\Pi_{K+1})$ for which $|\beta_L(\Pi_{K+1}) - \beta|$ is sufficiently small, there will be token supplies $\alpha < (K+1)/2$

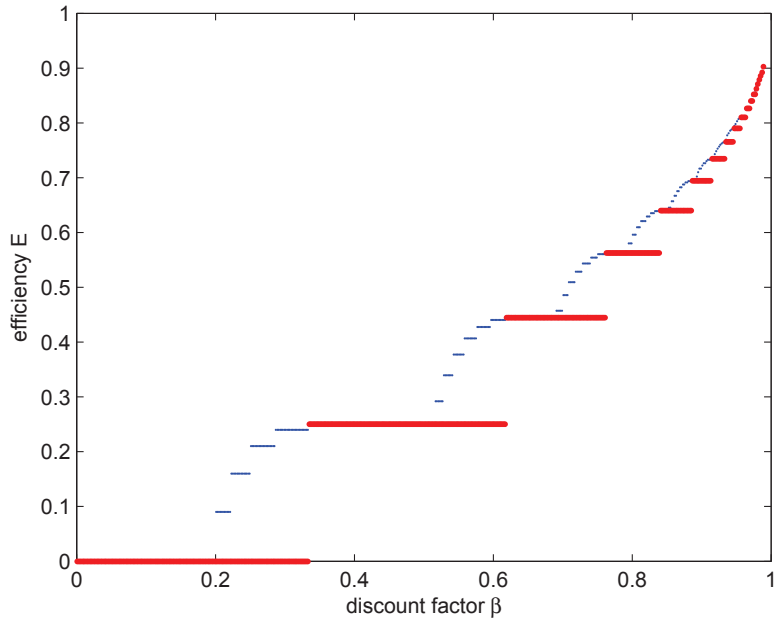


Figure 2.3: Optimal Equilibrium Protocols
(red (thick) - Π_K; blue (thin) - other protocols)

for which $|(K+1)/2 - \alpha|$ is as small as we like and for which (α, σ_{K+1}) is an equilibrium protocol. If $|(K+1)/2 - \alpha|$ is small then the invariant distributions for (α, σ_{K+1}) and for $((K+1)/2, \sigma_{K+1})$ will be close, and hence the efficiency of (α, σ_{K+1}) will be almost equal to the efficiency of $((K+1)/2, \sigma_{K+1}) = \Pi_{K+1}$. Since the efficiency of Π_{K+1} is strictly greater than the efficiency of Π_K this means that (α, σ_{K+1}) is an equilibrium protocol that is more efficient than Π_K. In other words, for discount factors less than but very close to β_{K+1}, $K/2$ is *not the optimal quantity of money.*

As this discussion illustrates, it is crucial to the design problem that the designer be able to choose the quantity of money α, since it is through α that the designer controls efficiency (social welfare).

2.4.2 Choosing the Right Protocol

The reader may wonder why we have put so much emphasis on choosing the *right* protocol. As Figure 2.4 already shows, the reason is simple: choosing the wrong protocol can result in

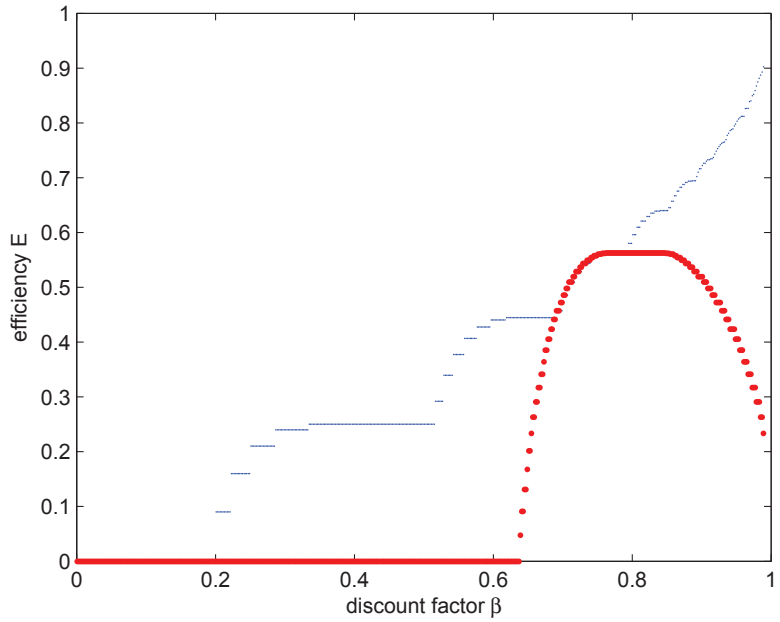


Figure 2.4: Inefficient Equilibrium Protocols
(red (thick) - Π_3; blue (thin) - optimal equilibrium protocols)

an enormous efficiency loss. Figure 4, which compares efficiency of the most efficient protocol with efficiency of a protocol for which the strategic threshold is constrained to be $K = 3$, makes this point in an even starker way: as the reader will see, except for a small range of discount factors, the efficiency loss is enormous.

2.5 Conclusions

In this chapter, we have analyzed in some detail a simple, practicable and distributed method to incentivize trade in on-line environments through the use of (electronic) tokens. We have shown that when agents are patient, the method we offer can achieve outcomes that are nearly efficient, provided the right protocol (supply of tokens and recommended threshold) is chosen, but that equilibrium and efficiency are both sensitive to the precise choice of protocol. Surprisingly, the "optimal" supply of tokens need *not* be half the recommended threshold; this conclusion, and others, and much of the difficulty of our arguments are a consequence of our allowing agents to accumulate as many tokens as they wish, rather than

imposing an exogenous bound on token holdings (which is common in the literature).

Our analysis is silent about convergence to the steady state. In particular, we do not know whether the recommended strategies would lead to convergence to the invariant distribution for all initial token distributions or for some particular token distributions. Berentsen (2002) [Ber02] proves convergence under some conditions, but in a continuous time model in which token holdings are subject to an exogenous bound; we have already noted that the latter is a strong (and, in our view, unrealistic) assumption. Another point is worth making as well. By definition, the recommended strategy is a best reply when the system is in the steady state, but the recommended strategy need not be a best reply – and very likely is *not* a best reply – when the system is *not* in the steady state – so why should agents follow it?

We have assumed that service and tokens are both indivisible. This seems a natural assumption given the environment in which we are interested because a partial file is usually worthless by itself and because there is no (extant) technology for online exchange of fractional tokens. The assumption that service and tokens are exchanged one-for-one is a genuine restriction. It is conceivable that there would exist an equilibrium in which different quantities of tokens sometimes change hands, and such equilibria (if they do exist) might be more efficient than the ones we consider here. Determining whether such equilibria exist and characterizing them (if they do exist) seems a daunting task that none of the literature seems to have addressed.[11]

We have considered the simplest setting, in which agents are identical, all files are equally valuable, and no errors occur. In a more realistic setting, we would need to take account of heterogeneous agents and files and allow for the possibilities of errors (in transmission of files or exchange of tokens or both). We have followed here the well-known adage "one has to start somewhere" – but we are keenly aware that there is much more work to be done.

[11] Zhou (1999) [Zho99] considers equilibria for various prices, but all the equilibria she studies are assumed to have a single price and agents holding in these equilibria are only in integral multiples of the single price.

2.6 Applications in Communications

The framework proposed in this chapter has many practical applications in communications networks such as wireless relaying [XS13] [MPX13] and interference mitigation in heterogeneous small cell networks [SXS15].

In [XS13], the token system design framework is extended to more practical deployment scenarios where services have heterogeneous value and is applied to solve to solve the incentive problem in wireless relay networks. A complete and rigorous system design of the overlay token system in wireless relay networks is provided, taking into account the unique wireless characteristics. A later work [MPX13] then investigated how an individual device can learn its optimal strategy online rather than focusing on incentive design from the designer's perspective.

In [SXS15], a distributed token exchange framework is proposed which can be used in heterogenous small cell networks to successfully mitigate interference among the self-interested users. Contrary to the traditional role of buying transmission [XS13] [MPX13], tokens are exchanged between users to buy silence. This paper focuses on the rigorous design of the optimal token scheme that minimizes the system outage probability.

2.7 Appendix

Proof. of Theorem 1 We first estimate $V(n+1) - V(n)$ (for $n \geq 0$) which is the loss from having one less token. To this end fix an optimal Markov strategy (τ, σ). We define a history-dependent strategy (τ', σ') and estimate the expected utility to an agent who begins with n tokens and follows (τ', σ'); this is a lower bound on $V(n)$. The strategy (τ', σ') is most easily described in the following way: Begin by following the behavior prescribed by the strategy (σ, τ) but for an agent who holds one more token than is actually held; i.e., $(\tau', \sigma')(h) = (\tau, \sigma)(N(h) + 1)$. If it never happens that the agent holds 0 tokens, requests service, and is matched with an agent who is willing to provide service, then continue in this way forever. If it does happen that the agent holds 0 tokens, requests service, and is

matched with an agent who is willing to provide service, then service is not provided in that period (because the agent cannot pay) and after that period $(\tau', \sigma') = (\tau, \sigma)$. In other words, the agent behaves "as if" he held one more token than actually held until the first time such behavior results in requesting service, being offered service, and being unable to pay for service; after that point, revert to (τ, σ). The point to keep in mind is that if a moment of deviation occurs then an agent with one more token would hold exactly 1 token, would request and receive service, and in the next period would have 0 tokens – so that reverting to (τ, σ) is possible. Beginning with n tokens and following the strategy (τ', σ') yields the same string of payoffs as beginning with $n+1$ tokens and following the strategy (τ, σ) except in the single period in which deviation occurs; in that period the expected loss of utility is at most $b\rho$. Hence the expected utility from beginning with n tokens and following the strategy (τ', σ') yields utility at least $V(n+1) - b\rho$. Hence $V(n+1) - V(n) \leq b\rho < b < b/\beta$. However, this is the incentive compatibility condition that guarantees that an agent strictly prefers to request service when holding $n+1$ tokens, so the proof is complete. □

At this point it is convenient to collect some notation and isolate two technical results. Fix ρ, b, c, μ, ν and consider a Markov strategy σ. For each k, let $V_\sigma(k, \beta)$ be the value of following σ when the initial token holding is k and the discount factor is β. As with the optimal value function V defined in the text, the value function V_σ can be defined by a recursive system of equations:

$$\begin{aligned} V_\sigma(0,\beta) &= \rho\sigma(0)[(1-\mu)(-c+\beta V_\sigma(1,\beta)) \\ &\quad +\rho[1-\sigma(0)]\beta V_\sigma(0,\beta) + (1-2\rho)\beta V_\sigma(0,\beta) \end{aligned}$$

$$\begin{aligned} V_\sigma(k,\beta) &= \rho[(1-\nu)(b+\beta V_\sigma(k-1,\beta) + \nu\beta V_\sigma(k,\beta)] \\ &\quad +\rho\sigma(k)[(1-\mu)(-c+\beta V_\sigma(k+1,\beta)) + \mu\beta V_\sigma(k,\beta)] \\ &\quad +\rho[1-\sigma(k)]\beta V_\sigma(k,\beta) + (1-2\rho)\beta V_\sigma(k,\beta) \\ &\quad \text{for} \quad k > 0 \end{aligned} \tag{2.9}$$

From the value function, we define the marginal utilities

$$M_\sigma(k,\beta) = V_\sigma(k+1,\beta) - V_\sigma(k,\beta) \tag{2.10}$$

If β is fixed/understood, we simplify notation by writing $V_\sigma(k) = V_\sigma(k,\beta)$ and $M_\sigma(k) = M_\sigma(k,\beta)$.

It is also convenient to introduce some auxiliary parameters:

$$\begin{aligned} \phi_l &= -(1-\nu)\rho\beta \\ \phi_c &= 1-\beta+((1-\nu)+(1-\mu))\rho\beta \\ \phi_r &= -(1-\mu)\rho\beta \end{aligned} \tag{2.11}$$

We note the signs of these parameters and various combinations:

$$\begin{aligned} \phi_l < 0 \quad , \qquad \phi_c > 0 \quad , \qquad \phi_r < 0 \\ \phi_l+\phi_c+\phi_r > 0 \quad , \quad \phi_l+\phi_c > 0 \quad , \quad \phi_r+\phi_c > 0 \end{aligned} \tag{2.12}$$

Using these auxiliary parameters and the recursion relations for V_σ and performing some simple algebraic manipulations yields a useful matrix representation involving marginals that we will use frequently:

$$\begin{bmatrix} \phi_c & \phi_r & 0 & \cdots & 0 \\ \phi_l & \phi_c & \phi_r & 0 & \vdots \\ 0 & \phi_l & \phi_c & \phi_r & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & \phi_l & \phi_c \end{bmatrix}_{K\times K} \begin{bmatrix} M_\sigma(0) \\ M_\sigma(1) \\ \vdots \\ M_\sigma(K_1-1) \end{bmatrix} = \begin{bmatrix} (1-\nu)\rho b \\ 0 \\ \vdots \\ 0 \\ (1-\mu)\rho c \end{bmatrix} \tag{2.13}$$

In short form,write this matrix representation as

$$\Phi\mathbf{M} = \mathbf{u} \tag{2.14}$$

Lemma 1. *Fix ρ, b, c, μ, ν and β. Let σ be a Markov strategy with the property that*

$$\sigma(k) = \begin{cases} 1 & \text{if} \quad 0 \le k < K_1 \\ 0 & \text{if} \quad K_1 \le k < K_2 \end{cases}$$

Then:

(i) if $0 \leq k < K_2$ *then* $M_\sigma(k) > 0$

(ii) in the range $0 \leq k < K_1$, M_σ *is either increasing, decreasing or decreasing then increasing*

(iii) if $M_\sigma(K_1 - 1) \geq c/\beta$ *then*

$$M_\sigma(0) > M_\sigma(1) > \ldots > M_\sigma(K_1 - 2) \geq M_\sigma(K_1 - 1) \geq c/\beta$$

Proof. We first consider the token holding levels $0 \leq k < K_1$. We make use of the matrix representation (2.13).

To prove (i), we first show that $M_\sigma(k) > 0$ for $0 \leq k < K_1$. If $K_1 < 3$, this follows by simply solving the matrix representation, so we henceforward assume $K_1 \geq 3$. If there exists a token holding level k^* with $0 \leq k^* < K_1$ such that $M_\sigma(k^*) \leq 0$ then one of the following must hold: either (a) there two consecutive such token holding levels, or (b) the marginal payoffs of the neighboring token holding levels are both positive. We consider these cases separately.

(a) In this case, there exists k^* such that $M_\sigma(k^*), M_\sigma(k^* + 1)$ are both non-positive. Of these, one is at least as big; say $M_\sigma(k^*) \geq M_\sigma(k^* + 1)$. From the identities above we see that

$$\begin{aligned} M_\sigma(k^* + 2) &= \frac{\phi_l M_\sigma(k^*) + \phi_c M_\sigma(k^* + 1)}{-\phi_r} \\ &\leq \frac{(\phi_l + \phi_c) M_\sigma(k^* + 1)}{-\phi_r} \\ &\leq M_\sigma(k^* + 1) \end{aligned}$$

Proceeding inductively, it follows that

$$0 \geq M_\sigma(k^*) \geq M_\sigma(k^* + 1) \geq \ldots \geq M_\sigma(K_1 - 1)$$

Moreover,

$$\begin{aligned} \phi_c M_\sigma(K_1 - 1) &= (1 - \mu)\rho c - \phi_l M_\sigma(K_1 - 2) \\ &> -\phi_l M_\sigma(K_1 - 2) \\ &> -\phi_l M(K - 1) \end{aligned}$$

This requires $\phi_c < -\phi_l$ which contradicts the sign relations (2.12). The argument when $M_\sigma(k^*+1) \geq M_\sigma(k^*)$ is similar and is left for the reader.

(b) In this case, there exists k^* such that $M_\sigma(k^*-1) > 0$, $M_\sigma(k^*) \leq 0$, $M_\sigma(k^*+1) > 0$. This entails

$$\phi_l M_\sigma(k^*-1) + \phi_c M_\sigma(k^*) + \phi_r M_\sigma(k^*+1) < 0 \tag{2.15}$$

which again contradicts the sign relations (2.12).

From the above we conclude $M_\sigma(k) > 0$ for $0 < k < K_1$. To see that $M_\sigma(0) > 0$ note that

$$-\phi_r M_\sigma(1) = \phi_c M_\sigma(0) - (1-\nu)\rho b < -\phi_r M_\sigma(0) \tag{2.16}$$

Therefore, $M_\sigma(1) < M_\sigma(0)$, so $M_\sigma(0) > 0$, as desired.

Finally, to see that $M_\sigma(k) > 0$ for $K_1 \leq k < K_2$, apply the recursion equations (2.9) to obtain

$$\phi_l M_\sigma(k-1) + (\phi_c + \phi_r) M_\sigma(k) = 0 \tag{2.17}$$

We know that $M_\sigma(K_1 - 1) > 0$ so the sign relations (2.12) imply that $M_\sigma(K_1) > 0$ as well. Now it follows inductively that $M_\sigma(k) > 0$ for $K_1 \leq k < K_2$. This completes the proof of (i).

To prove (ii) it is enough to show that M_σ has no local maximum for $0 < k < K_1$. If M had a local maximum k^* in this range we would have $M_\sigma(k^*) \geq M_\sigma(k^*-1)$ and $M_\sigma(k^*) \geq M_\sigma(k^*+1)$. However, algebraic manipulation yields the inequalities

$$\begin{aligned} M_\sigma(k^*) &= \frac{-\phi_l M_\sigma(k^*-1) - \phi_r M_\sigma(k^*+1)}{\phi_c} \\ &\leq \frac{-\phi_l - \phi_r}{\phi_c} M_\sigma(k^*) \\ &< M_\sigma(k^*) \end{aligned}$$

which is a contradiction. This establishes (ii)

To prove (iii), first manipulate the matrix identity (2.13) to obtain:

$$\begin{aligned}
&(1-\nu)\rho\beta M_\sigma(K_1-2)\\
&=(1-\beta+((1-\nu)+(1-\mu))\rho\beta)M_\sigma(K-1)-(1-\mu)\rho c\\
&\geq(1-\beta+(1-\nu)\rho\beta)M_\sigma(K_1-1)\geq(1-\nu)\rho\beta M_\sigma(K_1-1)
\end{aligned} \tag{2.18}$$

In view of (ii), the marginal payoffs are decreasing, so this establishes (iii). □

Lemma 2. *Fix ρ, b, c and a threshold protocol $\Pi=(\alpha,\sigma_K)$ with corresponding μ^Π,ν^Π. The marginal utility $M_{\sigma_K}(k,\beta)$ is strictly increasing in the discount factor β, i.e., if $0\leq\beta_1<\beta_2<1$, then,*

$$M_{\sigma_K}(k,\beta_1)<M_{\sigma_K}(k,\beta_2)\quad \textit{for all}\ \ k \tag{2.19}$$

Proof. To economize slightly on notation we write $\sigma=\sigma_K$. We present the proof in three steps.

In Step 1, we prove that if there exist $0<K_1\leq K_2<K-1$ such that $\forall k\in[K_1,K_2], M_\sigma(k,\beta_1)\geq M_\sigma(k,\beta_2)$, then at least one of the following is true, $M_\sigma(K_1-1,\beta_1)\geq M_\sigma(K_1-1,\beta_2)$ or $M_\sigma(K_2+1,\beta_1)\geq M_\sigma(K_2+1,\beta_2)$.

In Step 2, we prove that if there exists a $k^*\in[0,K-1]$ such that $M_\sigma(k^*,\beta_1)\geq M_\sigma(k^*,\beta_2)$, then for all $k\in[0,K-1]$, $M_\sigma(k,\beta_1)\geq M_\sigma(k,\beta_2)$. Step 2 uses the result of Step 1.

In Step 3, we disprove the possibility that $k\in[0,K-1]$, $M_\sigma(k,\beta_1)\geq M_\sigma(k,\beta_2)$.

Step 2 and Step 3 together show a contradiction and therefore, $k\in[0,K-1]$, $M_\sigma(k,\beta_1)<M_\sigma(k,\beta_2)$.

Step 1 We assert that if there are indices $0<K_1\leq K_2<K-1$ such that $M_\sigma(k,\beta_1)\geq M_\sigma(k,\beta_2)$ for all $K_1\leq k\leq K_2$ then at least one of the following must hold:

(A) $M_\sigma(K_1-1,\beta_1)\geq M_\sigma(K_1-1,\beta_2)$

(B) or $M_\sigma(K_2+1,\beta_1)\geq M_\sigma(K_2+1,\beta_2)$.

To see this, note that simple manipulations of the matrix representation (2.13) yield

- if $K_2 = K_1$ then

$$\begin{aligned}(1-\nu)\rho M_\sigma(K_1-1,\beta) &+ (1-\mu)\rho M_\sigma(K_2+1,\beta)\\ &= (1/\beta - 1 + ((1-\nu)+(1-\mu))\rho) M_\sigma(K_1,\beta)\end{aligned}$$

- if $K_2 > K_1$ then

$$\begin{aligned}(1-\nu)\rho M_\sigma(K_1-1,\beta) &+ (1-\mu)\rho M_\sigma(K_2+1,\beta)\\ &= (1/\beta - 1 + (1-\mu)\rho) M_\sigma(K_1,\beta)\\ &= +(1/\beta - 1)[M_\sigma(K_1+1,\beta) + ... + M_\sigma(K_2-1,\beta)]\\ &= +(1/\beta - 1 + (1-\nu)\rho) M_\sigma(K_2,\beta)\end{aligned}$$

Since $\beta_1 < \beta_2$ and we have assumed $M_\sigma(k,\beta_1) \geq M_\sigma(k,\beta_2)$ for $0 < K_1 \leq K_2 < K-1$, in each of the cases above the right-hand side is larger when $\beta = \beta_1$ than when $\beta = \beta_2$. Because the terms in the left-hand sides are positive, it follows that at least one of (A), (B) must hold, as asserted.

Step 2 We assert first that if there is a $k^*, 0 \leq k^* \leq K_1$ such that $M_\sigma(k^*,\beta_1) \geq M_\sigma(k^*,\beta_2)$, then at least one of the following must hold:

(C) there exists some K_3, $0 \leq K_3 \leq K_1$, such that $M_\sigma(k,\beta_1) \geq M_\sigma(k,\beta_2)$ for all k, $0 \leq k \leq K_3$

(D) there exists some K_4, $0 \leq K_4 \leq K_1$, such that $M_\sigma(k,\beta_1) \geq M_\sigma(k,\beta_2)$ for all k, $K_4 \leq k \leq K\_1$

To see this, note first that if $k^* = 0$ satisfies the hypothesis, then (C) holds with $K_3 = 0$ and that if $k^* = K-1$ satisfies the hypothesis, then (D) holds with $K_4 = K-1$. Hence it suffices to consider a k^*, $0 < k^* < K-1$, that satisfies the hypothesis. We now make use of Step 1. Set $K_1 = K_2 = k^*$. Applying Step 1 once increases the token holding interval where $M_\sigma(k,\beta_1) \geq M_\sigma(k,\beta_2)$ by 1. Let K_1 and K_2 be the new end points of the interval and apply Step 1 again. Continuing in this way we come eventually to a point where either $K_1 = 0$ or $K_2 = K-1$. If $K_1 = 0$, set $K_3 = K_2$ and note that (C) holds. If $K_2 = K-1$, set $K_4 = K-1$ and note that (D) holds

We now show that either (C) or (D) leads to the desired conclusion. Consider (C) first. Using the matrix representation (2.13) we obtain

$$\begin{aligned}(1-\nu)\rho\beta M_\sigma(K_1+1,\beta) \;+\; & (1-\nu)\rho b \\ = \;& [1-(1-(1-\mu)\rho)\beta]M_\sigma(0,\beta) \\ & + (1-\beta)[M_\sigma(1,\beta)+...+M_(K_1-1,\beta)] \\ & + [1-(1-(1-\nu)\rho)\beta]M_\sigma(K_1,\beta)\end{aligned}$$

The right-hand side is bigger when $\beta=\beta_1$ than when $\beta=\beta_2$. Therefore $M_\sigma(K_1+1,\beta_1)\geq M_\sigma(K_1+1,\beta_2)$. By induction, $M_\sigma(k,\beta_1)\geq M_\sigma(k,\beta_2)$ for all k, $0\leq k\leq K-1$.

Now consider (D). Using the matrix representation (2.13) we obtain

$$\begin{aligned}(1-\mu)\rho\beta M_\sigma(K_2-1,\beta) \;+\; & (1-\mu)\rho c \\ = \;& [1-(1-(1-\nu)\rho)\beta]M_\sigma(K-1,\beta) \\ & + (1-\beta)[M_\sigma(K-2,\beta)+...+M_\sigma(K_2+1,\beta)] \\ & + [1-(1-(1-\mu)\rho)\beta]M_\sigma(K_2,\beta)\end{aligned}$$

The right-hand side is bigger when $\beta=\beta_1$ than when $\beta=\beta_2$. Therefore $M_\sigma(K_2-1,\beta_1)\geq M_\sigma(K_2-1,\beta_2)$. By induction, $M_\sigma(k,\beta_1)\geq M_\sigma(k,\beta_2)$ for all k, $0\leq k\leq K-1$.

Taking (C) and (D) together completes Step 2.

Step 3 Using the matrix representation (2.13) we obtain

$$\begin{aligned}& [1-(1-(1-\mu)\rho)\beta]M_\sigma(0,\beta) \\ & +(1-\beta)[M_\sigma(1,\beta)+...+M_\sigma(K_1-1,\beta)] \\ & +[1-(1-(1-\nu)\rho)\beta]M_\sigma(K-1,\beta) \\ = \;& (1-\nu)\rho b+(1-\mu)\rho c\end{aligned}$$

In view of Step 2, the left-hand side is bigger when $\beta=\beta_1$ than when $\beta=\beta_2$. However, the right-hand side is independent of β, so this is a contradiction. We conclude that $M_\sigma(k,\beta_1)<M_\sigma(k,\beta_2)$ for every k, $0\leq k\leq K-1$. □

Proof. of Theorem 2 Fix β. The Markov strategy σ is optimal if and only if it satisfies the Bellman optimality conditions:

$$\beta(V_\sigma(k+1) - V_\sigma(k)) \geq c, \text{if } \sigma(k) = 1 \tag{2.20}$$

$$\beta(V_\sigma(k+1) - V_\sigma(k)) \leq c, \text{if } \sigma(k) = 0 \tag{2.21}$$

If σ is not a threshold strategy, there must exist integers $K_1 < K_2$ such that

$$\begin{aligned} \sigma(k) &= 1, \quad 0 \leq k < K_1 \\ \sigma(k) &= 0, \quad K_1 \leq k < K_2 \\ \sigma(k) &= 1, \quad k = K_2 \end{aligned} \tag{2.22}$$

We will show that the Bellman optimality conditions are violated at K_2 and $K_2 - 1$. To this end, let K_3 be the smallest integer greater than K_2 for which $\sigma(K_3) = 0$. (Such an integer exists because it cannot be optimal to serve when the token holding is sufficiently high.) Thus $\sigma(k) = 1$, for $K_2 \leq k < K_3$ and $M_\sigma(K_3 - 1) \geq c/\beta$. Following σ,

$$M_\sigma(K_3 - 2) = [(1-\mu)\rho c - \phi_c M_\sigma(K_3 - 1)]/\phi_l > M_\sigma(K_3 - 1) \geq c/\beta \tag{2.23}$$

An inductive argument shows that $M_\sigma(K_2) > M_\sigma(K_2+1) \geq c/\beta$. According to the recursion equations (2.9) we have

$$M_\sigma(K_2 - 1) = (\phi_c M_\sigma(K_2) + \phi_r M_\sigma(K_2 + 1))/(-\phi_l) > c/\beta$$

which is a contradiction. We conclude that a non-threshold strategy cannot be optimal; equivalently, only threshold strategies can be optimal strategies.

It remains to show that the only possible optimal threshold strategies have adjacent thresholds. Consider first two threshold strategies with consecutive thresholds K and $K+1$. We assert that

$$M_{\sigma_K}(K) < c/\beta \Leftrightarrow M_{\sigma_{K+1}}(K) < c/\beta \tag{2.24}$$

We prove direction "$\Rightarrow$"; the "$\Leftarrow$" direction is similar and left to the reader. Suppose instead that $M_{\sigma_{K+1}}(K) \geq c/\beta$. It follows that $-\phi_r M_{\sigma_{K+1}}(K) \geq (1-\mu)\rho c$. If we delete the last line

in the matrix equation (2.13) for σ_{K+1} and move $M_{\sigma_{K+1}}(K)$ to the right-hand side, we get another matrix equation

$$\Phi_{K\times K}\mathbf{M}_{\sigma_{K+1}} = \tilde{\mathbf{u}}$$

where $\tilde{\mathbf{u}} = ((1-\nu)\rho b, 0, ..., 0, -\phi_r M_{\sigma_{K+1}}(K))^T$. For the threshold K, $\Phi_{K\times K}\mathbf{M}_{\sigma_K} = \mathbf{u}$. Therefore,

$$\Phi_{K\times K}(\mathbf{M}_{\sigma_{K+1}} - \mathbf{M}_{\sigma_K}) = \tilde{\mathbf{u}} - \mathbf{u} \tag{2.25}$$

Lemma 1 guarantees that $\tilde{\mathbf{u}} - \mathbf{u} \geq 0$, so $\mathbf{M}_{\sigma_{K+1}} \geq \mathbf{M}_{\sigma_K}$. That is, $M_{\sigma_{K+1}}(k) \geq M_{\sigma_K}(k)$ for $0 \leq k \leq K-1$. Because $M_{\sigma_{K+1}}(K) \geq c/\beta > M_{\sigma_K}(K)$, it follows that $M_{\sigma_{K+1}}(k) \geq M_{\sigma_K}(k)$ for $0 \leq k \leq K$. According to the matrix equation, the following identity holds for both $\sigma = \sigma_K$ and $\sigma = \sigma_{K+1}$:

$$\begin{aligned} &(1-\nu)\rho b + (1-\mu)\rho c \\ =&(1-\beta+(1-\mu)\rho\beta)M_\sigma(0) \\ +&(1-\beta)\sum_{k=1}^{K-1} M_\sigma(k) + (1-\beta+(1-\nu)\rho\beta)M_\sigma(K) \end{aligned} \tag{2.26}$$

This is a contradiction so we have established the direction $\Rightarrow$, as desired.

It follows directly from the matrix identity that

$$M_{\sigma_K}(K) = c/\beta \Leftrightarrow M_{\sigma_{K+1}}(K) = c/\beta$$

Hence

$$M_{\sigma_K}(K) > c/\beta \Leftrightarrow M_{\sigma_{K+1}}(K) > c/\beta \tag{2.27}$$

We now assert that if $\tilde{K} > K$ then

$$M_{\sigma_K}(K) < c/\beta \Rightarrow M_{\sigma_{\tilde{K}}}(\tilde{K}-1) < c/\beta \tag{2.28}$$

We have already shown that this is true when $\tilde{K} = K+1$; i.e. $M_{\sigma_{K+1}}(K) < c/\beta$. Consider $\tilde{K} = K+2$. Of $M_{\sigma_{K+2}}(K+1) \geq c/\beta$, then (2.27) implies that $M_{\sigma_{K+1}}(K+1) \geq c/\beta$. Therefore, $M_{\sigma_{K+1}}(K+1) > M_{\sigma_{K+1}}(K)$. This is a contradiction to $M_{\sigma_{K+1}}(K+1) < M_{\sigma_{K+1}}(K)$. Following inductively we obtain the assertion (2.28).

A similar argument (which we omit) shows that:

$$M_{\sigma_K}(K-1) > c/\beta \Rightarrow M_{\sigma_{\tilde{K}}}(\tilde{K}) > c/\beta, \forall \tilde{K} < K \tag{2.29}$$

Finally, suppose σ_K is an optimal threshold strategy. Then $M_{\sigma_K}(K-1) \geq c/\beta$ and $M_{\sigma_K}(K) \leq c/\beta$. If the equalities hold strictly, (2.28) and (2.29) guarantee that σ_K is the only optimal threshold strategy. If $M_{\sigma_K}(K-1) = c/\beta$ (and hence, $M_{\sigma_K}(K) < c/\beta$), only σ_K and σ_{K-1} are optimal threshold strategies. If $M_{\sigma_K}(K) = c/\beta$ (and hence, $M_{\sigma_K}(K-1) > c/\beta$), only σ_K and σ_{K+1} are optimal threshold strategies. This completes the proof. □

Proof. of Theorem 3 This follows immediately from the representation of η_+ and the definition of invariance . □

Proof. of Theorem 4 Given a protocol $\Pi = (\alpha, \sigma)$, let η^Π be the unique invariant distribution; let μ^Π be the fraction of agents who have no tokens and ν^Π the fraction of agents who do not provide service; these depend only on Π and not on the population parameters. If $\sigma = \sum \gamma(K)\sigma_K$ is a best response given the population parameters and μ^Π, ν^Π, γ must put strictly positive weight only on threshold strategies σ_K that are pure best responses. In view of Theorem 2, there are at most two threshold strategies that are pure best responses and they are at adjacent thresholds. That is, σ is either a pure threshold strategy or a mixture of two adjacent threshold strategies, as asserted. □

Proof. of Theorem 5 Suppose to the contrary that $\Pi = (\alpha, \sigma)$ is a robust equilibrium protocol and that $\sigma = \sum \gamma(K)\sigma_K$ is a proper mixed strategy, so that $\gamma(K) > 0$ for at least two values of the threshold K, Let μ^Π be the fraction of agents who have no tokens and ν^Π the fraction of agents who do not provide service; these depend only on Π and not on the population parameters. In view of Theorem 4, σ must assign positive probability only to two adjacent threshold strategies; say $\sigma = \gamma(K)\sigma_K + \gamma(K+1)\sigma_{K+1}$ with $\gamma(K) > 0$ and $\gamma(K+1) > 0$, and both σ_K, σ_{K+1} must be best responses. Because $\sigma_K(K+1) = 0$ and $\sigma_{K+1}(K+1) = 1$, equations (8), (9) (which provide necessary and sufficient conditions for

optimality in terms of the *true* value function) entail that

$$\begin{aligned} -c + \beta V_{K+1} &\leq \beta V_K \\ -c + \beta V_{K+1} &\geq \beta V_K \end{aligned}$$

Hence $-c + \beta V_{K+1} = \beta V_K$. Because σ_K is a best response, the value functions V_{σ_K} must coincide with the true value function V. Hence, an agent following σ_K must be indifferent to providing service when holding K tokens. However, if β increases slightly M_{σ_K} also increases, whence an agent following σ_K must strictly prefer to provide service. In other words, when β increases slightly, σ_K can no longer be a best response and σ_K can no longer be an equilibrium protocol. This is a contradiction, so we conclude that a robust equilibrium protocol Π cannot involve proper mixed strategies, as asserted. □

Proof. of Theorem 6 We divide the proof of (i) into several steps.

Step 1 We first prove there exists $\beta^L \in [0, 1)$ such that

$$\begin{aligned} M_\sigma(K-1, \beta) &< \frac{c}{\beta} \text{ for } \beta < \beta^L \\ M_\sigma(K-1, \beta^L) &= \frac{c}{\beta} \\ M_\sigma(K-1, \beta) &> \frac{c}{\beta} \text{ for } \beta > \beta^L \end{aligned}$$

To see this, define the auxiliary function

$$F(\beta) = M_\sigma(K-1, \beta) - \frac{c}{\beta}$$

F is evidently continuous. Lemma 2 guarantees that $M_\sigma(K-1, \beta)$ is strictly increasing in β, so $F(\beta)$ is also strictly increasing in β as well. We show that $F(1) > 0$ and $\lim_{\beta \to 0} F(\beta) < 0$ and then apply the intermediate value theorem to find β^L.

To see that $F(1) > 0$, note first that the coefficients in the left-hand matrix of (2.13) are simply $\phi_l = -\rho(1-\nu)$, $\phi_c = \rho(1-\nu+1-\mu)$ and $\phi_r = \rho(1-\mu)$. We split the matrix $\mathbf{M}_{\sigma_K}$ in two parts. To do this, write

$$\begin{aligned} \mathbf{u}' &= (\rho(1-\nu)c \quad 0 \quad ... \quad 0 \quad \rho(1-\mu)c)^T \\ \mathbf{u}'' &= (\rho(1-\nu)(b-c) \quad 0 \quad ... \quad 0 \quad 0)^T \end{aligned} \tag{2.30}$$

and define $\mathbf{M'}_{\sigma_K}, \mathbf{M''}_{\sigma_K}$ to be the solutions to the equations

$$\Phi \mathbf{M'}_{\sigma_K} = \mathbf{u'}, \quad \Phi \mathbf{M''}_{\sigma_K} = \mathbf{u''} \tag{2.31}$$

Note that $\mathbf{M}_{\sigma_K} = \mathbf{M'}_{\sigma_K} + \mathbf{M''}_{\sigma_K}$ and $\mathbf{M}_{\sigma_K}$ is the solution to (2.13). It is easy to check that $\mathbf{M'}_{\sigma_K}$ is a constant matrix: $M'_{\sigma_K}(k) = c$ for $0 \leq k < K-1$. Lemma 1 guarantees that the entries of $\mathbf{M''}_{\sigma_K}$ are strictly positive: $M''_{\sigma}(k) > 0$ for $0 \leq k < K-1$. Hence the entries of $\mathbf{M}_{\sigma_K}$ are strictly greater than c: $M_{\sigma}(k) > c$ for$0 \leq k < K-1$. In particular, $F(1) > 0$.

To see that $\lim_{\beta \to 0} F(\beta) < 0$, suppose not. Because F is strictly increasing, this means $F(\beta) \geq 0$ for every $\beta \in (0, 1]$, which entails that $M_{\sigma}(k) \geq \frac{c}{\beta}$ for $0 \leq k < K-1$. Summing the rows in (2.13) yields:

$$\rho(1-\nu)b + \rho(1-\mu)c > K(1-\beta)\frac{c}{\beta} = \frac{Kc}{\beta} - Kc \tag{2.32}$$

Note that Kc/β flows up as $\beta \to 0$, so this is impossible. We conclude that $\lim_{\beta \to 0} F(\beta) < 0$, as asserted.

Because F is strictly increasing, the intermediate value theorem guarantees that we can find an unique β^L such that

$$\begin{aligned} F(\beta) &< 0 \quad \text{for} \quad \beta < \beta^L \\ F(\beta^L) &= 0 \\ F(\beta) &> 0 \quad \text{for} \quad \beta > \beta^L \end{aligned}$$

The definition of F yields the desired property of β^L

Step 2 Next we prove there exists $\beta^H \in (\beta^L, 1)$ such that if $\beta \in [0, \beta^H]$ then

$$\begin{aligned} M_{\sigma_K,\beta}(K-1) &< \frac{\phi_c + \phi_r}{-\phi_l}\frac{c}{\beta} \quad \text{for } \beta < \beta^H \\ M_{\sigma_K,\beta^H}(K-1) &= \frac{\phi_c + \phi_r}{-\phi_l}\frac{c}{\beta} \\ M_{\sigma_K,\beta}(K-1) &> \frac{\phi_c + \phi_r}{-\phi_l}\frac{c}{\beta} \quad \text{for } \beta > \beta^H \end{aligned}$$

To see this, note first that $\frac{\phi_c+\phi_r}{-\phi_l}\frac{c}{\beta} = \left[1 - \frac{1}{\rho(1-\nu)} + \frac{1}{\rho(1-\nu)\beta}\right]\frac{c}{\beta}$ and define another auxiliary function:

$$G(\beta) = M_{\Pi}(K-1, \beta) - (1 - \frac{1}{\rho(1-\nu)} + \frac{1}{\rho(1-\nu)\beta})\frac{c}{\beta}$$

G is continuous and increasing. From Step 1 it follows that $M_{\sigma_K}(K-1,1) > c$ so $G(1) = M_{\sigma_K}(K-1,1) - c > 0$. It also follows that $M_{\sigma_K}(K-1,\beta^L) = \frac{c}{\beta^L}$; because $(1 - \frac{1}{\rho(1-\nu)} + \frac{1}{\rho(1-\nu)\beta^L})\frac{c}{\beta^L} > \frac{1}{\beta^L}$, we conclude that $G(\beta^L) < 0$. Because G is continuous and increasing, there is a unique $\beta^H \in (\beta^L, 1)$ such that

$$\begin{aligned} G(\beta) &< 0 \quad \text{for} \quad \beta < \beta^H \\ G(\beta^H) &= 0 \\ G(\beta) &> 0 \quad \text{for} \quad \beta > \beta^L \end{aligned}$$

Step 3 The definitions of F, G imply that in order for Π to be an equilibrium protocol when the discount factor is β it is the necessary and sufficient condition that $F(\beta) \geq 0$ and $G(\beta) \leq 0$. Hence Π is an equilibrium protocol when the discount factor is β exactly for $\beta \in [\beta^L, \beta^H]$.

Because F, G are continuous in all their arguments and strictly increasing, β^L, β^H, which are the zeroes of F, G, are continuous functions of the parameters as well. This completes the proof of (i).

The proof of (ii) is similar and left to the reader. □

Proof. of Theorem 7 We first consider (i). Fix r. Consider the two protocols $\Pi_K = (K/2, \sigma_K)$ and $\Pi_{K+1} = ((K+1)/2, \sigma_{K+1})$ and the corresponding intervals $[\beta_1^L, \beta_1^H]$ and $[\beta_2^L, \beta_2^H]$ of discount factors that sustain equilibrium. We need to show that

$$\beta_1^L < \beta_2^L < \beta_1^H < \beta_2^H$$

(The sustainable ranges for two consecutive threshold protocols overlap but are not nested.) There are three inequalities to be established; we carry out the analyses in (A), (B), (C) below.

(A) To prove $\beta_2^L > \beta_1^L$, write $\beta = \beta_1^L$. We show that $M_{\sigma_{K+1}}(K) < \frac{c}{\beta}$. To see this, suppose not; i.e. $M_{\sigma_{K+1}}(K) \geq \frac{c}{\beta}$. The construction of β_1^L guarantees that $M_{\sigma_K}(K-1) = c/\beta$. We will use this inequality and equality to show that *all* marginal payoffs of Π_{K+1} so large that they violate the restrictions imposed by the bounded benefit b and cost c.

To simplify the notation, let $\omega_X = \frac{X+1}{X}(\frac{1}{\beta}-1)\frac{1}{\rho}$. Note $\omega_{K+1} < \omega_K$. Then the matrix identity (2.13) becomes:

$$\begin{bmatrix} \omega_X+2 & -1 & 0 & \cdots & 0 \\ -1 & \omega_X+2 & -1 & 0 & \vdots \\ 0 & -1 & \omega_X+2 & -1 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & -1 & \omega_X+2 \end{bmatrix}_{X\times X} \begin{bmatrix} M_{\sigma_X}(0) \\ M_{\sigma_X}(1) \\ \vdots \\ M_{\sigma_X}(X-1) \end{bmatrix} = \begin{bmatrix} b/\beta \\ 0 \\ \vdots \\ 0 \\ c/\beta \end{bmatrix} \tag{2.33}$$

Suppose $M_{\sigma_{K+1}}(K) \geq M_{\sigma_K}(K-1) = \frac{c}{\beta}$. We investigate the relation between $M_{\sigma_{K+1}}(K-1)$ and $M_{\sigma_K}(K-2)$. Using the matrix identity,

$$\begin{aligned} \frac{M_{\sigma_{K+1}}(K-1)}{M_{\sigma_K}(K-2)} &= \frac{(\omega_{K+1}+2)M_{\sigma_{K+1}}(K) - \frac{c}{\beta}}{(\omega_K+2)M_{\sigma_K}(K-1) - \frac{c}{\beta}} \\ &> \frac{(\omega_{K+1}+2)M_{\sigma_{K+1}}(K)}{(\omega_K+2)M_{\sigma_K}(K-1)} > \frac{\omega_{K+1}+1}{\omega_K+1} \end{aligned}$$

Moreover if $2 \leq k \leq K-1$ then

$$\frac{M_{\sigma_{K+1}}(K-k)}{M_{\sigma_K}(K-k-1)} = \frac{(\omega_{K+1}+1)[M_{\sigma_{K+1}}(K) + M_{\sigma_{K+1}}(K-k+1)] - \frac{c}{\beta}}{(\omega_K+1)[M_{\sigma_K}(K-1) + M_{\sigma_K}(K-k)] - \frac{c}{\beta}}$$

By induction,

$$\begin{aligned} \frac{M_{\sigma_{K+1}}(K-k)}{M_{\sigma_K}(K-k-1)} &> \left(\frac{\omega_{K+1}+1}{\omega_K+1}\right)^k > \left(\frac{\omega_{K+1}}{\omega_K}\right)^k > \left(1 - \frac{1}{(K+1)^2}\right)^k \\ &> 1 - \frac{k}{(K+1)^2} > \frac{K+1}{K+2}, \forall 0 \leq k \leq K-1 \end{aligned}$$

Next we prove $M_{\sigma_{K+1}}(0) \geq M_{\sigma_K}(0)$. This is relatively easy since, if $M_{\sigma_{K+1}}(0) < M_{\sigma_K}(0)$, then using the marginal payoff matrix and by induction, $M_{\sigma_{K+1}}(K-1) < M_{\sigma_K}(K-1) = \frac{c}{\beta}$. This is a contradiction to $M_{\sigma_{K+1}}(K-1) > M_{\sigma_{K+1}}(K) = \frac{c}{\beta}$. Therefore, $M_{\sigma_{K+1}}(0) \geq M_{\sigma_K}(0)$.

The marginal payoffs are bounded as follows,

$$(M_{\sigma_X}(0) + M_{\sigma_X}(X-1)) + \omega_X \sum_{k=0}^{X-1} M_{\sigma_X}(k) = b/\beta + c/\beta \tag{2.34}$$

However, since

$$\begin{aligned}\omega_{K+1}\sum_{k=0}^{K} M_{\sigma_{K+1}}(k) &> \frac{K+1}{K}\omega_{K+1}\sum_{k=1}^{K} M_{\sigma_{K+1}}(k)\\ &> \frac{K+1}{K}\frac{K(K+2)}{(K+1)^2}\frac{K+1}{K+2}\omega_K\sum_{k=0}^{K-1} M_{\sigma_K}(k)\\ &= \omega_K\sum_{k=0}^{K-1} M_{\sigma_K}(k)\end{aligned}$$

and $M_{\sigma_{K+1}}(0) + M_{\sigma_{K+1}}(K) > M_{\sigma_K}(0) + M_{\sigma_K}(K-1)$, a contradiction occurs. Therefore, for $\beta = \beta_1^L$, $M_{\sigma_{K+1}}(K) < \frac{c}{\beta}$. This means $\beta_2^L > \beta_1^L$. This completes (A).

(B) To prove $\beta_2^H > \beta_1^H$, let $\beta = \beta_1^H$, we need to show that the protocol Π_{K+1} must have $M_{\sigma_{K+1}}(K+1) < c/\beta$. We use contradiction to prove this. The idea is: Suppose $M_{\sigma_{K+1}}(K+1) \geq c/\beta$, then we show that all the marginal payoffs of Π_{K+1} are large enough such that they violate the restriction imposed by the bounded benefit b and cost c.

Suppose $M_{\sigma_{K+1}}(K+1) \geq M_{\sigma_K}(K) = c/\beta$. According to the matrix equation, similar to part (A), by induction we can get,

$$\frac{M_{\sigma_{K+1}}(K+1-k)}{M_{\sigma_K}(K-k)} > \left(\frac{\omega_{K+1}+1}{\omega_K+1}\right)^k > \frac{(K+1)^3}{K(K+2)^2}, \forall 0 \leq k \leq K$$

Also $M_{\sigma_{K+1}}(0) \geq M_{\sigma_K}(0)$. The marginal payoffs are bounded as follows,

$$(M_{\sigma_X}(0) + M_{\sigma_X}(X)) + \omega_X\sum_{k=0}^{X} M_{\sigma_X}(k) = b/\beta + c/\beta \tag{2.35}$$

However, since

$$\begin{aligned}\omega_{K+1}\sum_{k=0}^{K+1} M_{\sigma_{K+1}}(k) &> \frac{K+2}{K+1}\omega_{K+1}\sum_{k=1}^{K+1} M_{\sigma_{K+1}}(k)\\ &> \frac{K+2}{K+1}\frac{K(K+2)}{(K+1)^2}\frac{(K+1)^3}{K(K+2)^2}\omega_K\sum_{k=0}^{K} M_{\sigma_K}(k)\\ &= \omega_K\sum_{k=0}^{K} M_{\sigma_K}(k)\end{aligned}$$

and $M_{\sigma_{K+1}}(0) + M_{\sigma_{K+1}}(K+1) > M_{\sigma_K}(0) + M_{\sigma_K}(K)$, a contradiction occurs. Therefore, for $\beta = \beta_1^H$, $M_{\sigma_{K+1}}(K+1) < \frac{c}{\beta}$. This means $\beta_2^H > \beta_1^H$. This completes part (B).

(C) To prove $\beta_2^L < \beta_1^H$, wreite $\beta = \beta_1^H$. We show that $M_{\sigma_{K+1}}(K) > M_{\sigma_K}(K) = \frac{c}{\beta}$. If not, then as in (A) we must have $M_{\sigma_{K+1}}(K) \leq M_{\sigma_K}(K) = \frac{c}{\beta}$; in that case we show $M_{\sigma_{K+1}}(k) \leq M_{\sigma_K}(k)$ for $0 \leq k \leq K$. This will again violate the restrictions imposed by b and c.

We extend the marginal payoff matrix in (2.33) from $K \times K$ to $(K+1) \times (K+1)$ and incorporate $M_{\sigma_K}(K)$. If $M_{\sigma_K}(K) = \frac{c}{\beta}$, such extension does not change the solution of the marginal payoffs $M_{\sigma_K}(k), \forall k \in [0, K]$. Note the new coefficient matrix has the same size of the coefficient matrix for σ_{K+1}. Suppose $M_{\sigma_{K+1}}(K) < M_{\sigma_K}(K) = \frac{c}{\beta}$. According to the matrix equation,

$$\frac{M_{\sigma_{K+1}}(K-1)}{M_{\sigma_K}(K-1)} = \frac{(\omega_{K+1}+2)M_{\sigma_{K+1}}(K) - c/\beta}{(\omega_K+2)M_{\sigma_K}(K) - c/\beta} < 1$$

Moreover, for $0 \leq k \leq K$ we have

$$\frac{M_{\sigma_{K+1}}(K-k)}{M_{\sigma_K}(K-k)} = \frac{(\omega_{K+1}+1)[M_{\sigma_{K+1}}(K) + M_{\sigma_{K+1}}(K-k+1)] - c/\beta}{(\omega_K+1)[M_{\sigma_K}(K) + M_{\sigma_K}(K-k+1)] - c/\beta}$$

By induction, $M_{\sigma_{K+1}}(k) < M_{\sigma_K}(k)$ $0 \leq k \leq K$. However, since

$$(M_{\sigma_X}(0) + M_{\sigma_X}(X-1)) + \omega_X \sum_{k=0}^{X-1} M_{\sigma_X}(k) = b/\beta + c/\beta \tag{2.36}$$

Again, the left-hand side is bigger when $X = K$ than when $X = K+1$, which is a contradiction. This completes part (C).

Combining (A), (B) and (C) establishes the desired string of inequalities. The remaining conclusions of (i) follow immediately.

The argument for (ii) is very similar and left to the reader. □

Proof. of Theorem 8 Fix a protocol $\Pi = (\alpha, \sigma_K)$ and let η^Π be the corresponding invariant distribution. We first find a closed form expression for η^Π. To do this, plug the strategy σ_K into the characterization of the invariant distribution given in Theorem 3. A little algebra provides an identify involving $\eta^\Pi(0), \eta^\Pi(1), \eta^\Pi(K)$ and a simpler recursion relationship.

$$\eta^\Pi(1) = \left[\frac{1-\eta^\Pi(0)}{1-\eta^\Pi(K)}\right]\eta^\Pi(0)$$

$$\eta^\Pi(k) = \left[\frac{2-\eta^\Pi(0)-\eta^\Pi(K)}{1-\eta^\Pi(K)}\right]\eta^\Pi(k-1) + \left[\frac{1-\eta^\Pi(0)}{1-\eta^\Pi(K)}\right]\eta^\Pi(0)\eta^\Pi(k-2)$$

From this we can solve recursively, obtaining

$$\eta^{\Pi}(k) = \left[\frac{1-\eta^{\Pi}(0)}{1-\eta^{\Pi}(K)}\right]^k \tag{2.37}$$

for all $k = 0, 1, \ldots, K$. Note that the one remaining degree of freedom is pinned down by the requirement that the total token holding be equal to α.

We next solve the following simple maximization problem:

$$\begin{array}{ll} \underset{0\le x_1,x_2\le 1}{\text{maximize}} & E^*(x_1,x_2) = 1 - x_1 - x_2 + x_1x_2 \\ \text{subject to} & x_1(1-x_1)^K = x_2(1-x_2)^K \end{array} \tag{2.38}$$

To solve this problem, set $f(x) = x(1-x)^K$. A straightforward calculus exercise shows that if $0 \le x_1 \le \frac{1}{K+1} \le x_2 \le 1$ and $f(x_1) = f(x_2)$ then:

(a) $x_1 + x_2 \ge \frac{2}{K+1}$, with equality achieved at $x_1 = x_2 = \frac{1}{K+1}$.

(b) $x_1x_2 \le \frac{1}{K+1}$, with equality achieved at $x_1 = x_2 = \frac{1}{K+1}$.

Putting (a) and (b) together shows that the optimal solution to the maximization problem (2.38) is to have $x_1 = x_2 = \frac{1}{K+1}$ and $\max E^* = \left(1 - \frac{1}{K+1}\right)^2$.

If we take $x_1 = \mu^{\Pi}, x_2 = \nu^{\Pi}$ and apply the closed form solution (2.37) for the invariant distribution, we see that $f(x_1) = f(x_2)$. By definition, $\text{Eff}(\Pi) = E^*(x_1, x_2)$ so

$$\text{Eff}(\Pi) \le \max E^* = \left(1 - \frac{1}{K+1}\right)^2$$

On the other hand, if $\alpha = K/2$ then the invariant distribution has $\eta^{\Pi}(k) = \frac{1}{K+1}$ for all k and

$$\text{Eff}(K/2, \sigma_K) = \left(1 - \frac{1}{K+1}\right)^2 = [K/(K+1)]^2$$

Taken together, part (ii) and (iii) are proved..

Next fix a protocol (α, σ_K). Let $\lceil\alpha\rceil$ be the least integer greater than or equal to α and set $K^* = 2\lceil\alpha\rceil$. There are two cases to consider.

In the first case, $K \le K^*$.

$$\text{Eff}(\alpha, \sigma_K) \;\le\; \left(1 - \frac{1}{K+1}\right)^2 \le \left(1 - \frac{1}{K^*+1}\right)^2 = \left(1 - \frac{1}{2\lceil\alpha\rceil+1}\right)^2$$

which is the desired result in the first case.

In the second case, $K > K^*$. Define the protocol $\Pi' = (\lceil\alpha\rceil, \sigma_K)$; let η' be the invariant token distribution for Π'. Let $\Pi^* = (\lceil\alpha\rceil, \sigma_{K^*})$; note that the invariant token distribution η^* is uniform ($\eta^*(k) = \frac{1}{K^\star+1} = \frac{1}{2\lceil\alpha\rceil+1}$ for all $k = 0, 1, ..., K^*$). Note that Π' and Π have the same strategy component but that the token supply for Π' is larger than for Π, and that Π' and Π^* have the same token supply but that the strategy component of Π' has a higher threshold.

We assert that $\eta'(0) \geq \frac{1}{2\lceil\alpha\rceil+1}$. If not then $\eta'(0) < \frac{1}{2\lceil\alpha\rceil+1} = \frac{1}{K^\star+1}$. It follows that for all $k \in \{0, 1, ..., K\}$ we have $\eta'(k) < \frac{1}{K^*+1} = \eta^*(k)$. Hence

$$\begin{aligned}
\lceil\alpha_0\rceil =& \sum_{k=0}^{K^*} k\eta^*(k) = \sum_{k=0}^{K^*} k(\eta^*(k) - \eta'(k)) + \sum_{k=0}^{K^*} k\eta'(k) \\
\leq& K^* \sum_{k=0}^{K^*} (\eta^*(k) - \eta'(k)) + \sum_{k=0}^{K^*} k\eta'(k) = K^*(1 - \sum_{k=0}^{K^*} \eta'(k)) + \sum_{k=0}^{K^*} k\eta'(k) \\
=& K^* \sum_{k=K^*}^{K} \eta'(k) + \sum_{k=0}^{K^*} k\eta'(k) \leq \sum_{k=K^*}^{K} k\eta'(k) + \sum_{k=0}^{K^*} k\eta'(k) = \lceil\alpha_0\rceil
\end{aligned}$$

This is a contradiction. Hence, $\eta'(0) \geq \frac{1}{2\lceil\alpha\rceil+1}$.

Because the token supply for Π is less than Π', the number of agents with no tokens is larger, so $\eta(0) > \eta'(0) \geq \frac{1}{2\lceil\alpha\rceil+1}$. Hence

$$\text{Eff}(\Pi) = (1 - \eta(0))(1 - \eta(K)) < (1 - \eta(0)) < \left(1 - \frac{1}{2\lceil\alpha\rceil + 1}\right)$$

which is the desired result in the second case. This complete the proof for part (i). □

Proof. of Theorem 9 Both assertions follow immediately by combining Theorems 7 and 8. □

Proof. of Theorem 10 We first derive the lower bound K^L. If $\Pi_K = (K/2, \sigma_K)$ is an equilibrium protocol then consecutive marginal utilities bear the relationship

$$\phi_l M_{\sigma_K}(k-1) + \phi_c M_{\sigma_K}(k) = -\phi_r M_{\sigma_K}(k+1) > 0$$

(Because β is fixed, we suppress it in the notation.) Therefore, $M_{\sigma_K}(k) > \frac{-\phi_l}{\phi_c} M_{\sigma_K}(k-1)$. By induction,

$$M_{\sigma_K}(k) > \left(\frac{-\phi_l}{\phi_c}\right)^k M_{\sigma_K}(0) > \left(\frac{\rho\beta}{2(1-\beta)+2\rho\beta}\right)^k M_{\sigma_K}(0)$$

Because $\phi_c M_{\sigma_K}(0) = (1-\nu)\rho b - \phi_r M_{\sigma_K}(1) > (1-\nu)\rho b + (1-\nu)\rho c$, we have

$$M_{\sigma_K}(0) > \frac{(1-\nu)\rho b}{\phi_c} = \frac{\rho\beta}{2(1-\beta)+2\rho\beta}\frac{b+c}{\beta}$$

Therefore,

$$M_{\sigma_K}(k) > \left(\frac{\rho\beta}{2(1-\beta)+2\rho\beta}\right)^{k+1} \frac{b+c}{\beta} \tag{2.39}$$

Because Π_K is assumed to be an equilibrium protocol, we must have $M_{\sigma_K}(K) \leq c/\beta$. Moreover, we must also have

$$\left(\frac{\rho\beta}{2(1-\beta)+2\rho\beta}\right)^{K+1} \frac{b+c}{\beta} \leq \frac{c}{\beta}$$

because otherwise $M_{\sigma_K}(K) > c/\beta$. Therefore,

$$K \geq \max\{\log_{\frac{\rho\beta}{2(1-\beta)+2\rho\beta}} \frac{c}{b+c} - 1, 0\} \tag{2.40}$$

This provides the lower bound K^L.

We now derive the upper bound K^H. Rewriting the relation between consecutive marginal utilities we obtain

$$\begin{aligned} 0 &= \phi_l M_{\sigma_K}(k-1) + \phi_c M_{\sigma_K}(k) + \phi_r M_{\sigma_K}(k+1) \\ &> \phi_l M_{\sigma_K}(k-1) + (\phi_c + \phi_r) M_{\sigma_K}(k) \end{aligned}$$

Therefore, $M_{\sigma_K}(k) < \frac{-\phi_l}{\phi_c+\phi_r} M_{\sigma_K}(k-1)$. By induction,

$$M_{\sigma_K}(k) < \left(\frac{-\phi_l}{\phi_c+\phi_r}\right)^k M_{\sigma_K}(0) < \left(\frac{\rho\beta}{1-\beta+\rho\beta}\right)^k M_{\sigma_K}(0)$$

Because $\phi_c M_{\sigma_K}(0) = (1-\nu)\rho b - \phi_r M_{\sigma_K}(1) < (1-\nu)\rho b - \phi_r b/\beta = 2(1-\nu)\rho b$, we have,

$$M_{\sigma_K}(0) < \frac{\rho\beta}{1-\beta+\rho\beta}\frac{2b}{\beta}$$

Therefore,

$$M_{\sigma_K}(k) < \left(\frac{\rho\beta}{1-\beta+\rho\beta}\right)^{k+1} \frac{2b}{\beta} \tag{2.41}$$

Because Π_K is assumed to be an equilibrium protocol, we must have $M_{\sigma_K}(K-1) \geq c/\beta$. Moreover,

$$\left(\frac{\rho\beta}{1-\beta+\rho\beta}\right)^{K} \frac{2b}{\beta} \geq \frac{c}{\beta}$$

because otherwise $M_{\sigma_K}(K-1) < c/\beta$. Therefore,

$$K \leq \log_{\frac{\rho\beta}{1-\beta+\rho\beta}} \frac{c}{2b} \tag{2.42}$$

This provides the upper bound K^H.

Combining the two estimates yields the range containing all integers K for which Π_K is an equilibrium protocol. The estimate for efficiency follows immediately since $\text{Eff}(\Pi_K) \geq \text{Eff}(\Pi_{K^L})$ if $K \geq K^L$, so the proof is complete. □

CHAPTER 3

Sharing in Networks of Strategic Agents

In recent years, extensive research effort has been devoted to studying cooperative networks where autonomous agents interact repeatedly with each other over an exogenously given network by sharing information (such as measurements, estimates, beliefs, or opinions) or goods (such as endowments or production). These networks require various levels of coordinated behavior and cooperation among the autonomous agents. However, in many scenarios, participating in the cooperative process entails costs to the agents, such as the cost of producing, processing and transferring information/goods to their neighbors. If agents are strategic, they will choose to cooperate with other agents in the network only if cooperation maximizes their own long-term utilities, which take into account both current and future benefits. Absent incentives for cooperation, agents will free-ride and the networks will work inefficiently or even collapse [Ost08]. If a central authority existed in the network, which was omniscient about agents' utilities and actions as well as capable of computing and enforcing an efficient behavior profile for all the agents, the social optimum could be attained; but, in practice, such central authorities do not exist. On the contrary, agents usually possess only local information, and they act selfishly to maximize their own payoff. Hence, incentives are needed to compel the strategic agents to act in a socially optimal manner. Designing incentives for networks of strategic agents is significantly more challenging than in scenarios where agents are randomly matched [Kan92] [ZPS14] [XS12] [XZV12] or interact as independent pairs [Axe81] [MJS06], since the incentives of agents are complexly coupled based on the connectivity of agents. Moreover, effective implementation of an incentive scheme requires that it be distributed, which represents another key challenge. In this work we present the first scheme that solves these problems.

To better motivate this work, we provide two concrete application scenarios. Establishing a secure cyber environment requires investments on security technologies (e.g. firewalls, access control etc.) from autonomous systems (ASes). Improved security can be achieved if ASes deploy proactive protection technologies (e.g. outbound traffic control) which are more effective because ASes have better control over their own devices and traffic originating from their own users [XZS13]. However, ASes are self-interested and are reluctant to make security investment on these proactive technologies since doing that is not directly beneficial to themselves [XZS13]. The similar incentive problem also exists in joint spectrum sensing problems in cognitive radio systems [SZ09]. To enable dynamic spectrum access, the preliminary requirement is the ability to accurately identify the presence of primary users over a wide range of spectrum. With joint spectrum sensing, each secondary user senses the spectrum individually and then shares the raw sensing results to their neighbors at the beginning of each transmission slot to improve the detection probability in this slot. However, secondary users are self-interested and lack the incentives to send their sensing raw results to their neighbors which will cost extra resources such as energy and transmission time.

We resolve the above incentive problem by deploying a *distributed* rating protocol. The rating protocol consists of three components: a set of ratings, recommended strategies (for each agent) and rating update rules (for each agent). In each period, each agent is assigned a rating, which is maintained and updated according to the rating protocol. The actions recommended to the agents by the system (e.g. how much to share) depend on the ratings of their neighboring agents (i.e. the agents with whom they are directly connected). The recommendations can be determined in a distributed manner by the system. For example, each agent could be interacting with other agents through a software client (similar to BitTorrent). Each agent's software is then preprogrammed to recommend actions based on the local network structure it observes, information that is received from the software of neighboring agents and the current ratings of neighboring agents. Since agents are strategic and want to maximize their own utility, they have the freedom to decide their own actions and they may comply or not with the recommended actions. Based on the agent's current rating and whether it has followed/deviated from the recommended strategy, the software increas-

es/decreases an agent's rating. We design rating protocols (i.e. recommended strategies and rating update rules) that are incentive-compatible (i.e. agents have incentives to follow the recommended strategies) and maximize the social welfare (i.e. the sum of the utility of all agents).

There are two central challenges. The first arises from the fact that agents interact over a network. In particular, the agents' interactions are subject to network constraints, i.e. agents can only interact with their neighbors. This is in stark contrast with existing works in repeated games relying on social reciprocation which assume that the agents are randomly matched [Kan92] [ZPS14] [XZV12] or interact on a complete graph [SZ09]. Due to the network constraints, agents' incentives are coupled in a much more complex manner since they depend directly on the behavior of their immediate neighbors and indirectly on the behavior of more distant remaining agents. Because of the different network constraints, there is not a universal rating protocol that can work efficiently in all networks. Instead, the rating protocol design must explicitly take into account the specific coupling among agents arising from the specific network.

The second arises because we insist on protocols that are distributed and informationally decentralized. We do not need to assume the existence of any central entity that can monitor the entire network (i.e. network topology, all agents' utility functions and actions) and communicate to all individual agents about all other agents' behavior. Decentralization rules out protocols proposed in prior works [ZPS14] [XS12] since they are designed and implemented in a centralized manner, requiring the knowledge of the entire network at a central entity. In this work, the rating protocol is designed and implemented in a distributed manner, requiring only limited message exchange (i.e. Lagrangian multipliers during configuration and agents' ratings during interaction) among the software of neighboring agents.

The main contributions of this work are:

1. We develop a framework for providing incentives in networks where heterogeneous agents interact repeatedly over a network. This framework is very general and can be employed for a variety of applications, including in networks where bilateral interest

may not exist between agents and hence, existing works based on direct reciprocation such as Tit-for-Tat [Axe81] [MJS06] do not work.

2. We rigorously analyze the incentives (Theorem 1) of agents operating under the rating protocol framework using a novel repeated game with imperfect monitoring formalism, which explicitly considers the network structure, agents' utility functions etc. With these constraints and using the dual decomposition method, we propose a novel and fully distributed algorithm to compute the optimal recommended strategy of the rating protocol that maximizes the social welfare.

3. We show how different networks may affect agents' incentives in different ways and how to design rating protocols that are tailored to different networks. Modified rating protocols that apply to various dynamic networks are also proposed and analyzed.

3.1 Related Works

Cooperation among the agents (e.g. repeated sharing) is critical for the enhanced performance and robustness of various types of social, economic and engineering networks [KP13]. The main focus of this literature is on determining the resulting network performance if agents repeatedly share and process information/goods. However, absent incentives and in the presence of strategic agents, these networks will work inefficiently or even collapse [Ost08]. Thus, the main focus of the current work is how to incentivize strategic agents to cooperate such that networks can operate efficiently.

A variety of incentive schemes has been proposed to encourage cooperation among agents (see e.g. [PS10] for a review of different game theoretic solutions). Two popular incentive schemes are pricing and differential service. Pricing schemes [BO06] [MV95] use payments to reward and punish individuals for their behavior. However, they often require complex accounting and monitoring infrastructure, which introduces substantial communication and computation overhead. Differential service schemes, on the other hand, reward and punish individuals by providing differential services depending on their behavior. Differential ser-

vices can be provided by the network operator [Kan92] [Del05] [ZPS14]. However, in many networks of autonomous agents, such a centralized network operator does not exist. Alternatively, differential services can also be provided by the other agents participating in the network since agents in the considered applications derive their utility from their interactions with other agents [Axe81] [MJS06] [SZ09] [Kan92] [MA09] [JRT12] [ZPS14] [XZV12]. Such incentive schemes are based on the principle of reciprocity and can be classified into direct (personal) reciprocation and social reciprocation. In direct (personal) reciprocation schemes (e.g. the widely adopted Tit-for-Tat strategy [Axe81] [MJS06]), the behavior of an individual agent toward another is based on its personal experience with that agent. However, they only work when two interacting agents have bilateral interests. In social reciprocation schemes [SZ09] [Kan92] [MA09] [JRT12] [ZPS14] [XZV12], individual agents obtain some (public) information about other individuals (e.g. their ratings) and decide their behavior toward other agents based on this information.

Incentive mechanisms based on social reciprocation are often studied using the familiar framework of repeated games. In [SZ09], the sharing game is studied in a narrower context of cooperative spectrum sensing and various simple strategies are investigated. Agents are assumed to be able to communicate and share sensing results with all other agents, effectively forming a complete graph where the agents' knowledge of the network is complete and symmetric. However, such an assumption rarely holds in distributed networks where, instead, agents may interact over arbitrary topologies and have incomplete and asymmetric knowledge of the entire network. In such scenarios, simple strategies proposed in [SZ09] will fail to work and the incentives design becomes significantly more challenging.

Contagion strategies on networks [Kan92] [MA09] [JRT12] are proposed as a simple method to provide incentives for agents to cooperate. However, such methods do not perform well if monitoring is imperfect since any single error can lead to a network collapse. Even if certain forms of forgiveness are introduced, contagion strategies are shown to be effective only in very specific networks [MA09] [JRT12]. It is still extremely difficult, if not impossible, to design efficient forgiving schemes in arbitrary distributed networks since agents will have difficulty in conditioning their actions on history, e.g. whether they are in the contagion

	Social Learning [1]	**Direct Reciprocation** [7][8]	**Social Reciprocation** [10][19][20]	**This paper**
Information/goods exchange	Costless	Costly	Costly	Costly
Asymmetric interests	No	No	Yes	Yes
Objective of study	Convergence of agents' beliefs and actions	Incentives for agents to cooperate	Incentives for agents to cooperate	Incentives for agents to cooperate
Game type	One-shot game/ Bayesian game	Repeated game	Repeated game	Repeated game
Robust to monitoring errors	No	Yes & No	Yes & No	Yes
Equilibrium concept	Bayesian equilibrium	Subgame perfect equilibrium	Public perfect equilibrium	Perfect local equilibrium
Network topology	Arbitrary	Arbitrary	Fully connected/ Random matching	Arbitrary
Agents actions	Belief update	Cooperation level	Cooperation level	Cooperation level
Agents' utility depends on	Self belief/action	Own actions and others actions	Own actions and others actions	Own actions and others (joint) actions
Utility function	Homogeneous & Heterogeneous	Homogeneous	Homogenous	Heterogeneous
Distributed design	Yes	Yes	No	Yes

Table 3.1: Comparison with existing works.

phase or the forgiving phase, due to the asymmetric and incomplete knowledge.

Rating/reputation mechanisms are proposed as another promising solution to implement social reciprocation. Much of the existing work on reputation mechanism is concerned with practical implementation details such as effective information gathering techniques [KSG03] or determining the impact of reputation on a seller's prices and sales [BP02] [RZ02]. The few works providing theoretical results on rating protocol design consider either one (or a few) long-lived agent(s) interacting with many short-lived agents [Del05] [FTW05] [ZMM00] or anonymous, homogeneous and unconnected agents selected to interact with each other using random matching [Kan92] [ZPS14] [XZV12]. Importantly, few of the prior works consider the design of such rating protocols for networks where agents interact over a network, which leads to extremely complex and coupled interactions among agents. Moreover, the distributed nature of the considered sharing networks imposes unique challenges for the rating protocol design and implementation which are not addressed in prior works [ZPS14] [XZV12].

In Table 3.1, we compare the current work with existing works on social learning and incentive schemes based on direct reciprocation and social reciprocation.

3.2 System Model

3.2.1 Network Environment

We consider a network of N agents, indexed by $\{1, 2, ..., N\} = \mathcal{N}$. Agents are connected subject to an underlying topology $G = \{g_{ij}\}_{i,j\in\mathcal{N}}$ with $g_{ij} = g_{ji} = 1$ (here we consider undirected connection) representing agent i and j being connected (e.g. there is a communication channel between them) and $g_{ij} = g_{ji} = 0$ otherwise. Moreover, we set $g_{ii} = 0$. We say that agent i and agent j are neighbors if they are connected. For now we assume a static network G but dynamic networks are also allowed in our framework and this will be discussed in detail in Section VI.

Time is infinite and divided into discrete periods. In each time period, each agent i decides its action (e.g. information/goods sharing) towards each of its neighbors j, denoted by $a_{ij} \in \mathbb{R}_+$ [1]. For example, a_{ij} can represent the effort spent (e.g. information/goods shared) by agent i when interacting with agent j. We collect the actions of agent i towards all its neighbors in the notation $\boldsymbol{a}_i = \{a_{ij}\}_{j:g_{ij}=1}$. Denote $\boldsymbol{a} = (\boldsymbol{a}_1, ..., \boldsymbol{a}_N)$ as the action profile of all agents and $\boldsymbol{a}_{-i} = (\boldsymbol{a}_1, ..., \boldsymbol{a}_{i-1}, \boldsymbol{a}_{i+1}, ..., \boldsymbol{a}_N)$ as the action profile of agents except i. Let $\mathcal{A}_i = \mathbb{R}_+^{d_i}$ be the action space of agent i where $d_i = \sum_j g_{ij}$ is the number of agent i's neighbors. Let $\mathcal{A} = \times_{i\in\mathcal{N}}\mathcal{A}_i$ be the action space of all agents.

Agents obtain benefits from the information/goods shared by neighbors. We denote the actions of agent i's neighbors towards agent i by $\hat{\boldsymbol{a}}_i = \{a_{ji}\}_{j:g_{ij}=1}$ and let $b_i(\hat{\boldsymbol{a}}_i)$ be the benefit that agent i obtains from these actions. Spending effort (e.g. sharing information/goods) is costly and the cost $c_i(\boldsymbol{a}_i)$ depends on an agent i's own actions $\boldsymbol{a}_i$. Hence, given the action profile $\boldsymbol{a}$ of all agents, the utility of agent i is

$$u_i(\boldsymbol{a}) = b_i(\hat{\boldsymbol{a}}_i) - c_i(\boldsymbol{a}_i) \tag{3.1}$$

We impose some constraints on the benefit and cost functions.

Assumption 1. *For each i, the benefit $b_i(\hat{\boldsymbol{a}}_i)$ is non-decreasing in each $a_{ji}, \forall j : g_{ij} = 1$ and is strictly concave in $\hat{\boldsymbol{a}}_i$ (in other words, jointly strictly concave in $a_{ji}, \forall j : g_{ij} = 1$).*

[1] More general action space is also allowed, e.g. a_{ij} is upper bounded.

Assumption 2. *For each i, the cost is linear in its sum action, i.e. $c_i(\boldsymbol{a}_i) = \sum_{j:g_{ij}} a_{ij}$.*

The above assumptions state that (1) agents receive decreasing marginal benefits of information/goods acquisition, which captures the fact that agents become more or less "satiated" when they possess sufficient information/goods, in the sense that additional information/goods would only generate little additional payoff; (2) the cost incurred by an agent is equal (or proportional) to the sum effort spent to cooperate with all its neighbors. We note that the utility model is general enough to account for the heterogeneity of the value of information/goods to different users since $b_i(\hat{\boldsymbol{a}}_i)$ is agent-specific and depends on the action vector of all agent i's neighbors. For a concrete example, the benefit function can be the widely-adopted Dixit-Stiglitz utility function [DS77] which captures the information/goods heterogeneity and diversity produced by different agents, i.e.

$$b_i(\hat{\boldsymbol{a}}_i) = f\left(\left(\sum_{j\in\mathcal{N}_i}(w_{ji}a_{ji})^{\gamma_i}\right)^{\frac{1}{\gamma_i}}\right) \tag{3.2}$$

where $w_{ji} \geq 0$ describes the relative importance of agent j's information/goods to agent i, $\gamma_i \in (0,1)$ measures agent i' appreciation for information/goods diversity and $f(\cdot)$ is a concave and increasing function.

3.2.2 Rating Protocol

Each agent i is associated with a rating $\theta_i(t) \in \Theta = \{1, 2, ..., K\}$ in each period t which is maintained and updated according to the rating protocol. The rating of agent i is maintained by the software client of agent i. We collect agent i's neighbors' ratings in $\hat{\theta}_i = \{\theta_j\}_{j:g_{ij}=1} \in \Theta^{d_i}$. The rating protocol recommends actions to an agent depending on neighbors' ratings $\sigma_i : \Theta^{d_i} \to \mathcal{A}_i$. We refer to this recommendation as the *recommended strategy*. For agent i, σ_i=$\{\sigma_{ij}\}_{j\in\mathcal{N}_i}$ consists of d_i elements with $\sigma_{ij}(\theta_j)$ representing the recommended sharing action of agent i towards agent j if agent j's rating is θ_j. We collect the strategies of agent i's neighbors towards agent i in $\hat{\sigma}_i(\theta_i) = \{\sigma_{ji}(\theta_i)\}_{j:g_{ij}=1}$. These recommendations are done in a distributed manner by the system, through the software clients of the agents.

Depending on whether or not agent i followed the recommended strategy, its software

		*Agent (**Strategic**)*	*Software client (**Non-strategic**)*
Configuration	*Information*	---	Own agents' utility function and local connectivity
	Action/ Functionality	---	Determine the rating protocol in a distributed manner
Interaction Periods	*Information*	The instantiated rating protocol; Neighbors' ratings.	Whether or not the agent followed the recommended strategy
	Action/ Functionality	Choose sharing actions aiming to maximize own utility	Update the agent's rating; Broadcast it in the neighborhood

Table 3.2: Operation of the rating protocol.

client updates agent i's rating at the end of each period. Let $y_i \in Y = \{0, 1\}$ be the public monitoring signal of agent i with $y_i = 1$ if $\boldsymbol{a}_i = \sigma_i$ and $y_i = 0$ if $\boldsymbol{a}_i \neq \sigma_i$ which is generated by the software of agent i. However, monitoring may not be perfect and hence it is possible that even if $\boldsymbol{a}_i = \sigma_i$, it can still be $y_i = 0$ (and if $\boldsymbol{a}_i \neq \sigma_i$, $y_i = 1$). The rating update rule for agent i is a function $\tau_i : \Theta \times Y \to \Delta(\Theta)$ where $\Delta(\Theta)$ is the probability simplex of the rating set and $\tau_i(\theta_i^+; \theta_i, y_i)$ is the probability that the updated rating is θ_i^+ if agent i's current rating is θ_i and the public signal is y_i. In particular, we consider the following parameterized rating update rule, for agent i,

$$\tau_i(\theta_i^+; \theta_i, y_i p) = \begin{cases} \alpha_{i,k}, & if \;\; \theta_i^+ = \max\{1, k-1\}, y_i = 0 \\ 1 - \alpha_{i,k}, & if \;\; \theta_i^+ = k, y_i = 0 \\ \beta_{i,k}, & if \;\; \theta_i^+ = \min\{K, k+1\}, y_i = 1 \\ 1 - \beta_{i,k}, & if \;\; \theta_i^+ = k, y_i = 1 \end{cases} \tag{3.3}$$

In words, compliant agents are rewarded with a higher rating with some probability while deviating agents are punished with a lower rating with some (other) probability. These probabilities $\alpha_{i,k}, \beta_{i,k}$ are in the range of $[0, 1]$. Note that when $\alpha_{i,k} = 0$, the rating set of agent i effectively reduces to a subset $\{k, k+1, ..., K\}$ since its rating will never drop below k (if its initial rating is higher than k). Note also that agents remain at the highest rating $\theta = K$ if they always follow the recommended strategy regardless of the choice of $\beta_{i,K}$.

To sum up, the rating protocol is uniquely determined by the recommended (public) strategies $\sigma_i(\hat{\theta}_i), \forall i, \forall \hat{\theta}_i$ and the rating update probabilities $\alpha_{i,k}, \beta_{i,k}$ for every i and k. These

will be our design parameters. We denote the rating protocol by $\pi = (\Theta, \sigma, \alpha, \beta)$. The rating protocol is configured (i.e. the values of the design parameters are determined) at the beginning of the system by the software clients of the agents. The configuration is carried out in a distributed way, requiring the software clients to exchange with neighbors limited messages (i.e. Lagrangian multipliers etc.). When the network is static, the rating protocol is configured only once at the beginning. When the network is dynamic, the rating protocol is reconfigured once in a while, to adapt to the varying network. We assume that all agents are synchronized and enter the reconfiguration period simultaneously. This synchronization can be coordinated by an exogenous stochastic process (not controlled by any central planner), for instance a random sequence generator with the same seed for each agent. Alternatively, the reconfiguration can also be initiated by a particular agent and then this signal is spread over the entire network. We also note that agents will not have incentives not to perform reconfiguration since the protocol is designed in such a way that participation in this period produces a higher utility for the agent than not participating. Table 3.2 summarizes the operation of the rating protocol.

3.2.3 Problem Formulation

The objective of the protocol designer is to maximize the social welfare of the network, which is defined as the time-average sum utility of all agents, i.e.

$$V = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} \sum_{i} u_i(\boldsymbol{a}(t)) \tag{3.4}$$

If agents are obedient, then the system designer can assign socially optimal actions, denoted by $\boldsymbol{a}^{opt}(t), \forall t$, to agents and then agents will simply take the actions prescribed by the system designer. Determining the socially optimal actions involves solving the following utility maximization problem:

$$\begin{aligned} &\underset{\boldsymbol{a}}{maximize} && V \\ &subject\ to && a_{ij}(t) \geq 0, \forall i, j : g_{ij} = 1, \forall t \end{aligned} \tag{3.5}$$

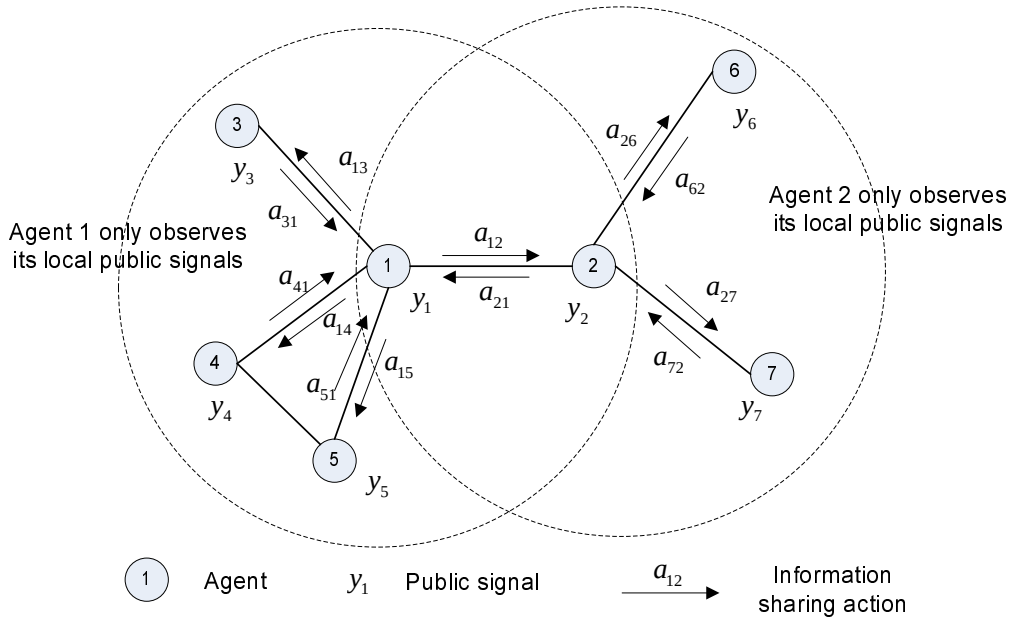


Figure 3.1: Illustration of local public signals.

This problem can be easily solved and any action profile $\boldsymbol{a}^{opt}$ that satisfies

$$\hat{\boldsymbol{a}}_i^{opt}(t) \in \arg\max_{\hat{a}} b_i(\hat{\boldsymbol{a}}_i(t)) - c_i(\hat{\boldsymbol{a}}_i(t))p \tag{3.6}$$

is a solution. We denote the optimal social welfare by V^{opt}.

The network cooperation (e.g. information/goods sharing) problem becomes much more difficult in the presence of strategic agents: strategic agents may not want to take the prescribed actions because these actions do not maximize their own utilities.

Definition 1. *A (one-shot) network sharing game is a tuple* $\mathcal{G} = \langle \mathcal{N}, \mathcal{A}, \{u_i(\cdot)\}_{i\in\mathcal{N}}; G\rangle$ *where* $\mathcal{N}$ *is the set of players,* $\mathcal{A}$ *is the action space of all players,* $u_i(\cdot)$ *is the utility function of player* i *(defined by* (3.1)*) and* G *is the underlying network.*

Consider the utility of an agent i in (3.1). In order to maximize its own utility, agent i will take the action $\boldsymbol{a}_i = 0$ regardless of other agents' actions $\boldsymbol{a}_{-i}$. Therefore, there exists a unique Nash equilibrium (NE) $\boldsymbol{a}^{NE} = 0$ in the network sharing game in any period.

In this work, we exploit the repeated interactions among agents to provide agents with incentives to cooperate. In the following, we introduce the equilibrium concept used in this work.

At the end of each interaction period, each agent i observes the (imperfect) monitoring signal $y_j \in Y = \{0, 1\}$ of the action of each of its neighbor j. Write $\mathcal{Y}_i$ for the space of signals observed by agent i and $\mathcal{Y} = \times_{i \in \mathcal{N}} \mathcal{Y}_i$ for the space of signal profiles. A profile of actions $\boldsymbol{a} \in \mathcal{A}$ determines a distribution of signals $\mu_{\boldsymbol{a}} \in \Delta(\mathcal{Y})$; agents observe a realization drawn at random from this distribution. In our network setting, the signal distribution is *local* in the sense that agent i's observed signal depends only on the actions of i' neighbors. Figure 3.1 illustrates the local signals observed by agents. A signal history of length T is an element $\boldsymbol{y} = (\boldsymbol{y}^1, ..., \boldsymbol{y}^T) \in \mathcal{Y}^T$; $\boldsymbol{y}^t$ is the signal profile at time t and $\boldsymbol{y}_i^t$ is the signal profile observed by agent i at time t. In addition to signals, agents know their own actions and their realized own utilities, so a private history of length T for agent i is an element $h \in (\mathcal{A}_i \times \mathbb{R} \times \mathcal{Y}_i)^T = \mathcal{H}_i^T$ and a private history of length T is a profile of private histories for each agent. A strategy for agent i is a function $\sigma_i : \mathcal{H}_i \to \mathcal{A}_i$, prescribing an action following each history. The strategy σ_i is a *local strategy* (or a *local signal strategy*) if it depends only on the history of local signals observed by i (and not on the history of i's actions or realized utilities.) An infinite history for agent i is an element of $(\mathcal{A}_i \times \mathbb{R} \times \mathcal{Y}_i)^\infty = \mathcal{H}_i^\infty$. Note that a strategy profile σ defines, for each agent i, a probability distribution $\zeta_i(\sigma)$ on the infinite histories $\mathcal{H}_i^\infty$ and hence a probability distribution $\nu_i(\sigma)$ on infinite utility streams $\mathbb{R}^\infty$. Agents discount future utilities, so the utility agent i derives from the infinite utility stream $\boldsymbol{u}_i = (u_i^1, u_i^2, ...)$ is

$$W_i(\boldsymbol{u}_i) = \sum_{t=0}^{\infty} \delta^t u_i^t \tag{3.7}$$

where $\delta \in (0, 1)$ is the discount factor. Hence the (expected) utility agent i derives if agents follow the strategy profile σ is

$$U_i(\sigma) = \mathbb{E}W_i(\boldsymbol{u}_i) = \int_{\boldsymbol{u}_i} W_i(\boldsymbol{u}_i) d\nu_i(\sigma)(\boldsymbol{u}_i) \tag{3.8}$$

A strategy profile σ is a Nash equilibrium if for each agent i, the strategy σ_i is a best response to other agents' strategy profile σ_{-i}; that is

$$U_i(\sigma_i, \sigma_{-i}) \geq U_i(\hat{\sigma}_i, \sigma_{-i}) \tag{3.9}$$

for every strategy $\hat{\sigma}_i$. The profile σ is a *local equilibrium* if it is a Nash equilibrium and every

agent uses a local strategy; it is a *perfect local equilibrium* (PLE) if in addition it is a Nash equilibrium following every history.

The proposed rating protocol assigns each agent a rating that summarizes the public signal history of the action of that agent. Hence, $\sigma_i : \mathcal{H}_i \to \mathcal{A}_i$ is reduced to $\sigma_i : \Theta^{d_i} \to \mathcal{A}_i$. This significantly reduces the implementation complexity since agents need to keep only the current ratings of their neighbors instead of the entire signal history of their neighbors. If a recommended strategy profile constitutes a PLE, then agents have incentives to follow their recommended strategies. Denote the achievable social welfare by adopting the rating protocol by $V(\pi)$. The rating protocol design problem thus is

$$\begin{array}{ll} \underset{\pi=(\Theta,\sigma,\alpha,\beta)}{maximize} & V(\pi) \\ subject\ to & \sigma\ constitutes\ a\ PLE \end{array} \tag{3.10}$$

3.2.4 Illustrative Example: Cooperative Estimation

We illustrate the generality of our formalism by showing how well-studied joint estimation problems [SZ09] [YSS13] can be cast into it. Our proposed framework can also be used to solve problems such as distributed cybersecurity investment [XZS13] and cooperation in economics networks [EG13] etc.

Suppose that each agent observes in each period a noisy version of a time-varying underlying system parameter $s(t)$ of interest. Denote the observation of agent i by $o_i(t)$. We assume that $o_i(t) = s(t) + \epsilon_i(t)$, where the observation error $\epsilon_i(t)$ is i.i.d. Gaussian across agents and time with mean zero and variance r^2. Agents can exchange observations with their neighbors to obtain better estimations of the system parameter. Let $a_{ij}(t)$ be the transmission power spent by agent i. The higher the transmission power the larger probability that agent j receives this additional observation from agent i. Agents can use various combination rules [SZ09] to obtain the final estimations. The expected mean square error (MSE) of agent i's final estimation will depend on the actions of its neighbors, denoted by $MSE_i(\hat{\boldsymbol{a}}_i(t))$. If we define the MSE improvement as the benefit of agents, i.e. $b_i(\hat{\boldsymbol{a}}_i(t)) = r^2 - MSE(\hat{\boldsymbol{a}}_i(t))$,

then the utility of agent i in period t given the received benefit and its incurred cost is $u_i(\boldsymbol{a}(t)) = r^2 - MSE_i(\hat{\boldsymbol{a}}_i(t)) - \|\boldsymbol{a}_i(t)\|_1$.

3.3 Distributed Optimal Rating Protocol Design

If a rating protocol constitutes a PLE, then all agents will find it in their self-interest to follow the recommended strategies. If the rating update rule updates the ratings of compliant agents upward with positive probabilities, then eventually all agents will have the highest ratings forever (assuming no update errors). Therefore, the social welfare, which is the time-average sum utility, is asymptotically the same as the sum utility of all agents when they have the highest ratings and follow the recommended strategy, i.e.

$$V = \sum_i (b_i(\hat{\sigma}_i(K)) - c_i(\sigma_i(\boldsymbol{K}))) \tag{3.11}$$

This means that the recommended strategy profile $\sigma(\boldsymbol{K})$ for the highest ratings determines the social welfare that can be achieved by the rating protocol. If this strategy profile can be arbitrarily chosen, then we can solve a similar problem as (3.5) for the obedient agent case. However, in the presence of self-interested agents, this strategy profile, together with the other components of a rating protocol, need to satisfy the equilibrium constraint such that self-interested agents have incentives to follow the recommended strategies. In Theorem 1, we identify a sufficient and necessary condition on $\sigma(\boldsymbol{K})$ such that an equilibrium rating protocol can be constructed. With this, we are able to determine the optimal rating protocol in a distributed way in order to maximize the social welfare. We denote the social welfare that can be achieved by the optimal rating protocol as V^* and use *the price of anarchy* (PoA), defined as $PoA = V^{opt}/V^*$, as the performance measure of the rating protocol.

3.3.1 Sufficient and Necessary Condition

To see whether a rating protocol can constitute a PLE, it suffices to check whether agents can improve their long-term utilities by one-shot unilateral deviation from the recommended strategy after any history (according to the one-shot deviation principle in repeated game

theory [MS06]). Since in the rating protocol, the history is summarized by the ratings, this reduces to checking the long-term utility in any state (i.e. any rating profile θ of agents). Agent i's long-term utility when agents choose the action profile $\boldsymbol{a}$ is

$$U_i(\theta; \boldsymbol{a}) = u_i(\theta; \boldsymbol{a}) + \delta \sum_{\theta'} p(\theta'|\theta; \boldsymbol{a}) U_i^*(\theta'), \tag{3.12}$$

where $p(\theta'|\theta; \boldsymbol{a})$ is the rating profile transition probability which can be fully determined by the rating rating update rule based on agents' actions and U_i^* which is the optimal value of agent i at the rating profile θ', i.e. $U_i^*(\theta') = \max_{\boldsymbol{a}} U_i(\theta; \boldsymbol{a})$. PLE requires that the recommended actions for any rating profile are the optimal actions that maximize agents' long-term utilities. Before we proceed to the proof of Theorem 1, we prove the following Lemma, whose proof is deferred to the appendix.

Lemma 1. *1. $\forall \theta$, the optimal action of agent i is either $\boldsymbol{a}_i^*(\theta) = \mathbf{0}$ or $\boldsymbol{a}_i^*(\theta) = \sigma_i$.*

2. $\forall \theta_i$, if for $\hat{\theta}_i = \boldsymbol{K}$, $\boldsymbol{a}_i^*(\theta) = \sigma_i(\hat{\theta}_i)$, then for any other $\hat{\theta}_i$, $\boldsymbol{a}_i^*(\theta) = \sigma_i(\hat{\theta}_i)$.

3. Let $\hat{\theta}_i = \boldsymbol{K}$, suppose $\forall \theta \theta_i$, $\boldsymbol{a}_i^*(\theta) = \sigma_i(\hat{\theta}_i)$, then $\theta_i < \theta \Leftrightarrow U_i^*(\theta_i, \hat{\theta}_i) \leq U_i^*(\theta_i', \hat{\theta}_i)$

Lemma 1.1 characterizes the set of possible optimal actions. That is, self-interested agents choose to either share nothing or the recommended amount of information/goods with their neighbors. Lemma 1.2 states that if an agent has an incentive to follow the recommended strategy when all its neighbors have the highest ratings, then it will also have an incentive to follow the recommended strategy in all other cases. Lemma 1.3 shows that the optimal long-term utility of an agent is monotonic in its rating when all its neighbors have the highest ratings – the higher the rating the larger the long-term utility the agent obtains. With these results in hand, we are ready to present and prove Theorem 1.

Theorem 1. *Given the rating protocol structure and the network structure, at least one rating protocol can be constructed to be a PLE if and only if $\delta b_i(\hat{\sigma}_i(K)) \geq c_i(_i(\boldsymbol{K})), \forall i$.*

Proof. (*Sketch*): For the sake of conciseness, we provide only the proof sketch of Theorem 1. The complete proof can be found in appendix. According to Lemma 1.2, it suffices to ensure that agent i has an incentive to take the recommended strategy when it neighbors'

ratings are $\hat{\theta}_i = \boldsymbol{K}$. However, we need to prove that this holds for all ratings of agent i. To prove the "only if" "only if" part, we show that if $\delta b_i(\hat{\sigma}_i) < c_i(\sigma_i), \forall i$, then no rating constitute a PLE by showing a contradiction. To prove the "if" part, we construct a binary rating protocol that can constitute a PLE when a PLE when $\delta b_i(\hat{\sigma}_i) \geq c_i(\sigma_i)$ is satisfied. In particular, we choose $\alpha_{i,2} = \beta_{i,1} = 1, \forall i$ as the rating update probabilities in such a rating protocol. □

3.3.2 Computing the Recommended Strategy

Theorem 1 provides a sufficient and necessary condition for the existence of a PLE with respect to the recommended strategies when agents have the highest ratings. From (3.11) we already know that these strategies fully determine the social welfare that can be achieved by the rating protocol. Therefore, the optimal values of $\sigma(\boldsymbol{K})$ can be determined by solving the following *optimal recommended strategy design* problem:

$$\begin{aligned} \underset{\sigma}{\textit{maximize}} \quad & \textstyle\sum_i (b_i(\hat{\sigma}_i(\sigma_i(\boldsymbol{K})))) \\ \textit{subject to} \quad & c_i(\sigma_i(\boldsymbol{K})) \leq \delta b_i(\hat{\sigma}_i(K)), \forall i \\ & \sigma \geq 0 \end{aligned} \tag{3.13}$$

where the constraint ensures that an equilibrium rating protocol can be constructed. Note that this problem implicitly depends on the network since both $\hat{\sigma}_i(K)$ and $\sigma_i(\boldsymbol{K}), \forall i$ are network-dependent (since for each agent i, the strategy is only towards its neighbors). In this subsection, we subsection, we will write $\sigma_i(\boldsymbol{K})$ as σ_i and $\hat{\sigma}_i(K)$ as $\hat{\sigma}_i$ to keep the notation simple.

Firstly, we show the strong duality holds for the problem (3.13) under mild conditions.

Proposition 1. *Strong duality holds for* (3.13) *if the following condition on the benefit function holds:* $\forall i \in \mathcal{N}$

$$\max_j \left. \frac{\partial b_i(\hat{\boldsymbol{x}}_i)}{\partial x_{ji}} \right|_{\hat{\boldsymbol{x}}_i = \boldsymbol{0}} > \frac{d_i}{\delta} \tag{3.14}$$

Proof. It is easy to see that the problem in (3.13) is a convex optimization problem. According to the Slater's condition [BV04], strong duality holds if there exists a strictly feasible

solution σ such that

$$\begin{aligned} & c_i(\sigma_i(K)) < \delta b_i(\hat{\sigma}_i), \forall i \\ & \sigma \geq \mathbf{0} \end{aligned} \tag{3.15}$$

i.e. a solution such that the non-linear constraints are strictly satisfied.

Consider agent i, we find its neighbor j^* such that $j^* = \arg\max_j \frac{\partial b_i(\hat{\boldsymbol{x}}_i)}{\partial x_{ji}}\Big|_{\hat{\boldsymbol{x}}_i=\mathbf{0}}$. We construct a strategy $\hat{\sigma}_i$ such that $\sigma_{j^*i} = \epsilon$ and $\sigma_{ji} = 0, \forall j \neq j^*$. Because $\frac{\partial b_i(\hat{\boldsymbol{x}}_i)}{\partial x_{j^*i}}\Big|_{\hat{\boldsymbol{x}}_i=\mathbf{0}} > \delta$ and the $b_i(\hat{\boldsymbol{x}}_i)$ is concave, we can we can always find an $\bar{\epsilon} > \mathbf{0}$ such that $\left|\frac{\partial b_i(\hat{\boldsymbol{x}}_i)}{\partial x_{j^*i}}\right|_{\hat{\boldsymbol{x}}_i=\hat{\sigma}_i} = \frac{d_i}{\delta}$. Hence, for any $\epsilon \in (0, \bar{\epsilon})$, $b_i(\hat{\sigma}_i) > \frac{d_i}{\delta} \geq \frac{c(\sigma_i)}{\delta}$. The last inequality is because the cost of agent i is at most $c_i(\sigma_i) \leq d_i$. If we do this for all agents, then we find a strategy profile σ that is a strictly feasible solution. □

The condition in the above proposition requires that each agent can obtain a sufficiently large marginal benefit at $\mathbf{0}$ from at least one of its neighbors. This is a mild condition and holds for numerous benefit functions such as the Dixit-Stiglitz utility function in (3.2). Moreover, (3.15) is rather conservative: in many problems, the right-hand side of (3.15) can be much smaller.

Now, we propose a distributed algorithm to compute these recommended strategies using the dual decomposition method [PC07] [BV04]. The idea is that we decompose the Optimal Recommended Strategy Design problem (3.13) into N sub-problems each of which is locally solved for each agent. Note that unlike the case with obedient agents, these sub-problems have coupled constraints. Therefore, the software of agents will need to go through an iterative process to exchange messages (i.e. the Lagrangian multipliers) with their neighbors such that their local solutions converge to the global optimal solution. We describe the algorithm in detail below.

We perform dual decomposition on (3.13) and form the partial Lagrangian,

$$\begin{aligned} & L(\sigma, \lambda) = \textstyle\sum_i (b_i(\hat{\sigma}_i) - c_i(\sigma_i)) + \sum_i \lambda_i (c_i(\sigma_i) - \delta b_i(\hat{\sigma}_i)) \\ & = \textstyle\sum_i \left[(1 + \lambda_i \delta) b_i(\hat{\sigma}_i) - \sum_{j:g_{ij}=1} (1 + \lambda_j) \sigma_{ji} \right] \\ & \triangleq \textstyle\sum_i L_i(\hat{\sigma}_i, \lambda) \end{aligned} \tag{3.16}$$

where $\lambda_i \geq 0$ is the Lagrange multiplier associated with the incentive constraint of agent i. The second equality is due to the linearity of the cost function. The master dual problem is,

$$\begin{array}{ll} \underset{\lambda}{minimize} & g(\lambda) = \sum_i g_i(\lambda) \\ subject\ to & \lambda_i \geq 0, \forall i \end{array} \tag{3.17}$$

where $g(\lambda) = \max_{\lambda} L(\sigma, \lambda)$. When strong duality holds, the optimal value $g^*(\lambda)$ equals the optimal value of value of the original primal problem (3.13). Next, we solve $g^*(\lambda)$ using the subgradient method. A subgradient of $-g$ is as follows: for λ_i, the subgradient is $c_i(\sigma_i^*(\lambda)) - \delta b_i(\hat{\sigma}_i^*(\lambda))$. Therefore, we need to solve the optimal $\sigma^*(\lambda)$ for a given λ to get the subgradient. Notice that the Lagragian $L(\sigma, \lambda)$ can be separated into N separated into N sub-Lagrangians $L_i(\hat{\sigma}_i, \lambda_i)$, we can obtain $\hat{\sigma}_i^*, \forall i$ by solving each subproblem individually,

$$\underset{\hat{\sigma}_i}{maximize} \qquad [1 + \lambda_i \delta] b_i(\hat{\sigma}_i) - \sum_{j: g_{ij}=1} (1 + \lambda_j) \sigma_{ji} \tag{3.18}$$

The above problem is a convex optimization problem and hence is easy to solve. Now we have found the subgradient, that master algorithm updates the dual varible λ based on this subgradient,

$$\lambda_i(q+1) = \left[\lambda_i(q) + w(c_i(\sigma_i^*(\lambda(q))) - \delta b_i(\hat{\sigma}_i^*(\lambda(q))))\right]^+, \forall i \tag{3.19}$$

where q is the iteration index, $w > 0$ is a sufficiently small positive step-size. Because (3.13) is a convex optimization, it is well known [BV04] that such an iterative algorithm will converge to the dual optimal λ^* as $q \to \infty$ and the primal variable $\sigma^*(\lambda(q))$ will also converge to the primal optimal σ^*.

This iterative process can be made fully distributed which requires only limited message exchange between the software clients of neighboring agents. We present the Distributed Computation of the Recommended Strategy (DCRS) Algorithm below which is run locally by the software client of each agent.

The above DCRS algorithm has the following interpretation. In each configuration slot, the software client of each agent computes the sharing actions of the agent's neighbors that maximize the social surplus with respect to its own agent (i.e. the benefit obtained by its own

Table 3.3: Algorithm:Distributed Computation of the Recommended Strategy (DCRS)

(Run by the software client of agent i)

Input: Connectivity and utility function of agent i.

Output: $\sigma_i(\boldsymbol{K}) = \{\sigma_{ij}(K)\}_{j:g_{ij}=1}$ (denoted by $\sigma_i = \{\sigma_{ij}\}_{j:g_{ij}=1}$ for simplification)

Initialization: $q = 0$; $\lambda_i(q) = 0$

Repeat:

Send $\lambda_i(q)$ to neighbor j, $\forall j : g_{ij} = 1$.

(Obtain $\lambda_j(q)$ from j, $\forall j$)

Solve (3.18) using $\lambda_i(q)$, $\{\lambda_j(q)\}_{j:g_{ij}=1}$ to obtain $\hat{\sigma}_i(\lambda(q))$.

Send $\sigma_{ji}(\lambda(q))$ to neighbor j, $\forall j : g_{ij} = 1$.

(Obtain $\sigma_{ij}(\lambda(q))$ from j, $\forall j$)

Update $\lambda_i(q+1)$ according to (3.19).

Stop until $\|\lambda_{ji}(q+1) - \lambda_{ji}(q)\|_2 < \varepsilon_\lambda$

agent minus the cost incurred by its neighbors). However, this computation has to take into account whether neighboring agents' incentive constraints are satisfied, which are reflected by the Lagrangian multipliers. The larger λ_i is, the more likely it is that agent i's incentive constraint is violated. Hence, the neighbors of agent i should acquire less information/goods from it. We note that the DCRS algorithm needs to be run to compute the optimal strategy only once at the beginning if the network is static.

3.3.3 Computing the Remaining Components of the Rating Protocol

Even though the DCRS algorithm provides a distributed way to compute the recommended strategy when agents have the highest ratings, the other elements of the rating protocol remain to be determined. There are many possible rating protocols that can constitute a PLE given the obtained recommended strategies. In fact, we already provided one way to compute these remaining elements when we determined the sufficient condition in Theorem

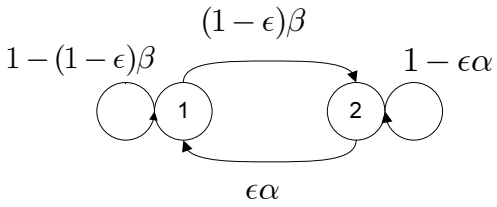


Figure 3.2: Markov chain of the rating transition.

1 by using a constructive method. However, this is not the most efficient design in the imperfect monitoring scenario where ratings will occasionally drop due to monitoring errors. Therefore, the remaining components of the rating protocol should still be smartly chosen in the presence of monitoring errors. In this subsection, we consider a rating protocol with a binary rating set $\Theta = \{1, 2\}$ and $\sigma_{ij}(\theta = 1) = 0, \forall i, j : g_{ij} = 1$. We design the rating update probabilities $\alpha_{i,2}, \beta_{i,1}, \forall i$ to maximize the social welfare when monitoring error exists.

Proposition 2. *Given a binary rating protocol* $\Theta = \{1, 2\}$, $\sigma_{ij}(2), \forall i, j : g_{ij} = 1$ *determined by the DCRS algorithmp and* $\sigma_{ij}(1) = 0, \forall i, j : g_{ij} = 1$, *when the monitoring error* > 0, *the optimal rating update probability that maximize the social welfare is,* $\forall i$, $\beta^*_{i,1} = 1, \alpha^*_{i,2} = \frac{c_i(\sigma_i(\mathbf{2}))}{\delta b_i(\hat{\sigma}_i(2))}$

Proof. The social welfare is the time-average sum utility of all agents. Therefore, we need to maximize the expected utility for each individual agent. Since we consider a binary rating protocol, let η^1_i, η^2_i be the probability that agent i has rating 1 and rating 2, respectively. Note that $\eta^1_i + \eta^2_i = 1$. The expected time-average utility of agent i can be written as $\mathbb{E}V_i = \eta^1_i u_i(1) + \eta^2_i u_i(2)$. Since the utility of having a higher rating is larger than that of having a lower rating, $u_i(2) \geq u_i(1)$. In order to maximize $\mathbb{E}V_i$, we need to maximize π^2_i. Given $\alpha_{i,2}, \beta_{i,1}$, we can determine η^2_i by solving the stationary distribution of a two-state Markov chain. In this Markov chain, the states are the ratings and the transition probabilities are depicted in Figure 3.2. A simple calculation of this Markov chain yields the solution $\eta^2_i = \frac{(1-)\beta_{i,1}}{\alpha_{i,2}+(1-)\beta_{i,1}}$.

Now, in order to maximize η^2_i, it is equivalent to maximize $\beta_{i,1}/\alpha_{i,2}$. However, $\alpha_{i,2}$ and $\beta_{i,1}$ are subject to the incentive constraints and we can derive the feasible values of $\alpha_{i,2}, \beta_{i,1}, \forall i$

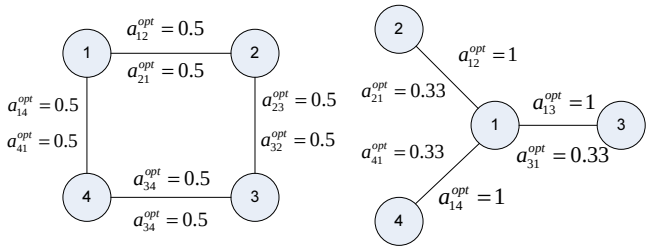


Figure 3.3: Optimal strategies for obedient agents.

as follows,

$$\begin{aligned} \beta_{i,1} &\geq \tfrac{1-\delta}{\delta} \tfrac{c_i(\sigma\sigma_i(\mathbf{2}))}{b_i(\hat{\sigma}_i(2))-c_i(\sigma_i(\mathbf{2}))}, \\ \alpha_{i,2} &\geq \tfrac{1-\delta(1-\beta_{i,1})}{\delta} \tfrac{c_i(\sigma_i(\mathbf{2}))}{b_i(\hat{\sigma}_i(2))} \end{aligned} \tag{3.20}$$

For any $\beta_{i,1}$, the optimal value of $\alpha_{i,2}$ is the binding value of second inequality in (3.20) and hence, we need minimize $[1-\delta(1-\beta_{i,1})]/\beta_{i,1}$. Because $[1-\delta(1-\beta_{i,1})]/\beta_{i,1}$ is decreasing in $\beta_{i,1}$, the optimal value of $\beta_{i,1}$ is $\beta_{i,1}^* = 1$. Using (3.20) again, the optimal value of $\alpha_{i,2}^* = \frac{c_i(\sigma_i(\mathbf{2}))}{\delta b_i(\hat{\sigma}_i(2))}$. □

It is worth noting that these probabilities can be computed locally by the software of the agents which do not require any information from other agents.

3.3.4 Illustrative Rating Protocols

In this section we show how the rating protocol can be determined in a distributed manner given the network structure. Specifically, we consider a set of 4 agents performing cooperative estimation (as in Section III. D) over two fixed networks – a ring and a star. A possible approximation of the utility function of each agent i when the uniform combination rule is used is $u_i(\boldsymbol{a}(t)) = [r^2 - \frac{r^2}{1+\sum_{j:g_{ij}} a_{ji}}] - \sum_{j:g_{ij}} a_{ij}$. We assume that the noise variance $r^2 = 4$. Figure 3.3 illustrates the optimal actions in different networks by solving (3.5). In both networks, the optimal social welfare is $V^{opt} = 4$. Figure 3.4 illustrates the optimal recommended strategies computed using the method developed in this section for these two topologies (assuming $\epsilon_i \to 0, \forall i$).

In the ring network, agents are homogeneous and links are symmetric. As we can see, the optimal recommended strategy σ^* is exactly the same as the socially optimal action profile

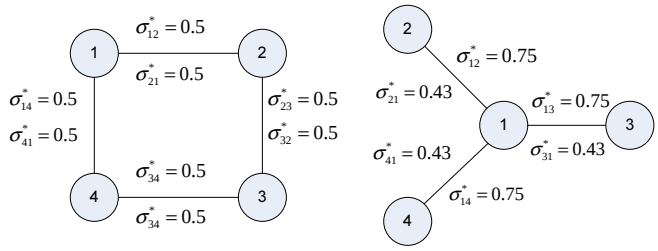


Figure 3.4: Optimal strategies for strategic agents.

$\boldsymbol{a}^{opt}$ for obedient agent case because $\boldsymbol{a}^{opt}$ already provides sufficient incentive for strategic agents to follow. Therefore, we can easily determine that $PoA = 1$. However, the strategic behavior of agents indeed degrades the social welfare in other cases, especially when the network becomes more heterogeneous and asymmetric, e.g. the star network. Even though taking $\boldsymbol{a}^{opt}$ maximizes the social welfare $V^{opt} = 4$ in the star network, these actions are not incentive-compatible for all agents. In particular, the maximum welfare $V^{opt} = 4$ is achieved by sacrificing the individual utility of the center agent (i.e. agent 1 needs to contribute much more than it obtains). However, when agents are strategic, the center agent will not follow these actions $\boldsymbol{a}^{opt}$ and hence, $V^{opt} = 4$ cannot be achieved. More problematically, since the center agent will choose not to participate in the sharing process, the periphery agents do not obtain benefits and hence, they will also choose not to participate in the sharing process. This leads to a network collapse. In the proposed rating protocol, the recommended strategies satisfy all agents' incentive constraints, namely $\delta b_i(\hat{\sigma}_i(K)) \geq c_i(\sigma_i(\boldsymbol{K})), \forall i$. By comparing comparing $\boldsymbol{a}^{opt}$ and σ^*, we can see that the rating protocol recommends more sharing from the periphery agents to the center agent and less sharing from the center agent to the agents than the obedient agent case. In this way, the center agent will obtain sufficient benefits from participating in the sharing. However, due to this compensation for the center agent, the PoA is increased to $PoA = 1.036$.

3.4 Performance Analysis

In this section, we analyze the performance of the rating protocol and try to answer two important questions: (1) What is the performance loss induced by the strategic behavior

of agents? (2) What is the performance improvement compared to other (simple) incentive mechanisms?

3.4.1 Price of Anarchy

Consider the social welfare maximization problems (3.5) and (3.13) for obedient agents and strategic agents (by using rating protocols), respectively. It is clear that the social welfare achieved by the rating system is always no larger than that obtained when agents are obedient due to the equilibrium constraint; hence, i.e. $PoA \geq 1$. The exact value of PoA will, in general, depend on the specific network structure (topology and individual utility functions). In this subsection, we identify a sufficient condition for the connectivity degree of the network such that PoA is one. To simplify the analysis, we assume that agents' benefit functions are homogeneous and depend only on the sum sharing action of the neighboring agents, i.e. $b_i(\hat{\boldsymbol{a}}_i) = b(\sum_{j:g_{ij}=1} a_{ji})$. Recall that $d_i = \sum_j g_{ij}$ is the number of neighbors of agent i. The degree of network G is defined as $d = \max_i d_i$.

Proposition 3. *If the benefit function satisfies $b_i(\hat{\boldsymbol{a}}_i) = b(\sum_{j:g_{ij}=1} a_{ji}), \forall i$ and the sharing action is upper-bounded $a_{ij} \leq 1, \forall i, j$, then there exists a $\bar{d}$ such that if $d \leq \bar{d}$, $PoA = 1$.*

Proof. Due to the concavity of the benefit function (Assumption 1), there exists d^* such that if $d > d^*$, $b(d) - d$ is increasing and if $d \leq d^*$, $b(d) - d$ is decreasing. If the connectivity degree satisfies $d < d^*$, then the optimal solution of (3.5) is $a_{ij} = 1, \forall i, j : g_{ij} = 1$. That is, optimality is achieved when all agents share the maximal amount of information/goods with all their neighbors. Therefore, $\forall d < d^*$, the agent i's benefit is $b(d_i)$ and its cost is d_i in the optimal solution. Moreover due to the concavity of the benefit function, there exists d^{**} such that if $d > d^{**}$, $\delta b(d) - d < 0$ and if $d \leq d^{**}$, $\delta b(d) - d \geq 0$. Therefore, if $d \leq d^{**}$, then agents' incentives are satisfied. Therefore if we let $\bar{d} = \min\{d^*, d^{**}\}$, then $\forall d < \bar{d}$, all agents have incentives to share the maximal amount of information/goods with their neighbors in which case the social optimum is also obtained. Hence, $PoA = 1$. □

Proposition 3 states that when the connectivity degree is low, the proposed rating pro-

tocol can achieve the optimal performance even when agents are strategic.

3.4.2 Comparison with Direct Reciprocation

The proposed rating protocol is not the only incentive mechanism that can incentivize agents to share information/goods with other agents. A well-known direct reciprocation based incentive mechanism is the Tit-for-Tat strategy, which is widely adopted in many networking applications [Axe81] [MJS06]. The main feature of the Tit-for-Tat strategy is that it exploits the repeated *bilateral* interactions between connected agents, which can be utilized to incentivize agents to *directly* reciprocate to each other. However, when agents have asymmetric interests, such mechanisms fail to provide such incentives and direct reciprocity algorithms cannot be applied.

Moreover, even if we assume that interests are symmetric between agents, our proposed rating protocol is still guaranteed to outperform the Tit-for-Tat strategy when the utility function takes a concave form as assumed in this work. Intuitively, because the marginal benefit from acquiring information/goods from one neighbor is decreasing in the total number of neighbors, agents become less incentivized to cooperate when their deviation towards some neighboring agent would not affect future information/goods acquisition from others, as is the case with the Tit-for-Tat strategy. In the following, we formally compare our proposed rating protocol with the Tit-for-Tat strategy. We assume that an agent i has two possible actions towards its neighboring agent j from : either no cooperation at all, or a fixed sharing action, i.e. $\{0, \bar{a}_{ij}\}$ where $\bar{a}_{ij} \in \mathbb{R}_+$. The Tit-for-Tat strategy prescribes the action for each agent i as follows, $\forall j : g_{ij} = 1$,

$$\begin{aligned} & a_{ij}(0) = \bar{a}_{ij} \\ & a_{ij}(t+1) = \begin{cases} \bar{a}_{ij}, & if \;\; a_{ji}(t) = \bar{a}_{ji} \\ 0, & if \;\; a_{ji}(t) = 0 \end{cases}, \forall t \geq 0 \end{aligned} \tag{3.21}$$

Proposition 4. *Given the network structure and the discount factor, any action profile $\bar{\boldsymbol{a}}$ that can be sustained by the Tit-for-Tat strategy can also be sustained by the rating protocol.*

Proof. Consider the interactions between any pair of agents i, j. In the Tit-for-Tat strategy,

the long-term utility of agent i by following the strategy when agent j played $\bar{a}_{ji}$ in the previous period is $U_i = \frac{\tilde{b}_i(\bar{a}_{ji})-\bar{a}_{ij}}{1-\delta}$ where $\tilde{b}_i(x) = b_i(\hat{a}_i|a_{ki} = \bar{a}_{ki}, a_{ji} = x)$. If agent i deviates in the current period, Tit-for-Tat induces a continuation history $(\{\bar{a}_{ij}, 0\}, \{0, \bar{a}_{ji}\}, \{\bar{a}_{ij}, 0\}...)$ where the first components are agent i's actions and the second components is agent j's actions. The long-term utility of agent i by one-shot deviation is thus

$$\begin{aligned} U_i' &= \tilde{b}_i(\bar{a}_{ji}) + \delta[\tfrac{\tilde{b}_i(0)-\bar{a}_{ij}}{1-\delta^2} + \delta\tfrac{\tilde{b}_i(\bar{a}_{ji})}{1-\delta^2}] \\ &= \tfrac{\tilde{b}_i(\bar{a}_{ji})}{1-\delta^2} + \delta\tfrac{\tilde{b}_i(0)-\bar{a}_{ij}}{1-\delta^2} \end{aligned} \tag{3.22}$$

Incentive-compatibility requires that $U_i \geq U_i'$ and therefore

$$\delta(\tilde{b}_i(\bar{a}_{ji}) - \tilde{b}_i(0)) \geq \bar{a}_{ij} \tag{3.23}$$

Due to the concavity of the benefit function, it is easy to see that (3.23) leads to $\delta b_i(\hat{\boldsymbol{a}}_i) \geq c_i(\boldsymbol{a}_i)$ which is a sufficient condition for the rating protocol to be an equilibprium. □

Proposition 4 proves that the social welfare achievable by the rating protocol equals or exceeds that of the Tit-for-Tat strategy, which confirms the intuitive argument before that diminishing marginal benefit from information/goods acquisition would result in less incentives to cooperate in an environment with only direct reciprocation than in one allowing indirect reciprocation. We note that different action profiles $\bar{\boldsymbol{a}}$ will generate different social welfare. However, computing the best $\bar{\boldsymbol{a}}$ among the incentive-compatible Tit-for-Tat strategies is often intractable since (3.23) is a non-convex constraint. Hence, implementing the best Tit-for-Tat strategy to maximize the social welfare is often intractable. In contrast, the proposed rating protocol does not have this problem since the equilibrium constraint established in Theorem 1 is convex and hence, the optimal recommended strategy can be solved in a distributed manner by the proposed DCRS algorithm.

3.5 Dynamic Networks

In Section IV, we designed the optimal rating protocol by assuming that the network is static. In practice, the social network can also change over time due to, e.g., new agents

entering the network and new links being created. Nevertheless, our framework can easily handle such growing networks by adopting a simple extension which refreshes the rating protocol (i.e. re-computes the recommended strategy, rating update rules and re-initializes the ratings of agents) with a certain probability each period. We call this probability the refreshing rate and denote it by $\rho \in [0, 1]$. When networks are dynamic, the refreshing rate will also be an important design parameter of the rating protocol.

3.5.1 Refreshing Rate Design Problem

Denote the network in period t by $G(t)$. We assume that in each period an expected number $n(t)$ of new agents enter the network and stay forever. Therefore, the network $G(t+1)$ will be formed based on $G(t)$ and the new agents. Note that before the next protocol refreshing, these new agents do not create benefits to or obtain benefits from their neighbors due to the incentive problem. Let $V^{opt}(G(T); \rho)$ be the optimal social welfare and $V^*(G(T); \rho)$ be the social welfare achieved by the rating protocol starting from a network G for a refreshing rate ρ. Our objective is to minimize the PoA by choosing a proper ρ. The optimal social welfare $V^{opt}(G(T); \rho)$ can be computed as follows,

$$V^{opt}(G(T); \rho) = \mathbb{E} \sum_{t=0}^{\infty} \rho(1-\rho)^t \frac{1}{t+1} \sum_{\tau=0}^{t} V^{opt}(G(T+\tau)) \tag{3.24}$$

Due to the refreshing, agents' discount factor effectively becomes $(1-\rho)\delta$. Therefore, the social welfare achieved by the rating protocol $V^*(G(T); \rho)$ can be obtained by solving the following optimization problem

$$\begin{aligned} \underset{\sigma}{maximize} \quad & \textstyle\sum_i (b_i(\hat{\sigma}_i(K)) - c_i(\sigma_i(K))) \\ subject\ to \quad & c_i(\sigma_i(K)) \leq (1-\rho)\delta b_i(\hat{\sigma}_i(K)), \forall i \\ & \sigma \geq 0 \end{aligned} \tag{3.25}$$

Formally, the refreshing rate design problem is formulated as the following optimization

problem,

$$\begin{aligned} &\underset{\rho}{minimize} \quad PoA(\rho) \triangleq \tfrac{V^{opt}(G(T);\rho)}{V^{*}(G(T);\rho)} \\ &subject\ to \ \ V^{opt}(G(T);\rho) \ \ is\ computed\ by\ (3.24) \\ &\qquad\qquad\quad V^{*}(G(T);\rho) \ \ is\ solved\ by\ (3.25) \end{aligned} \tag{3.26}$$

3.5.2 Impact of the Refreshing Rate

In this subsection, we study the impact of ρ on $V^*(G(T);\rho)$ and $V^{opt}(G(T);\rho)$ separately and then provide guidelines on choosing the optimal ρ^* that minimizes $PoA(\rho)$.

Proposition 5. *Both $V^*(G(T);\rho)$ and $V^{opt}(G(T);\rho)$ are non-increasing in ρ.*

Proof. Since $V^*(G(T);\rho)$ is the optimal solution of (3.25), relaxing the constraints by decreasing ρ weakly increases $V^*(G(T);\rho)$. Therefore $V^*(G(T);\rho)$ is non-increasing in ρ.

It is easy to show that $V^{opt}(G(T+\tau))$ is non-decreasing in τ because we can let the new agents share nothing and the existing agents keep their previous strategies. Then according to (3.24) it is easy to see that $V^{opt}(G(T);\rho)$ is non-increasing in ρ. □

Proposition 5 shows the monotonicity of $V^*(G(T);\rho)$ and $V^{opt}(G(T);\rho)$ with respect to ρ. If ρ is smaller, then there are more new entering agents and hence, the time-average optimal social welfare is larger. Moreover, since a smaller ρ means a more static rating protocol, the existing agents have more incentives to follow it.

Proposition 6. *(1) $\lim_{\rho\to 1} PoA(\rho) \to \infty$. (2) If $\forall t_2 > t_1$, $V^{opt}(G(t_2)) - V^{opt}(G(t_1)) > \kappa > 0$, then $\lim_{\rho\to 0} PoA(\rho) \to \infty$. (3) If $\lim_{t\to\infty} V^{opt}(G(t)) - V^{opt}(G(T)) < \kappa$, then $\lim_{\kappa\to 0} \rho^* \to 0$.*

Proof. (1) Because $V^{opt}(G(T);1) = V^{opt}(G(T)) > 0$ and $V^*(G(T);1) = 0$, $\lim_{\rho\to 0} PoA(\rho) \to \infty$. (2) Since in each time the increase of the optimal social welfare is at least a constant positive value, $\lim_{\rho\to 0} V^{opt}(G(T);\rho) \to \infty$. Because $V^*(G(T);0) = V^*(G(T)) > 0$, $\lim_{\rho\to 0} PoA(\rho) \to \infty$. (3) $\kappa \to 0$ implies that $\lim_{\rho\to 0} V^{opt}(G(T);\rho) \to V^{opt}(G(T))$. Since $V^*(G(T);\rho)$ is non-increasing in ρ, $PoA(\rho)$ is non-decreasing in ρ. Therefore $\lim_{\kappa\to 0} \rho^* \to 0$. □

The first two parts of Proposition 6 reveals the impact of the refreshing rate on the PoA in two different ways. On one hand, a larger refreshing rate provides less incentives for agents to follow the current rating protocol designed in time T. One the other hand, a smaller refreshing rate leads to a worse adaptation of the rating protocol to the changing network. Therefore, the optimal refreshing probability ρ^* should be neither too larger nor too small. The third part states that if the speed of the optimal social welfare increase tends to 0 sufficiently quickly (e.g. the arrival rate of new agent is sufficiently smaller), then the optimal refreshing rate tends to be 0, i.e. the protocol is almost never refreshed. This is intuitive since if the network changes extremely slowly, then we almost do not need to refresh the rating protocol.

3.5.3 Exiting agents

The proposed rating protocol with refreshing can also be applied to the general dynamic networks with both entering and exiting agents. However, when agents are exiting, unlike (3.25), the social welfare $V^*(G(T);\rho)$ that can be achieved by the rating protocol is difficult to characterize analytically. In particular, agents' incentives can be affected in different ways for different networks and $V^*(G(T);\rho)$ could be 0 in the worst case. Below we provide two examples that illustrate the different impacts.

1. Consider a star network with N periphery agents where at time T each periphery agent shares one unit of information/goods with the center agent and vice versa. The center agent's incentive constraint satisfies $c(N) \leq (1-\rho)\delta b(N)$. Suppose one periphery agent exits the network before the next refreshing update of the rating protocol. The center agent then receives one less unit of information/goods and needs to send one less unit of information/goods. If N is large, the incentive constraint of the center agent is still satisfied $c(N-1) \leq (1-\rho)\delta b(N-1)$ since the benefit function is concave. Because the center agent still has an incentive to follow the recommended strategy with respect to the remaining periphery agents, the remaining periphery agents' incentives to follow the recommended strategy are not affected. Therefore, the rating protocol

works efficiently before the next refreshing update.

2. Consider a ring network where at time T each agent has the incentives to follow the recommended strategy which recommends sharing one unit of information/goods to its right-hand side neighbor. Each agent's incentive constraint satisfies $c(1) \leq (1-\rho)\delta b(1)p$. Suppose a single agent exits the network before the next refreshing update of the rating protocol. In this case, the incentive of its right-hand side neighbor to follow the recommended strategy is violated since all its benefit disappears. More problematically, this will cause a "chain effect" which leads tpo all remaining agents not sharing any information/goods with others. In such scenarios, the rating protocol fails to provide agents with sharing incentives.

From the above two examples, we see that it is significantly more difficult to understand the incentives of agents for the case with agents exiting since the game played by the agents may change in unpredictable ways. In this case, we may require other game theoretical concepts and tools to tackle this problem. One possible solution is making conjectures and using the notion of conjectural equilibrium [SV09] or using social learning [KP13]. We leave this as an interesting future research topic.

3.6 Illustrative Results

In this section, we provide simulation results to illustrate the performance of the rating protocol. In all simulations, we consider the cooperative estimation problem introduced in Section III (A). Therefore, agents' utility function takes the form of $u_i(\boldsymbol{a}(t)) = [r^2 - MSE_i(\hat{\boldsymbol{a}}_i(t))] - \boldsymbol{a}_i(t)$ [YSS13]. We will investigate different aspects of the rating protocol by varying the underlying topologies and the environment parameters.

3.6.1 Impact of Network Connectivity

Now we investigate in more detail how the agents' connectivity shapes their incentives and influences the resulting social welfare. In the first experiment, we consider the cooperative es-

timation over star topologies with different sizes (hence, different connectivity degrees). Figure 3.5 shows the PoA achieved by the rating protocol for discount factors $\delta = 1, 0.9, 0.8, 0.7$ for the noise variance $r^2 = 8$. As predicted by Proposition 3, when the connectivity degree is small enough, the PoA equals one and hence, the performance gap is zero. As the network size increases (hence the connectivity degree increases in the star network), the socially optimal action requires the center agent to share more with the periphery agents. However, it becomes more difficult for the center agent to have incentives to do so since the sharing cost becomes much larger than the benefit. In order to provide sufficient incentives for the center agent to participate in the sharing process, the rating protocol recommends less sharing from the center agent to each periphery agent. However, incentives are provided at a cost of reduced social welfare. Figure 3.5 also reveals that when agents' discount factor is lower (agents value less the future utility), incentives are more difficult to provide and hence, the PoA becomes higher. Since our applies to any benefit function that satisfies the Assumption, we show in Figure 3.6 the PoA for different noise variances r^2 for discount factor $\delta = 0.9$. As we can see that the above analysis holds for other values of r^2. Moreover, as the noise variance increases, PoA is smaller for the same network size. This is because the benefit from cooperation increases and hence, agents are more likely to cooperate at the optimal level.

In the next simulation, we study scale-free networks in the imperfect monitoring scenarios. We used the standard Barbasi-Albert (BA) model to create the networks [AB02]. In scale-free networks, the number of neighboring agents is distributed as a power law (denote the power law parameter by d^{SF}). Table 3.4 shows the mean and variance of PoA achieved by the rating protocol developed for various values of d^{SF} and different monitoring error probabilities ϵ. The noise variance is set to be $r^2 = 4$ and the discount factor is $\delta = 0.8$. Each result is obtained by running 100 random trials. As we can see, the proposed rating protocol achieves close-to-optimal social welfare in all the simulated environments. In Table 3.5, we further show the achievable PoA by the proposed rating protocol for scale-free networks of different sizes when $\epsilon = 0.05$. Since the considered network is scale-free, the performance is similar for different network sizes.

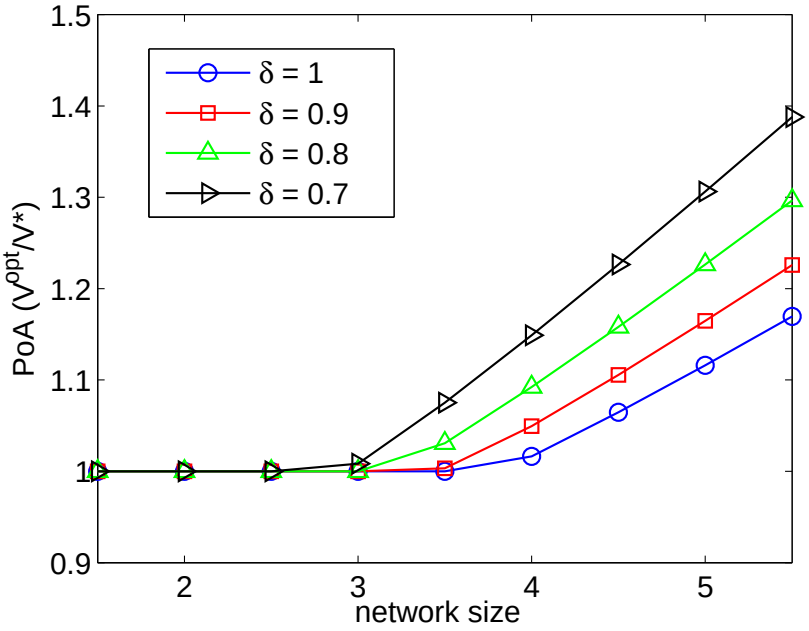


Figure 3.5: Performance for different connectivity degrees d of star networks.

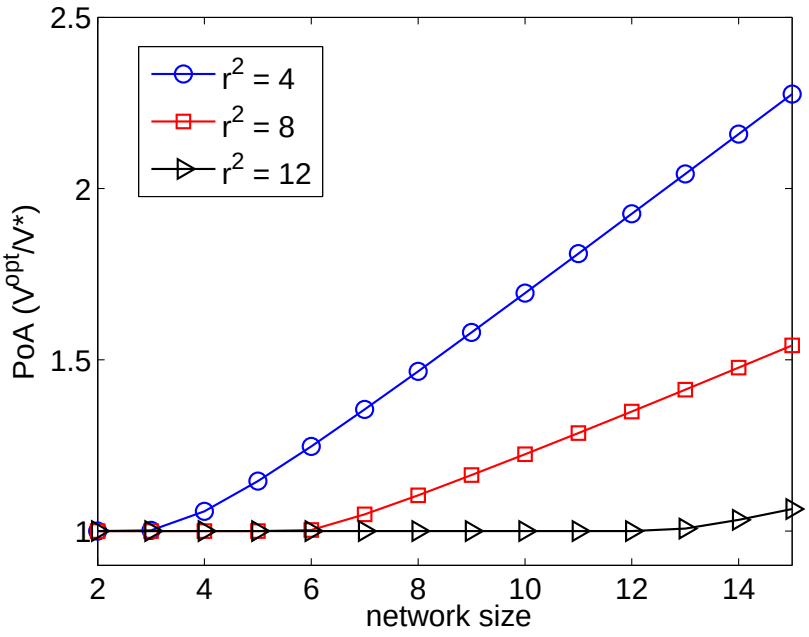


Figure 3.6: Performance for different noise variance r^2.

d^{SF}		$\epsilon = 0$	$\epsilon = 0.05$	$\epsilon = 0.1$
2	*Mean*	1.151	1.174	1.199
	Variance	5.9e-3	6.2e-3	6.4e-3
3	*Mean*	1.154	1.177	1.203
	Variance	8.6e-3	8.8e-3	9.2e-3
4	*Mean*	1.002	1.023	1.046
	Variance	3.1e-5	2.9e-5	2.7e-5
5	*Mean*	1.001	1.022	1.044
	Variance	1.6e-5	1.5e-5	1.4e-5
6	*Mean*	1.000	1.022	1.046
	Variance	~0	5.3e-7	2.5e-6

Table 3.4: Performance for various d^{SF} in scale-free networks.

d^{SF}	N = 100	N = 200	N = 500
2	1.174	1.173	1.176
3	1.177	1.175	1.179
4	1.023	1.023	1.023
5	1.022	1.021	1.022
6	1.022	1.022	1.020

Table 3.5: Performance for scale-free networks of different sizes

3.6.2 Comparison with Tit-for-Tat

As mentioned in the analysis, incentive mechanisms based on direct reciprocation such as Tit-for-Tat do not work in networks lacking bilateral interests between connected agents and hence, reasons to mutually reciprocate. In this simulation, to make possible a direct comparison with the Tit-for-Tat strategy, we consider a scenario where the connected agents do have bilateral interests and show that the proposed rating protocol significantly outperforms the Tit-for-Tat strategy. In general, computing the optimal action profile $\bar{a}^*$ for the Tit-for-Tat strategy is difficult because it involves the non-convex constraint $\delta(b_i(\{\bar{a}^*_{ki}\}_{k:g_{ik}=1}) - b_i(\{\bar{a}^*_{ki}\}_{k\neq j:g_{ik}=1}, 0)) \geq \bar{a}^*_{ij}$, $\forall i, \forall j \neq i : g_{ij} = 1$; such a difficulty is not presented in our proposed rating protocol because the constraints in our formulated problem are convex. For tractability, here we consider a symmetric and homogeneous network to enable the computation of the optimal action for the Tit-for-Tat strategy. We consider a number $N = 100$ of agents and that the number of neighbors of each agent is the same $d_i = d, \forall i$ and each agent adopts a symmetric action profile $\bar{a}_{ij} = \bar{a}, \forall i, j$. The noise vari-

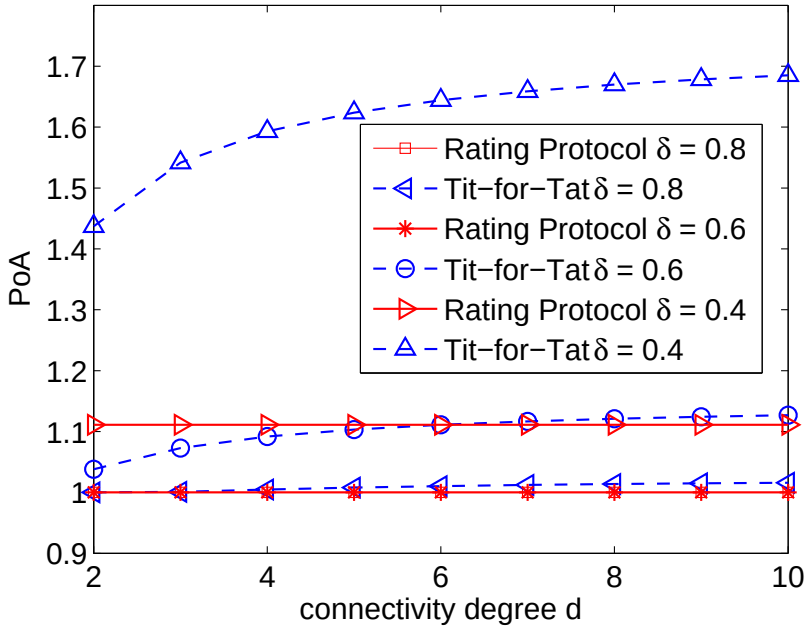


Figure 3.7: Performance comparison with Tit-for-Tat.

ance is set to be $r^2 = 4$ in this simulation. Figure 3.7 illustrates the PoA achieved by the proposed rating protocol and the Tit-for-Tat strategy. As predicted by Proposition 4, any action profile that can be sustained by the Tit-for-Tat strategy can also be sustained by the proposed rating protocol (for the same δ). Hence, the rating protocol yields at least as much social welfare as the Tit-for-Tat strategy (for the same δ). As the discount factor becomes smaller, agents' incentives to cooperate become less and hence, the PoA is larger. Note that for $\delta = 0.6, 0.8$, our rating protocol achieves $PoA = 1$ for all connectivity degrees.

3.6.3 Rating Protocol with Refreshing

Finally, we consider the optimal choice of the rating protocol refreshing rate ρ when the network is growing as considered in section VI. In this simulation, the network starts with $N = 50$ agents. In each period, a new agent enters the network with probability 0.1 and stays in the network forever. Any two agents are connected with *a priori* probability 0.2. We vary the refreshing rate from 0.005 to 0.14. Table 3.6 records the PoA achieved the rating protocol with refreshing for $\delta = 0.4$. It shows that the optimal refreshing rate needs to be

ρ	0.005	0.02	0.04	0.06	0.08	0.10	0.12	0.14
PoA	1.35	1.20	1.18	1.22	1.25	1.29	1.34	1.41

Table 3.6: PoA of rating protocols with different refreshing rates.

carefully chosen. If ρ is too large, the incentives for agents to cooperate is small hence, the incentive-compatible rating protocol achieves less social welfare. If ρ is too small, the rating protocol is not able to adapt to the changing network well. This introduces more social welfare loss in the long-term as well. The optimal refreshing rate in the simulated network is around 0.04.

3.7 Conclusions

In this chapter, we provided a framework for designing incentives protocols (based on ratings) aimed at maximizing the social welfare of strategic agents which are repeatedly sharing information/goods across a network. Our rating protocols can be implemented in a distributed and informationally decentralized manner and achieve much higher social welfare than existing incentive mechanisms. Our framework and analysis can also be used to provide guidelines for designing and planning social, economic and engineering networks of strategic agents, such that the social welfare of such networks is maximized. The proposed ratings framework can also be used to design protocols for a wide range of engineering networks where strategic agents interact - communications networks, power networks, transportation networks, and computer networks.

3.8 Appendix

Proof of Lemma 1

(1) Consider any action $\boldsymbol{a}_i(\boldsymbol{\theta}) \neq \sigma_i(\hat{\boldsymbol{\theta}}_i)$. According to the rating update rule, $p(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{a}_i(\theta)) = p(\boldsymbol{\theta}'|\boldsymbol{\theta}, \mathbf{0})$. Since $u_i(\theta, \mathbf{0}) > u_i(\theta, \boldsymbol{a}_i(\theta))$, we can see that $U_i(\theta, \mathbf{0}) > U_i(\theta, \boldsymbol{a}_i(\theta)))$. Therefore, there are only two possible actions that can potentially maximize the long-term utility.

(2) According to part (1), there are only two possible actions that can be optimal. First, we note that the continuation utility difference by choosing these two actions is

$$\delta \sum_{\boldsymbol{\theta}'} p(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i)) U_i^*(\hat{\boldsymbol{\theta}}_i) - \delta \sum_{\boldsymbol{\theta}'} p(\boldsymbol{\theta}'|\boldsymbol{\theta}, \mathbf{0}) U_i^*(\hat{\boldsymbol{\theta}}_i) \tag{3.27}$$

which is independent of other agents' ratings $\hat{\boldsymbol{\theta}}_i$ when we consider agent i's one-shot unilateral deviation. This is because the benefit that an agent can potentially receive only depends on its own rating while the cost that the agent incurs depends only on its neighbors' ratings. The benefit is determined by agent i's current action since different actions lead to different transitions of only agent i's own rating. The costs are cancelled out because the neighbors' ratings are independent on agent i's actions.

It is obvious that the current period utility different satisfies,

$$\begin{aligned} &u_i((\theta_i, \boldsymbol{K}), \mathbf{0}) - u_i((\theta_i, \boldsymbol{K}), \boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i)) \\ &\geq u_i((\theta_i, \boldsymbol{\theta}_{-i}), \mathbf{0}) - u_i((\theta_i, \boldsymbol{\theta}_{-i}), \boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i)), \forall \theta_{-i} \end{aligned} \tag{3.28}$$

If for $\boldsymbol{\theta}_{-i} = \boldsymbol{K}$, the optimal action of agent i is $\boldsymbol{a}_i^* = \boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i)$, then the following holds,

$$\begin{aligned} &u_i((\theta_i, \boldsymbol{K}), \mathbf{0}) - u_i((\theta_i, \boldsymbol{K}), \boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i)) \\ &\leq \delta \sum_{\boldsymbol{\theta}'} p(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i)) U_i^*(\hat{\boldsymbol{\theta}}_i) - \delta \sum_{\boldsymbol{\theta}'} p(\boldsymbol{\theta}'|\boldsymbol{\theta}, \mathbf{0}) U_i^*(\hat{\boldsymbol{\theta}}_i) \end{aligned} \tag{3.29}$$

which means that the following is also true,

$$\begin{aligned} &u_i((\theta_i, \boldsymbol{\theta}_{-i}), \mathbf{0}) - u_i((\theta_i, \boldsymbol{\theta}_{-i}), \boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i)), \forall \theta_{-i} \\ &\leq \delta \sum_{\boldsymbol{\theta}'} p(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i)) U_i^*(\hat{\boldsymbol{\theta}}_i) - \delta \sum_{\boldsymbol{\theta}'} p(\boldsymbol{\theta}'|\boldsymbol{\theta}, \mathbf{0}) U_i^*(\hat{\boldsymbol{\theta}}_i) \end{aligned} \tag{3.30}$$

Therefore, for any other $\boldsymbol{\theta}_{-i}$, the optimal action of agent i is also $\boldsymbol{a}_i^* = \boldsymbol{\sigma}_i(\hat{\boldsymbol{\theta}}_i)$.

(3) To simplify notations, we suppress $\boldsymbol{K}$ in the utility and simply write $U_i^*(\theta_i)$ instead of $U_i^*(\theta_i, \hat{\boldsymbol{\theta}}_i)$. We also write $\beta_{i,k}$ as β_k. The value functions can be obtained by solving the following recursive equations,

$$\begin{aligned} &U_i^*(K) = u(K, \boldsymbol{\sigma}_i) + \delta U_i^*(K) \\ &U_i^*(K-1) = u(K-1, \boldsymbol{\sigma}_i) + \\ &\qquad \delta(\beta_{K-1} U_i^*(K) + (1-\beta_{K-1}) U_i^*(K-1)) \\ &...U_i^*(1) = u(1, \boldsymbol{\sigma}_i) + \delta(\beta_1 U_i^*(2) + (1-\beta_1) U_i^*(1)) \end{aligned} \tag{3.31}$$

We prove by induction. Suppose $U_i^*(l) \geq U_i^*(l-1), \forall l : K \geq l \geq k+1$. We need to show that $U_i^*(k) \geq U_i^*(k-1)$. The value functions of level k and $k-1$ are

$$\begin{aligned} U_i^*(k) &= u_i(k, \boldsymbol{\sigma}_i) + \delta(\beta_k U_i^*(k+1) + (1-\beta_k) U_i^*(k)) \\ U_i^*(k-1) &= u_i(k-1, \boldsymbol{\sigma}_i) + \delta(\beta_{k-1} U_i^*(k) \\ &+ (1-\beta_k) U_i^*(k-1)) \end{aligned} \tag{3.32}$$

To prove $U_i^*(k) \geq U_i^*(k-1)$, we use contradiction. Suppose $U_i^*(k) < U_i^*(k-1)$, then

$$\begin{aligned} U_i^*(k) &\geq u_i(k, \boldsymbol{\sigma}_i) + \delta U_i^*(k) \\ U_i^*(k-1) &< u_i(k-1, \boldsymbol{\sigma}_i) + \delta U_i^*(k-1) \end{aligned} \tag{3.33}$$

This leads to $u_i(k, \boldsymbol{\sigma}_i) < u_i(k-1, \boldsymbol{\sigma}_i)$ which is a contradiction. Hence, it only remains to prove $U_i^*(K) \geq U_i^*(K-1)$. This can be easily shown by computing $U_i^*(K) - U_i^*(K-1)$, i.e.

$$U_i^*(K) - U_i^*(K-1) = \frac{u_i(K, \boldsymbol{\sigma}_i) - u_i(K-1, \boldsymbol{\sigma}_i)}{1 - \delta(1-\beta_{K-1})} > 0 \tag{3.34}$$

This completes the proof.

Proof of Theorem 1

According to Lemma 1, it suffices to ensure that agent i has an incentive to take the recommended strategy when it neighbors' ratings are $\hat{\theta}_i = \mathbf{K}$. However, we need to prove that this holds for all ratings of agent i. Therefore, we suppress $\hat{\theta}_i = \mathbf{K}$ and only write out θ_i whenever it is clear.

We prove the "only if" part first. We need to show that for all rating protocol that is an equilibrium, $\delta b_i(\hat{\boldsymbol{\sigma}}_i(K)) \geq c(\boldsymbol{\sigma}(\boldsymbol{K})), \forall i$ must be satisfied. Consider any rating level k of agent i, following the recommended strategy gives it the following long-term utility,

$$U_i(k, \boldsymbol{\sigma}) = u_i(k, \boldsymbol{\sigma}) + \delta(\beta_k U_i^*(k+1) + (1-\beta_k) U_i^*(k)) \tag{3.35}$$

Deviating to $\mathbf{0}$ gives the following long-term utility,

$$U_i(k, \mathbf{0}) = u_i(k, \mathbf{0}) + \delta(\alpha_k U_i^*(k-1) + (1-\alpha_k) U_i^*(k)) \tag{3.36}$$

Equilibrium requires that $U_i(k, \boldsymbol{\sigma}) \geq U_i(k, \mathbf{0})$. Therefore, the following must hold,

$$\begin{aligned} u_i(k, \mathbf{0}) - u_i(k, \boldsymbol{\sigma}) \leq \delta[\beta_k U_i^*(k+1) + (1-\beta_k) U_i^*(k) \\ - \alpha_k U_i^*(k-1) - (1-\alpha_k) U_i^*(k)] \end{aligned} \tag{3.37}$$

According to Lemma 1.3, $U_i^*(K) \geq U_i^*(k), \forall k$ in an equilibrium. Therefore, the following must hold,

$$u_i(k, \mathbf{0}) - u_i(k, \boldsymbol{\sigma}) \leq \delta U_i^*(K) \tag{3.38}$$

The left-hand side is $u_i(k, \mathbf{0}) - u_i(k, \boldsymbol{\sigma}) = c(\boldsymbol{\sigma}_i)$. Using the recursive equation of the optimal long-term utilities (3.32), we can compute the right-hand side as

$$U_i^*(K) = \frac{1}{1-\delta} u_i(1, \boldsymbol{\sigma}_i) = \frac{1}{1-\delta}(b_i(\hat{\boldsymbol{\sigma}}_i(1)) - c_i(\boldsymbol{\sigma}_i(\mathbf{K}))). \tag{3.39}$$

Substituting this into (3.38), we can obtain the desired result after simple manipulations.

Next, we prove the "if" part by constructing a binary rating protocol. According to the one-shot deviation principle, for agent i to follow the recommended strategy at $\theta_i = 2$, we need

$$u_i(2, \mathbf{0}) - u_i(2, \boldsymbol{\sigma}_i) \leq \delta \alpha_2 (U_i^*(2) - U_i^*(1)) \tag{3.40}$$

for agent i to follow the recommended strategy at $\theta_i = 1$, we need

$$u_i(1, \mathbf{0}) - u_i(1, \boldsymbol{\sigma}_i) \leq \delta \beta_1 (U_i^*(2) - U_i^*(1)) \tag{3.41}$$

Using the value function (3.32), we can compute $U_i^*(2) - U_i^*(1)$ which is

$$U_i^*(2) - U_i^*(1) = \frac{u_i(2, \boldsymbol{\sigma}_i) - u_i(1, \boldsymbol{\sigma}_i)}{1 - \delta(1-\beta_1)} \tag{3.42}$$

Moreover, $u_i(2, \mathbf{0}) - u_i(2, \boldsymbol{\sigma}_i) = u_i(1, \mathbf{0}) - u_i(1, \boldsymbol{\sigma}_i) = c_i(\boldsymbol{\sigma}_i)$. For the rating protocol to be an equilibrium, we need to choose α_2, β_1 such that

$$\begin{aligned} c_i(\boldsymbol{\sigma}_i) &\leq \{\alpha_2, \beta_1\} \tfrac{u_i(2, \boldsymbol{\sigma}_i) - u_i(1, \boldsymbol{\sigma}_i)}{1 - \delta(1-\beta_1)} \\ &= \{\alpha_2, \beta_1\} \tfrac{\delta b_i(\hat{\boldsymbol{\sigma}}_i(2))}{1 - \delta(1-\beta_1)} \end{aligned} \tag{3.43}$$

If we choose $\alpha_2 = \beta_1 = 1$, then the above inequality holds. This means that such a binary rating protocol is a PLE.

Discussion on the DCRS algorithm

When developing the DCRS algorithm, we used the widely-adopted dual decomposition method. However, there are several significant differences from existing problems.

First, in most existing problems [PC07] [BV04], the constraint in the optimization problem comes from the system resource constraints. Our problem is not a NUM problem since we do not have such resource constraints. Instead, the constraints are derived based on the incentive-compatibility of agents, i.e. the incentive condition under which the agents follow the recommended strategy. More specifically, they are derived in Theorem 1 (in the revised manuscript).

Second, in many standard dual decomposition problems [PC07] [BV04], the objective functions are directly separable, in the sense that an agent's utility depends on its own action. The coupling among agents only comes from the optimization constraints. For example, the objective function can have the form $\sum_i f_i(\boldsymbol{x}_i)$ where $\boldsymbol{x}_i$ is agent i's action and $f_i(\boldsymbol{x}_i)$ is its utility. The actions of all agents need to satisfy some resource constraints $\sum_i h_i(\boldsymbol{x}_i) \leq 0$. In our problem, an agent's utility depends not only on its own action but also on the neighboring agents' actions, i.e. $\sum_i (b_i(\hat{\boldsymbol{\sigma}}_i) - c_i(\boldsymbol{\sigma}_i))$ where $\boldsymbol{\sigma}_i$ is agent i's strategy and $\hat{\boldsymbol{\sigma}}_i$ is agent i's neighbors strategies towards agent i.

Third, even though dual decomposition allows distributed implementation, in many existing works [PC07] [BV04], agents still need to exchange messages with all other agents (e.g. by broadcasting). This requires intensive message exchanges among agents if broadcasting is not available and is even impossible if agents' interactions are subject to underlying topologies. However, our solution enables a completely distributed architecture and message exchange only occurs between connected agents.

CHAPTER 4

Distributed Multi-Agent Online Learning

In this chapter, we consider a multi-agent decision making and learning problem, in which a set of distributed agents select actions from their own action sets in order to maximize the overall system reward which depends on the joint action of all agents. In the considered scenario, agents do not know *a priori* how their actions influence the overall system reward, or how their influence may change dynamically over time. Therefore, in order to maximize the overall system reward, agents must dynamically learn how to select their best actions over time. But agents can only observe/measure the *overall* system performance and hence, they only obtain global feedback that depends on the joint actions of all agents. Since individualized feedback about individual actions is absent, it is impossible for the agents to learn how their actions alone affect the overall performance *without* cooperating with each other. However, because agents are distributed they are unable to communicate and coordinate their action choices. Moreover, agents' observations of the global feedback may be subject to individual errors, and thus it may be extremely difficult for an agent to conjecture other agents' actions based solely on its own observed reward history. The fact that individualized feedback is missing, communication is not possible, and the global feedback is noisy makes the development of efficient learning algorithms which maximize the joint reward very challenging. Importantly, the considered multi-agent learning scenario differs significantly from the existing solutions [AMT11] [TL12] [LLZ13], in which agents receive individualized rewards. To help illustrate the differences, Figures 1(a) and (b) portray conventional learning in multi-agent systems based on individualized feedback and the considered learning in multi-agent systems based on global feedback with individual noise, respectively. The considered problem has many application scenarios. For instance, in a stream mining system that uses

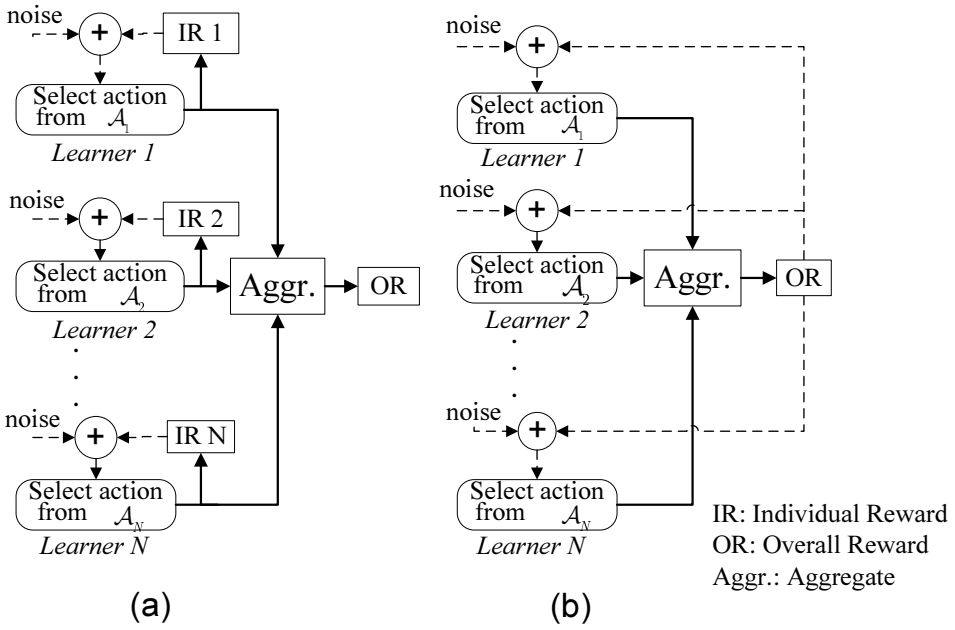


Figure 4.1: Learning in multi-agent systems with (a) individualized feedback; (b) global feedback (this work).

multiple classifiers for event detection in video streams, individual classifiers select operating points to classify specific objects or actions of interest, the results of which are synthesized to derive an event classification result. If the global feedback is only about whether the event classification is correct or not, individualized feedback about individual contribution is not available. For another instance, in a cooperative communication system, a set of wireless nodes forward signals of the same message copy to a destination node through noisy wireless channels. Each forwarding node selects its transmission scheme (e.g. power level) and the destination combines the forwarded signals to decode the original message using, e.g., a maximal ratio combination scheme. Since the message is only decoded using the combined signal but not individual signals, only a global reward depending on the joint effort of the forwarding nodes is available but not the nodes' individual contributions.

In this work, we formalize for the first time the above multi-agent decision making framework and propose a systematic solution based on the theory of multi-armed bandits [LR85] [ACF02]. We propose multi-agent learning algorithms which enable the various agents to individually learn how to make decisions to maximize the overall system reward

without exchanging information with other agents. In order to quantify the loss due to learning and operating in an unknown environment, we define the regret of an online learning algorithm for the set of agents as the difference between the expected reward of the best joint action of all agents and the expected reward of the algorithm used by the agents. We prove that, if the global feedback is received without errors by the agents, then all deterministic algorithms can be implemented in a distributed manner without message exchanges. This implies that the distributed nature of the system does not introduce any performance loss compared with a centralized system since there exist deterministic algorithms that are optimal. Subsequently, we show that if agents receive the global feedback *with* different (individual) errors, existing deterministic algorithms may break down and hence, there is a need for novel distributed algorithms that are robust to such errors. For this, we develop a class of algorithms which achieve a logarithmic upper bound on the regret, implying that the average reward converges to the optimal average reward[1]. The upper bound on regret also gives a lower bound on the convergence rate to the optimal average reward. For our first algorithm, DisCo, we start without any additional assumptions on the problem structure and show that the regret is still logarithmic in time. Although, the time order of the regret of the DisCo algorithm is logarithmic, due to its linear dependence on the cardinality of the joint action space, which increases exponentially with the number of agents, the regret is large and the convergence rate is very slow with many agents. Next, we define the informativeness of the overall reward function based on how effectively agents are able to distinguish the impact of their actions from the actions of others, and we exploit this informativeness in order to design improved learning algorithms. When the overall reward function is fully informative about the optimality of individual actions, the improved learning algorithm Disco-FI achieves a regret that is linear in the size of the action space of each agent, and logarithmic in time. The crucial idea behind this result is that, when the overall reward is fully informative, instead of using the *exact* reward estimates of every joint action, the agents can use the *relative* reward estimates of each individual action to learn their optimal

[1]It is shown in [LR85] that logarithmic regret is the best possible even for simple single agent learning problems. However, convergence to the optimal reward is a much weaker result than logarithmic regret. Any algorithm with a sublinear bound on regret will converge to the optimal average reward asymptotically.

actions at a much faster speed. Finally, we consider a more general setting where the global rewards are only partially informative. Our third algorithm (DisCo-PI) works when the overall reward function is informative for a group of agents instead of each agent individually, and it achieves a regret which is in between the first two algorithms. As an application of our theoretical framework, we then run simulations utilizing our algorithms for the problem of online Big Data mining using distributed classifiers [FTV07] [FS10] [FV09] [DTS10]. We show that the proposed algorithms achieve a very high classification accuracy when compared with existing solutions. Our framework could also be similarly applied to many other applications including online distributed decision making in cooperative multi-agent systems such as multi-path or multi-hop networks, cross-layer design, multi-core processing systems, etc.

4.1 Related Works

The literature on multi-armed bandit problems can be traced back to [Git79] [Whi80] which studies a Bayesian formulation and requires priors over the unknown distributions. In our work, such information is not needed. A general policy based on upper confidence bounds is presented in [LR85] that achieves asymptotically logarithmic regret in time given that the rewards from each arm are drawn from an independent and identically distributed (i.i.d.) process. It also shows that no policy can do better than $\Omega(K \ln t)$ [2] (i.e. linear in the number of arms and logarithmic in time) and therefore, this policy is order optimal in terms of time. In [ACF02], upper confidence bound (UCB) algorithms are presented which are proved to achieve logarithmic regret uniformly over time, rather than only asymptotically. These policies are shown to be order optimal when the arm rewards are generated independently of each other. When the rewards are generated by a Markov process, algorithms with logarithmic regret with respect to the best static policy are proposed in [RT10] and [Aue03]. However, all of these algorithms intrinsically assume that the reward process of each arm is independent, and hence they do not exploit any correlations that might be present between

[2] We adopt the standard asymptotic notations $\Omega(\cdot)$ and $O(\cdot)$ as in [CLR01].

the rewards of different arms. In this work the rewards may be highly correlated, and so it is important to design algorithms that take this into account.

Another interesting bandit problem, in which the goal is to exploit the correlations between the rewards, is the combinatorial bandit problem [CL12]. In this problem, the agent chooses an action vector and receives a reward which depends on some linear or non-linear combination of the individual rewards of the actions. In a combinatorial bandit problem the set of arms grows exponentially with the dimension of the action vector; thus standard bandit policies like the one in [ACF02] will have a large regret. The idea in these problems is to exploit the correlations between the rewards of different arms to improve the learning rate and thereby reduce the regret [AVW87] [AMT11]. Most of the works on combinatorial bandits assume that the expected reward of an arm is a linear function of the chosen actions for that arm. For example [GKJ12] assumes that after an action vector is selected, the individual rewards for each non-zero element of the action vector are revealed. Another work [CWY13] considers combinatorial bandit problems with more general reward functions, defines the approximation regret and shows that it grows logarithmically in time. The approximation regret compares the performance of the learning algorithm with an oracle that acts approximately optimally, while we compare our algorithm with the optimal policy. This work also assumes that individual observations are available. However, in this work we assume that only global feedback is available and individuals cannot observe each other's actions. Agents have to learn their optimal actions based only on the feedback about the overall reward. Other bandit problems which use linear reward models are studied in [RT10] [Aue03] [DHK08]. These consider the case where only the overall reward of the action profile is revealed but not the individual rewards of each action. However, our analysis is not restricted to linear reward models, but instead much more general. In addition, in most of the previous work on multi-armed bandits [LR85] [ACF02] [AVW87] [AMT11], the rewards of the actions (arms) are assumed to come from an unknown but fixed distribution. We also have this assumption in most of our analysis in this work. However, in Section VII we propose learning algorithms which can be used when the distribution over rewards is changing over time (i.e. exhibits

accuracy drift[3]).

Another line of work considers online optimization problems, where the goal is to minimize the loss due to learning the optimal vector of actions which maximizes the expected reward function. These works show sublinear (greater than logarithmic) regret bounds for linear or submodular expected reward functions, when the rewards are generated by an adversary to minimize the gain of the agent. The difference of our work is that we consider a more general reward function and prove logarithmic regret bounds. Recently, distributed bandit algorithms are developed in [SBH13] in network settings. In that work, agents have the same set of arms with the same unknown distributions and are allowed to communicate with neighbors to share their observed rewards. In contrast, in the current work, agents have distinct sets of arms, the reward depends on the joint action of agents and agents do not communicate at run-time.

4.2 System Model

There are N agents indexed by the set $\mathcal{N} = \{1, 2, ..., N\}$. Each agent has access to an action set $\mathcal{A}_n$, with the cardinality of the action set denoted by $K_n = |\mathcal{A}_n|$. Since we model the system using the multi-armed bandit framework, we will use "arm" and "action" interchangeably in this work. In addition to the number of its own arms, each agent n knows the number of arms K_j of all the other agents $j \neq n$. The model is set in discrete time $t = 1, 2, ..., T$. In each time slot, each agent selects a single one of its own arms $a_n(t) \in \mathcal{A}_n$. Agents are distributed and thus cannot observe the arm selections of the other agents. We denote by $\boldsymbol{a}(t)$ the vector of arm selections by all the agents at time t, which we call the *joint arm* that is selected at time t.

Given any joint arm selection, a random reward $r_t(\boldsymbol{a}(t))$ will be generated according to an unknown distribution, with a dynamic range bounded by a value D'. For now, we will assume that this global reward is i.i.d. across time. We denote the expected reward given a selection $\boldsymbol{a}(t)$ by $\mu(\boldsymbol{a})$=$\mathbb{E}[r_t(\boldsymbol{a}(t))]$. The agents do not know the reward function $\mu(\boldsymbol{a})$

[3]Accuracy drift is more general than concept drift and is formally defined later in Section VIII.

initially and must learn it over time. Every period each agent n privately observes a signal $r_t^n = r_t + \epsilon_t^n$, equal to the global reward r_t plus a random noise term ϵ_t^n. We assume that ϵ_t^n has zero mean, is bounded in magnitude by D'' and i.i.d. across time, but it does not need to be i.i.d. across agents. Let $D = D' + D''$. Agents cannot communicate, so at any time t each agent has access to only its own history of noisy reward observations $\mathcal{H}_t^n = \{r_\tau^n\}_{\tau=1}^t$.

Agents operate according to an *algorithm* $\pi_n(\mathcal{H}_t^n)$, which tells it which arm to choose after every history of observations. This algorithm can be deterministic, meaning that given any history it will map to a unique arm, or probabilistic, meaning that for some histories it will map to a probability distribution over arms. Let $\pi(\mathcal{H}_t^1, ..., \mathcal{H}_t^N)$=$\{\pi_1(\mathcal{H}_t^1), ..., \pi_N(\mathcal{H}_t^N)\}$ denote the *joint algorithm* that is used by all agents after every possible history of observations. Since agents cannot communicate, the joint algorithm may only select actions for each agent based on that agent's private observation history. We denote the joint arm selected at time t given the joint algorithm as $\boldsymbol{a}^\pi(t)$. Fixing any joint algorithm, we can compute the expected reward at time 0 as $\mathbb{E}\sum_{t=1}^T r_t(\boldsymbol{a}^\pi(t))$.

This work will propose a group of joint algorithms that can achieve sublinear regret in time given different restrictions on the expected reward function $\mu(\boldsymbol{a})$. Denote the optimal joint action by $\boldsymbol{a}^* := \arg\max_{\boldsymbol{a}} \mu(\boldsymbol{a})$. We will always assume that the optimal joint action is unique. The regret of a joint algorithm $\pi(\mathcal{H}_t^1, ..., \mathcal{H}_t^N)$ is given by

$$R(T;\pi) := T\mu(\boldsymbol{a}^*) - \mathbb{E}\sum_{t=1}^{T} r_t(\boldsymbol{a}^\pi(t)) \tag{4.1}$$

Regret gives the convergence rate of the total expected reward of the learning algorithm to the value of the optimal solution. Any algorithm whose regret is sublinear will converge to the optimal solution in terms of the average reward.

4.3 Robustness of Algorithms with Distributed Implementation

In the considered setting, there is no individual reward observation associated with each individual arm but only an overall reward which depends on the arms selected by all agents. Therefore agents have to learn how their individual arm selections influence the overall

reward, and choose the best joint set of arms in a cooperative but isolated manner. In general, agents may observe different noisy versions of the overall reward realization at each time, so we would like the algorithms to be robust to errors and perform efficiently in a noisy environment. But we will start by considering situations where there are no errors, and show that in this case agents are able to achieve the optimal expected reward even if they are distributed and unable to communicate.

4.3.1 Scenarios without individual observation errors

Let Π^c be the set of algorithms that can be implemented in a scenario where agents are allowed to exchange messages (reward observations, selected arms etc.) at run-time. Let Π^d be the set of algorithms that can be implemented in scenarios where agents cannot exchange messages at run-time. Obviously $\Pi^d \subseteq \Pi^c$. At the first sight, it seems that the restrictions on communication may result in efficiency loss compared to the scenario where agents can exchange messages. Next, we prove a perhaps surprising result – there is no efficiency loss even if agents cannot exchange messages at run-time as long as the agents observe the same overall reward realization in each time slot. Such a result is thus applicable if there are no errors, or even if the error terms, ϵ_t , are the same for every agent at every time t.

Theorem 2. *If agents observe the same reward realization in each time slot, then* $\min_{\pi \in \Pi^c} R(T;\pi) = \min_{\pi \in \Pi^d} R(T;\pi)$, $\forall T$.

Proof. See Appendix. □

Theorem 1 reveals that even if agents are distributed and not able to exchange messages at run-time, all existing deterministic algorithms proposed for centralized scenarios can still be used when agents observe the same reward realizations. The reason is that even though agents cannot directly communicate, as long as they know the algorithms of the other agents before runtime they can correctly predict which arms the other agents will choose based on the global reward history. In particular, the classic UCB1 algorithm can be implemented in distributed scenarios without loss of performance.

4.3.2 Scenarios with individual observation errors

When agents observe different noisy versions of the reward realizations, it is difficult for them to infer the correct actions of other agents based on their own private reward histories since their beliefs about others could be wrong and inconsistent. For instance, one agent may observe a high reward for a joint arm, while another agent observes a low reward. Then the first agent may decide to keep playing that joint arm, and believe that the other agent is also still playing it, while in actuality the other agent has already moved on to testing other joint arms. In such scenarios, even a single small observation error could cause inconsistent beliefs among agents and lead to error propagation that is never corrected in the future. In Proposition 1, we show this effect for the classic UCB1 algorithm and prove that UCB1 is not robust to errors when implemented in a distributed way.

Proposition 7. *In distributed networks where agents do not exchange messages at run-time, if the observations of the overall reward realization are subject to individual errors, then the expected regret of the distributed version of UCB1 algorithm, in which each agent keeps an instance of UCB1 for its own actions and $N-1$ different instances of UCB1 for the actions of other agents, can be linear.*

Proof. See Appendix. □

Proposition 1 implies that even if we implement existing deterministic algorithms such as UCB1 for distributed agents using the reward history, there is no guarantee on their performance when individual observation errors exist. Therefore, there is a need to develop new algorithms that are robust to errors in distributed scenarios. In the next few sections, we will propose such a class of algorithms that are robust to individual errors and can achieve logarithmic regret in time even in the noisy environment.

4.4 Distributed Cooperative Learning Algorithm

In this section, we propose the basic distributed cooperative learning (DisCo) algorithm which is suitable for any overall reward function. The proposed learning algorithm achieves logarithmic regret (i.e. $R(T) = O(\prod_{n=1}^{N} K_n \ln T)$). In Sections VI and VII, we will identify some useful reward structures and exploit them to design improved learning algorithms (DisCo-FI and DisCo-PI algorithms) which achieve even better regret results.

4.4.1 Description of the Algorithm

The DisCo algorithm is divided into phases: exploration and exploitation. Each agent using DisCo will alternate between these two phases, in a way that at any time t, either all agents are exploring or all are exploiting. In the exploration phase, each agent selects an arm only to learn about the effects on the expected reward, without considering reward maximization, and updates the reward estimates of the arm it selected. In the exploitation phase, each agent exploits the best (estimated) arm to maximize the overall reward.

Knowledge, Counters and Estimates: There is a deterministic control function $\zeta(t)$ of the form $\zeta(t) = A \ln t$ commonly known by all agents. This function will be designed and determined before run-time, and thus is an input of the algorithm. Each exploration phase has a fixed length of $L_1 = \prod_{n=1}^{N} K_n$ slots, equal to the total number of joint arms. Each agent maintains two counters [4]. The first counter $\gamma(t)$ records the number of exploration phases that they have experienced by time slot t. The second counter $E(t) \in \{0, 1, ..., L_1\}$ represents whether the current slot is an exploration slot and, if yes, which relative position it is at. Specifically, $E(t) = 0$ means that the current slot is an exploitation slot; $E(t) > 0$ means that the current slot is the $E(t)$-th slot in the current exploration phase. Both counters are initialized to zero: $\gamma(0) = 0, E(0) = 0$. Each agent n maintains L_1 sample mean reward estimates $\bar{r}^n(l)$ $\forall l \in \{1, ..., L_1\}$, one for each relative slot position in an exploration phase. Let b_l^n denote the arm selected by agent n in the l-th position in an exploration phase.

[4] Agents maintain these counters by themselves, $\gamma^n(t), E^n(t)$ $\forall n$. However, since agents update these counters in the same way, the superscript for the agent index is neglected in our analysis.

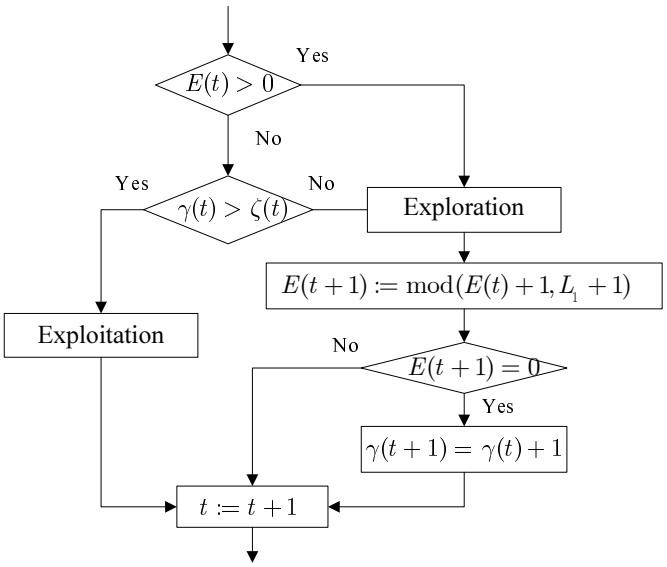


Figure 4.2: Flowchart of the phase transition.

These reward estimates are initialized to be $\bar{r}^n(l) = 0$ and will be updated over time using the realized rewards (the exact updating method will be explained shortly).

Phase Transition: Whether a new slot t is an exploration slot or an exploitation slot will be determined by the values of $\zeta(t), \gamma(t)$ and $E(t)$. At the beginning of each slot t, the agents first check the counter $E(t)$ to see whether they are still in the exploration phase: if $E(t) > 0$, then the slot is an exploration slot; if $E(t) = 0$, whether the slot is an exploration slot or an exploitation slot will then be determined by $\gamma(t)$ and $\zeta(t)$. If $\gamma(t) \leq \zeta(t)$, then the agents start a new exploration phase, and at this point $E(t)$ is set to be $E(t) = 1$. If $\gamma(t) > \zeta(t)$, then the slot is an exploitation slot. At the end of each exploration slot, counter $E(t+1)$ for the next slot is updated to be $E(t+1) \leftarrow mod(E(t)+1, L_1+1)$. When $E(t+1) = 0$, the current exploration phase ends, and hence the counter $\gamma(t+1)$ for the next slot is updated to be $\gamma(t+1) \leftarrow \gamma(t)+1$. Figure 2 provides the flowchart of the phase transition for the algorithm.

Prescribed Actions: The algorithm prescribes different actions for agents in different slots and in different phases.

(i) *Exploration phase*: As clear from the Phase Transition, an exploration phase consists of L_1 slots. In each phase, the agents select their own arms in such a way that every joint

arm is selected exactly once. This is possible without communication if agents agree on a selection order for the joint arms before run-time. At the end of each exploration slot (the l^{th} slot), $\bar{r}^n(l)$ is updated to

$$\bar{r}^n(l) \leftarrow \frac{(\gamma(t)-1)\bar{r}^n(l) + r_t^n}{\gamma(t)} \tag{4.2}$$

Note that the observed reward realization r_t^n at time t may be different for different agents due to errors.

(ii) *Exploitation phase*: Each exploitation phase has a variable length which depends on the control function $\zeta(t)$ and counter $\gamma(t)$. At each exploitation slot t, each agent n selects $a_n = \{b_{l^*}^n : l^* = \arg\max_l \bar{r}^n(l)\}$. That is, each agent n selects the arm with the best reward estimate among $\bar{r}^n(l)$, $\forall l \in \{1, ..., L_1\}$. Note that in the exploitation slots, an agent n does not need to know other agents' selected arms. Since agents have individual observation noises, it is also possible that l^* is different for different agents.

4.4.2 Analysis of the regret

At any exploitation slot, agents need sufficiently many reward observations from all sets of arms in order to estimate the best joint arm correctly with a probability high enough such that the expected number of mistakes is small. On the other hand, if the agents spend too much time in exploring, then the regret will be too large because they are not exploiting the best joint arm sufficiently often. The control function $\zeta(t)$ determines when the agents should explore and when they should exploit and hence balances exploration and exploitation. In Theorem 2, we will establish conditions on the control function $\zeta(t)$ such that the expected regret bound of the proposed DisCo algorithm is logarithmic in time. Let $\Delta^{max} = \max_{\boldsymbol{a} \neq \boldsymbol{a}^*} \{\mu(\boldsymbol{a}^*) - \mu(\boldsymbol{a})\}$ be the maximum reward loss by selecting any suboptimal joint arm, and let $\Delta^{min} = \min_{\boldsymbol{a} \neq \boldsymbol{a}^*} \{\mu(\boldsymbol{a}^*) - \mu(\boldsymbol{a})\}$ be the reward difference between the best joint arm and the second-best joint arm.

Theorem 3. *If* $\zeta(t) = A \ln t$ *with* $A > 2\left(\frac{D}{\Delta^{min}}\right)^2$, *then the expected regret of the DisCo*

algorithm after any number T periods is bounded by

$$R(T) \leq AL_1\Delta^{max} \ln T + B_1 \tag{4.3}$$

where $B_1 = L_1\Delta^{max} + \sum_t^{\infty} 2NL_1\Delta^{max} t^{-\frac{A}{2}\left(\frac{\Delta^{min}}{D}\right)^2}$ *is a constant.*

Proof. See Appendix. □

The regret bound proved in Theorem 2 is logarithmic in time which guarantees convergence in terms of the average reward, i.e. $\lim_{T\to\infty} \mathbb{E}[R(T)]/T = 0$. In fact, the order of the regret bound, i.e. $O(L_1 \ln T)$, is the lowest possible that can be achieved [LR85]. However, since the impact of individual arms on the overall (expected) reward is unknown and may be coupled in a complex way, it is necessary to explore *every* possible joint arm to learn its performance. This leads to a large constant that multiplies $\ln T$ which is on the order of $L_1 = \prod_{n=1}^{N} K_n$. If there are many agents, then $\prod_{n=1}^{N} K_n$ will be very large and hence, a large reward loss will be incurred in the exploration phases. This motivates us to design improved learning algorithms which do not require exploring all possible joint arms in order to improve the learning regret. In the next section, we will explore the informativeness (defined formally later) of the expected reward function to develop improved learning algorithms based on the basic DisCo algorithm. We first consider the best case (Full Informativeness) and then extend to the more general case (Partial Informativeness).

4.5 A Learning Algorithm for Fully Informative Rewards

In many application scenarios, even if we do not know exactly how the actions of agents determine the expected overall rewards, some structural properties of the overall reward function may be known. For example, in the classification problem which uses multiple classifiers [FS10], the overall classification accuracy is increasing in each individual classifier's accuracy, even though each individual's optimal action is unknown a priori. Thus, some overall reward functions may provide higher levels of informativeness about the optimality of individual actions. In this section, we will develop learning algorithms that achieve improved regret results and faster learning speed by exploiting such information.

4.5.1 Reward Informativeness

We first define the informativeness of an expected overall reward function.

Definition 2. *(Informativeness) An expected overall reward function $\mu(\boldsymbol{a})$ is said to be informative with respect to agent n if there exists a unique arm $a_n^* \in \mathcal{A}_n$ such that $\forall \boldsymbol{a}_{-n}$,, $a_n^* = \arg\max_{a_n} \mu(a_n, \boldsymbol{a}_{-n})$.*

In words, if the reward is informative with respect to agent n, then for any choices of arms selected by other agents, agent n's best arms in terms of the expected overall reward is the same. Lemma 1 helps explain why such a reward function is "informative".

Lemma 2. *Suppose that $\mu(a)$ is informative with respect to agent n and the unique optimal arm is a_n^*, then the following is true:*

$$a_n^* = \arg\max_{a_n} \sum_{\boldsymbol{a}_{-n}} \theta_{\boldsymbol{a}_{-n}} \mu(a_n, \boldsymbol{a}_{-n}), \forall \theta_{\boldsymbol{a}_{-n}} \geq 0, \sum_{\boldsymbol{a}_{-n}} \theta_{\boldsymbol{a}_{-n}} = 1 \tag{4.4}$$

Proof. This is a direct result of the Definition 1. □

Lemma 1 states that, for an agent n, the weighted average of the expected overall reward over all possible choices of arms by other agents is maximized at the optimal arm a_n^*. Moreover, the optimal arm is the same for all possible weights $\theta_{\boldsymbol{a}_{-n}}$, $\forall \boldsymbol{a}_{-n}$. It further implies that instead of using the *exact* expected overall reward estimate $\bar{r}^n(a_n, \boldsymbol{a}_{-n})$ to evaluate the optimality of an arm a_n, agent n can also use the *relative* overall reward estimate (i.e. the weighted average reward estimate). In this way, agent n needs to maintain only K_n relative overall reward estimates $\bar{r}^n(a_n)$ by selecting the arm a_n and can use these estimates to learn and select the optimal arm. In particular, let $w_{a_n, \boldsymbol{a}_{-n}}$ be the number of times that the joint arm $(a_n, \boldsymbol{a}_{-n})$ is selected in the exploration slots that are used to estimate $\bar{r}(a_n)$. Then,

$$\mathbb{E}\bar{r}^n(a_n) = \frac{\sum_{a_{-n}} w_{a_n, \boldsymbol{a}_{-n}} \mu(a_n, \boldsymbol{a}_{-n})}{\sum_{a_{-n}} w_{a_n, \boldsymbol{a}_{-n}}} \tag{4.5}$$

If $w_{a_n, \boldsymbol{a}_{-n}} = w_{\tilde{a}_n, \boldsymbol{a}_{-n}}, \forall a_n, \tilde{a}_n, \forall \boldsymbol{a}_{-n}$, then we have $\frac{w_{a_n, \boldsymbol{a}_{-n}}}{\sum_{a_{-n}} w_{a_n, \boldsymbol{a}_{-n}}} \triangleq \theta_{\boldsymbol{a}_{-n}}, \forall a_n$. Note that we don't need to know the exact value of $w_{a_n, \boldsymbol{a}_{-n}}$as long as $w_{a_n, \boldsymbol{a}_{-n}} = w_{\tilde{a}_n, \boldsymbol{a}_{-n}}, \forall a_n, \tilde{a}_n, \forall \boldsymbol{a}_{-n}$. Therefore, the relative reward estimates $\bar{r}^n(a_n)$ can be used to learn the optimal action a_n^* even if agents are not exactly sure what arms have been played by other agents.

Definition 3. *(Fully Informative) An expected overall reward function $\mu(a)$ is said to be fully informative if it is informative with respect to all agents.*

If the overall reward function is fully informative, then the agents only need to record the relative overall reward estimates instead of the exact overall reward estimates. Therefore, the key design problem of the learning algorithm is, for each agent n, to ensure that the weights in (4.4) are the same for the relative reward estimates of all its arms so that it is sufficient for agent n to learn the optimal arm using only these relative reward estimates.

We emphasize the importance of the weights $\theta_{\boldsymbol{a}_{-n}}, \forall \boldsymbol{a}_{-n}$ being the same for all $a_n \in \mathcal{A}_n$ of each agent n even though agent n does not need to know these weights exactly. If the weights are different for different a_n, then it is possible that $\bar{r}^n(a'_n) > \bar{r}^n(a^*_n)$ merely because other agents are using their good arms when agent n is selecting a suboptimal arm a_n while other agents are using their bad arms when agent n is selecting the optimal arm a^*_n. Hence, simply relying on the relative reward estimates does not guarantee obtaining the correct information needed to find the optimal arm.

Reward functions that are fully informative exist for many applications. We identify a class of overall reward functions that are fully informative below.

Fully Informative Reward Functions: For each agent n, if there exists a function $f_n : \mathcal{A}_n \to \mathbb{R}$, such that for all joint arms $\boldsymbol{a}$, the expected reward can be expressed as a function $F : \mathbb{R}^N \to \mathbb{R}$ where $\mu(\boldsymbol{a}) = F(f_1(a_1), ..., f_N(a_N))$ and μ is monotone in $f_n, \forall f_{-n}, \forall n$, then $\mu(\boldsymbol{a})$ is fully informative.

We provide two concrete examples below.

(1) *Classification with multiple classifiers.* In the problem of event classification using multiple classifiers, each classifier is in charge of the classification problem of one specific feature of the target event [FTV07] [FS10] [FV09] [DTS10]. The event is accurately classified if and only if all classifiers have their corresponding features classified correctly. Let $f_n(a_n)$ be the unknown feature classification accuracy of classifier n by selecting the operating point a_n. Assuming that the features are independent, then the event classification accuracy can be expressed as $\mu(\boldsymbol{a}) = \prod_{n=1}^{N} f(a_n)$ given the selection of the joint operating points $\boldsymbol{a}$ of all

classifiers. Hence the event classification accuracy is fully informative.

(2) *Network security monitoring using distributed learning*: A set of distributed learners (each in charge of monitoring a specific sub-network) make predictions about a potential security attack based on their own observed data (e.g. packets from different IP addresses to their corresponding sub-networks). Let the prediction of learner n be $\tilde{y}_n(t|a_n(t)) \in \{1, -1\}$ at time t by choosing a classification function a_n. Based on these predictions, an ensemble learner uses a weighted majority vote rule [LW94] to make the final prediction, i.e. $\hat{y}(t|\boldsymbol{a}(t)) = \text{sgn}(\boldsymbol{w} \cdot \tilde{\boldsymbol{y}}(t|a_n(t)))$, and takes the security measures accordingly. In the end, the distributed learners observe the outcome $r_n^t(\boldsymbol{a}(t))$ of the system which depends on the accuracy of the prediction, i.e. $|y(t) - \hat{y}(t|\boldsymbol{a}(t))|$ with $y(t)$ being the true security condition. Let $f_n(a_n) = \mathbb{E}\{|\tilde{y}_n(a_n) - \hat{y}|\}$ be the accuracy of learner n by choosing a classification function a_n. The reward function $\mu(\boldsymbol{a})$ is also monotone in $f_n(a_n)$ and hence is fully informative.

We note that in the first example different agents have orthogonal learning tasks (classification with respect to different features) while in the second example different agents have the same learning task (detecting the security attack). However, both examples exhibit the fully informative property and our proposed learning algorithms handle both cases effectively. The difference comes from the *speed of learning*. When agents have orthogonal learning tasks, they are more pivotal and so their actions have a greater influence on the rewards, which allows them to learn faster as well. This is highlighted in our simulation results in Section IX, where it is shown that when an agent becomes more pivotal it discovers its optimal action quicker.

4.5.2 Description of the Algorithm

In this subsection, we describe an improved learning algorithm. We call this new algorithm the DisCo-FI algorithm where "FI" stands for "Fully Informative" [5]. The key difference from the basic DisCo algorithm is that, in DisCo-FI, the agents will maintain relative reward estimates instead of the exact reward estimates.

[5]The algorithm can run in the general case, but we bound its regret only when the overall reward function is fully informative.

Knowledge, Counters and Estimates: Agents know a common deterministic function $\zeta(t)$ and maintain two counters $\gamma(t)$ and $E(t)$. Now each exploration phase has a fixed length of $L_2 = \sum_{n=1}^{N} K_n$ slots and hence, $E(t) \in \{0, 1, ..., L_2\}$ with $E(t) = 0$ representing that the slot is an exploitation slot and $E(t) > 0$ representing that it is the $E(t)$-th relative slot in the current exploration phase. As before, both counters are initialized to be $\gamma(0) = 0, E(0) = 0$. Each agent n maintains K_n sample mean (relative) reward estimates $\bar{r}^n(a_n), \forall a_n \in \mathcal{A}_n$, one for each one of its own arms. These (relative) reward estimates are initialized to be $\bar{r}^n(a_n) = 0$ and will be updated over time using the realized rewards.

Phase Transition: The transition between exploration phases and exploitation phases are almost identical to that in the DisCo algorithm. The only difference is that at the end of each exploration slot, the counter $E(t+1)$ for the next slot is updated to be $E(t+1) \leftarrow mod(E(t)+1, L_2+1)$. Hence, we ensure that each exploration phase has only L_2 slots.

Prescribed Actions: The algorithm prescribes different actions for agents in different slots and in different phases.

(i) Exploration phase: As clear from the Phase Transition, an exploration phase consists of L_2 slots. These slots are further divided into N subphases and the length of the n^{th} subphase is K_n. In the n^{th} subphase, agents take actions as follows (Figure 3 provides an illustration):

1. Agent n selects each of its arms $a_n \in \mathcal{A}_n$ in turn, each arm for one slot. At the end of each slot in this subphase, it updates its reward estimate using the realized reward in this slot as follows,
$$\bar{r}^n(a_n) \leftarrow \frac{\gamma(t)\bar{r}^n(a_n) + r_t^n}{\gamma(t)+1} \tag{4.6}$$
2. Agent $i \neq n$ selects the arm with the highest reward estimate for every slot in this subphase, i.e. $a_i(t) = \arg\max_{a_i \in \mathcal{A}_i} \bar{r}^i(a_i)$.

(ii) Exploitation phase: Each exploitation phase has a variable length which depends on the control function $\zeta(t)$ and counter $\gamma(t)$. In each exploitation slot t, each agent n selects $a_n(t) = \arg\max_{a \in \mathcal{A}_n} \bar{r}^n(a)$.

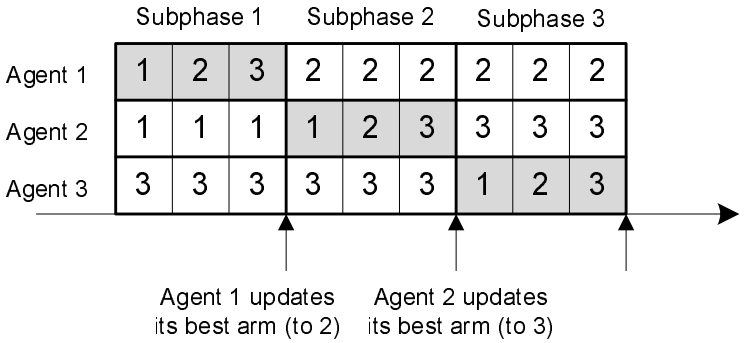


Figure 4.3: Illustration of one exploration phase with 3 agents, each of which having 3 arms.

4.5.3 Analysis of regret

We bound the regret of the DisCo-FI algorithm in Theorem 3. Let $\Delta_n^{min} = \min_{a_n \neq a_n^*, \boldsymbol{a}_{-n}} \{\mu(a_n^*, \boldsymbol{a}_{-n}) - \mu(a_n, \boldsymbol{a}_{-n})\}$ be the reward difference of agent n's best arm and its second-best arm, and let $\Delta_{FI}^{min} = \min_n \Delta_n^{min}$.

Theorem 4. *Suppose $\mu(\boldsymbol{a})$ is fully informative. If $\zeta(t) = A \ln T$ with $A \geq 2\left(\frac{D}{\Delta_{FI}^{min}}\right)^2$, then the expected regret of the DisCo-FI algorithm after any number T slots is bounded by*

$$R(T) < AL_2\Delta^{max} \ln T + B_2 \tag{4.7}$$

where $B_2 = L_2\Delta^{max} + 2L_2\Delta^{max} \sum_{t=1}^{\infty} t^{-\frac{A}{2}\left(\frac{\Delta_{FI}^{min}}{D}\right)^2}$ is a constant number.

Proof. See Appendix. □

The regret bound proved in Theorem 2 is also logarithmic in time for any finite time horizon T. Therefore, the average reward is guaranteed to converge to the optimal reward when the time horizon goes to infinity, i.e. $\lim_{T\to\infty} \mathbb{E}[R(T)]/T = 0$. Importantly, the proposed DisCo-FI algorithm exploits the informativeness of the expected overall reward function and achieves a much smaller constant that multiplies $\ln T$. Instead of learning every joint arm, agents can directly learn their own optimal arm through the relative reward estimates.

4.6 A Learning Algorithm For Partially Informative Rewards

In the previous section, we developed the DisCo-FI algorithm for reward functions that are fully informative. However, in problems where the full informativeness property may not hold, the DisCo-FI algorithm cannot guarantee a logarithmic regret bound. In this section, we extend DisCo-FI to the more general case where the full informativeness constraint is relaxed. For example, in the classification problem which uses multiple classifiers, each classifier consists of multiple components each of which is considered as an independent agent. The accuracy of each individual classifier may depend on the configurations of these components in a complex way but the overall classification accuracy is still increasing in the accuracy of each individual classifier. Specifically, if the accuracy of one of these classifiers is increased, then the overall accuracy will increase independently of which configuration of the components of that classifier are chosen.

4.6.1 Partially Informativeness

We define an *agent group* and a *group partition* first.

Definition 4. *(Agent Group and Group partition) An agent group g consists of a set of agents. A group partition $\mathcal{G} = \{g_1, ..., g_M\}$ of size M is a set of M agent groups such that each agent $n \in \mathcal{N}$ belongs to exactly one group.*

We will call the set of arms selected by the agents in a group g_m a *group-joint arm* with respect to group g_m, denoted by $\boldsymbol{a}_m = \{a_n\}_{n \in g_m}$[6]. Denote the size of group g_m by N_m. It is clear that $\sum_{m=1}^{M} N_m = N$.

Definition 5. *(Group-Informativeness) An expected overall reward function $\mu(\boldsymbol{a})$ is said to be informative with respect to a group g_m if there exists a unique group-joint arm $\boldsymbol{a}_m^* \in \times_{n \in g_m} \mathcal{A}_n$ such that $\forall \boldsymbol{a}_{-m}, \boldsymbol{a}_m^* = \arg \max_{\boldsymbol{a}_m \in \times_{i \in g_m} \mathcal{A}_i} \mu(\boldsymbol{a}_m, \boldsymbol{a}_{-m})$.*

In words, for different choices of arms by other agents, group g_m's best group-joint arm is

[6]We abuse notation by using $\boldsymbol{a}_m$ $\boldsymbol{a}_{g_m}$. This should not introduce confusion given specific contexts.

the same. Note that this is a generalization of Definition 1 because a group can also consist of only a single agent. Lemma 2 immediately follows.

Lemma 3. *If $\mu(\boldsymbol{a})$ is informative with respect to an agent group g_m and the unique group-joint optimal arm is $\boldsymbol{a}_m^*$, then the following is true:*

$$\begin{aligned} \boldsymbol{a}_m^* = \arg\max \textstyle\sum_{\boldsymbol{a}_{-m}} \theta_{\boldsymbol{a}_{-m}} \mu(\boldsymbol{a}_m, \boldsymbol{a}_{-m}), \\ \forall \theta_{\boldsymbol{a}_{-m}} \geq 0, \sum_{\boldsymbol{a}_{-m}} \theta_{\boldsymbol{a}_{-m}} = 1 \end{aligned} \tag{4.8}$$

Proof. This is a direct result of Definition 4. □

Lemma 2 states that for an agent group g_m, the weighted average of the expected reward over all possible choices of arms by other agents is maximized at the optimal group-joint arm $\boldsymbol{a}_m$. Moreover, the optimal group-joint arm is the same for all possible weights. Therefore, to evaluate the optimality of a group-joint arm $\boldsymbol{a}_m$, the agents in group g_m ($\forall n \in g_m$) can use the relative reward estimate for that group-joint arm $\bar{r}^n(\boldsymbol{a}_m)$ instead of using the exact expected reward estimate $\bar{r}^n(\boldsymbol{a}_m, \boldsymbol{a}_{-m})$ as long as the weights $\theta_{\boldsymbol{a}_{-m}}, \forall \boldsymbol{a}_{-m}$, are the same for all $\boldsymbol{a}_m$.

Definition 6. *(Partially Informative) An expected overall reward function $\mu(a)$ is said to be partially informative with respect to a group partition $\mathcal{G} = \{g_1, ..., g_M\}$ if it is informative with respect to all groups in $\mathcal{G}$.*

Consider a surveillance problem in a wireless sensor network. Assume that there are multiple areas that are monitored by clusters of sensors. Let m be the m-th cluster of sensors. Each sensor selects a surveillance action. For instance, this action can be the position of the video camera, channel listened to by the sensor, etc. Let $\mu_m(\boldsymbol{a}_m)$ be the reward of the joint surveillance action taken by the sensors in cluster m. For example, this reward can be the probability of detecting an intruder that enters the area surveyed by the sensors in cluster m, Then depending on the strategic importance of these areas, the global reward is a linear combination of the rewards of the clusters, i.e., $\mu(\boldsymbol{a}) = \sum_m w_m \mu_w(\boldsymbol{a}_m)$. However, improving each individual sensor's action may not necessarily improve the accuracy of the cluster. In

this case, the global reward is monotone in each cluster's reward but may not be monotone in each individual sensor's action. Thus, the reward function is partially informative.

If a reward function is fully informative, then it is also partially informative with respect to any group partition of the agents. On the other hand, if we take the entire agent set as one single group, then any reward function is partially informative with respect to this partition. Therefore, "Partially Informative" can apply to all possible reward functions through defining the group partition appropriately.

4.6.2 Description of the Algorithm

In this subsection, we propose the improved algorithm whose regret can be bounded for reward functions that are partially informative. We call this new algorithm the DisCo-PI algorithm where "PI" stands for "Partially Informative".

Knowledge, Counters and Estimates: Agents know a common deterministic function $\zeta(t)$ and maintain two counters $\gamma(t)$ and $E(t)$. In the DisCo-PI algorithm, each exploration phase has a fixed length of $L_3 = \sum_{m=1}^{M} \prod_{n \in g_m} K_n$ slots and hence, $E(t) \in \{0, 1, ..., L_3\}$ with $E(t) = 0$ representing that the slot is not an exploration slot and $E(t) > 0$ representing that it is the $E(t)$-th relative slot in the current exploration phase. Both counters are initialized to be $\gamma(0) = 0, E(0) = 0$. Each agent n in group g_m maintains $S_m = \prod_{i \in g_m} K_i$ reward estimates $\bar{r}^n(l), \forall l \in \{1, 2, ..., S_m\}$. Let $b_l^n \in \mathcal{A}_n$ denote the arm selected by agent n in the l^{th} slot in an exploration subphase. These (relative) reward estimates are initialized to be $\bar{r}^n(l) = 0$ and will be updated over time using the realized rewards.

Phase Transition: The algorithm works in a similar way as the first two algorithms in determining whether a slot is an exploration slot or an exploitation slot. The only difference is that the counter $E(t+1)$ is updated to be $E(t+1) \leftarrow mod(E(t)+1, L_3+1)$. This ensures that each exploration phase has L_3 slots.

Prescribed Actions: The algorithm prescribes different actions in different slots and in different phases.

(i) Exploration phase: An exploration phase consists of L_3 slots. These slots are further

divided into M subphases and the length of the m^{th} subphase is $\prod_{n\in g_m} K_n$. In the m^{th} subphase, agents take actions as follows:

1. Agents in group g_m select the arms in such a way that every group-joint arm $\boldsymbol{a}_m$ with respect to group g_m is selected exactly once in this exploration subphase. At the end of the l^{th} slot in the exploration subphase, $\bar{r}^n(b_l^n)$ is updated to be

$$\bar{r}^n(l) \leftarrow \frac{\gamma(t)\bar{r}^n(l) + r_t^n}{\gamma(t)+1} \tag{4.9}$$

2. Agents i in group $g_j \neq g_m$ selects the component arm that forms the group-joint arm with the highest reward estimate, i.e. $a_i = \{b_{l^*}^i : l^* = \arg\max_l \bar{r}^i(l)\}$, for every slot in this subphase.

(ii) Exploitation phase: Each exploitation phase has a variable length which depends on the control function $\zeta(t)$ and counter $\gamma(t)$. In each exploitation slot t, each agent n of group g_m selects $a_n = \{b_{l^*}^n : l^* = \arg\max_l \bar{r}^n(l)\}$.

4.6.3 Analysis of regret

We bound the regret by running the DisCo-PI algorithm in Theorem 4. Let $\Delta_m^{min} = \min_{\boldsymbol{a}_m \neq \boldsymbol{a}_m, \boldsymbol{a}_{-m}} \{\mu(\boldsymbol{a}_m^*, \boldsymbol{a}_{-m}) - \mu(\boldsymbol{a}_m, \boldsymbol{a}_{-m})\}$ be the reward difference of the best group-joint arm of g_m and the second-best group-joint arm of g_m, and let $\Delta_{PI}^{min} = \min_m \Delta_m^{min}$.

Theorem 5. *Suppose $\mu(\boldsymbol{a})$ is partially informative with respect to a group partition $\mathcal{G}$. If $\zeta(t) = A\ln T$ with $A \geq 2\left(\frac{D}{\Delta_{PI}^{min}}\right)^2$, then the expected regret of the DisCo-PI algorithm after any number T slots is bounded by*

$$R(T) < AL_3\Delta^{max}\ln T + B_3 \tag{4.10}$$

where

$$B_3 = L_3\Delta^{max} + 2\sum_{m=1}^{M} N_m \prod_{n\in g_m} K_n \Delta^{max} \sum_{t=1}^{\infty} t^{-\frac{A}{2}\left(\frac{\Delta_{PI}^{min}}{D}\right)^2} \tag{4.11}$$

is a constant number.

	DisCo	**DisCo-PI**	**DisCo-FI**
Reward Informativeness	Any	Partially Informative	Fully Informative
Learning Speed	Slow	Medium	Fast
Regret order	$O(\prod_{n=1}^{N} K_n \ln T)$	$O(\sum_{m=1}^{M} \prod_{n \in g_m} K_n \ln T)$	$O(\sum_{n=1}^{N} K_n \ln T)$

Table 4.1: Comparison of the proposed three algorithms.

Proof. See Appendix. □

The regret bound proved in Theorem 4 is also logarithmic in time for any finite time horizon T. Therefore, the average reward is guaranteed to converge to the optimal reward when the time horizon goes to infinity, i.e. $\lim_{T\to\infty} \mathbb{E}[R(T)]/T = 0$. However, instead of learning every joint arm like in DisCo, agents in each group can learn just their own optimal group-joint arm using the relative reward estimates. Note that the constant that multiplies $\ln T$ is smaller than that of DisCo but larger than DisCo-FI. Table II summarizes the characteristics of the three proposed algorithms.

4.7 Illustrative Results

In this section, we illustrate the performance of the proposed learning algorithms via simulation results for the Big Data mining problem using multiple classifiers.

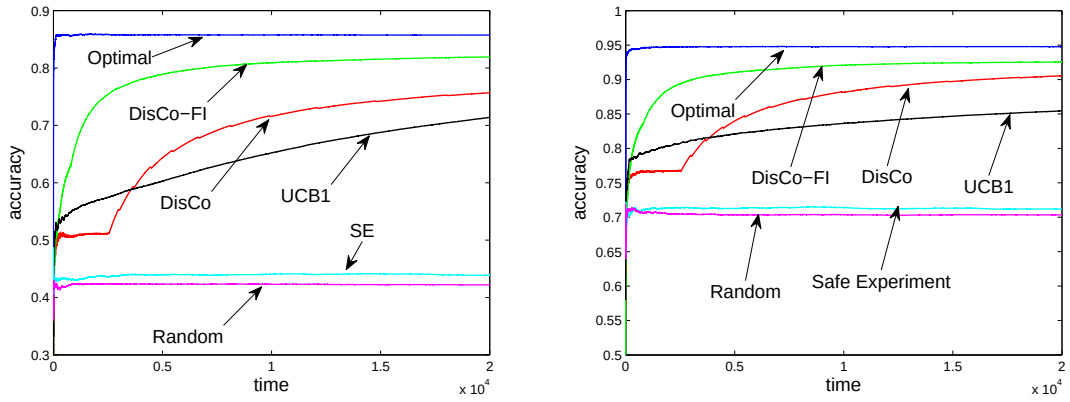
4.7.1 Big Data Mining using Multiple Classifiers

A plethora of online Big Data applications, such as video surveillance, traffic monitoring in a city, network security monitoring, social media analysis etc., require processing and analyzing streams of raw data to extract valuable information in real-time [SHC03]. A key research challenge [DS13] in a real-time stream mining system is that the data may be gathered online by multiple distributed sources and subsequently it is locally processed and classified to extract knowledge and actionable intelligence, and then sent to a centralized entity which

is in charge of making global decisions or predictions. The various local classifiers are not collocated and cannot communicate with each other due to the lack of a communication infrastructure (because of delays or other costs such as complexity [FS10] [DTS10]). Another stream mining problem may involve the processing of the same or multiple data stream, but require the use of classifier chains (rather than multiple single classifiers which are distributed as mentioned before) for its processing. For instance, video event detection [JBC13] [SMK13] requires finding events of interest or abnormalities which could involve determining the concurrent occurrence (i.e. classification) of a set of basic objects and features (e.g. motion trajectories) by chaining together multiple classifiers which can jointly determine the presence of the event or phenomena of interest. The classifiers are often implemented at various locations to ensure scalability, reliability and low complexity [FTV07] [FS10]. For all incoming data, each classifier needs to select an operating point from its own set, whose accuracy and cost (e.g. delay) are unknown and may depend on the incoming data characteristics, in order to classify its corresponding feature and maximize the event classification accuracy (i.e. the overall system reward). Hence, classifiers need to learn from past data instances and the event classification performance to construct the optimal chain of classifiers. This classifier chain learning problem can be directly mapped into the considered multi-agent decision making and learning problem: agents are the component classifiers, actions are the operating points and the overall system reward is the event classification performance (i.e. accuracy minus cost).

4.7.2 Experiment Setup

By extracting features such as color histogram, color correlogram, and co-occurrence texture, the classifiers are trained to detect high-level features, such as whether the video shot takes place outdoors or in an office building, or whether there is an animal or a car in the video. In the simulations, we use three classifiers (agents) to classify three features. By synthesizing the feature classification results, the event detection result is obtained under two different rules.



(a) Rule 1 (b) Rule 2

Figure 4.4: Performance comparison for various algorithms.

1. **Rule 1**: The event is correctly classified if all three features are correctly classified.

2. **Rule 2**: The event is correctly classified if Feature 1 (CAR) is correctly classified and either Feature 2 (MOU) or Feature 3 (SPO) is correctly classified.

In the simulations, each classifier can choose from 4 operating points which will result in different accuracies. Let p_n denote the classification accuracy with respect to feature n. Assume that the classification of features is independent among classifiers, then the event classification accuracy p_{event} depends on the feature classification accuracy as follows:

$$\begin{aligned} p_{event} &= p_1 p_2 p_3 && \text{under Rule 1} \\ p_{event} &= p_1(1-(1-p_2)(1-p_3)) && \text{under Rule 2} \end{aligned} \tag{4.12}$$

Hence, the reward structure is fully informative for both event synthesis rules.

4.7.3 Performance Comparison

We implement the proposed algorithms and compare their performance against four benchmark schemes:

(1) *Random*: In each period, each classifier randomly selects one operating point.

(2) *Safe Experimentation (SE)*: This is a method used in [FV09] when there is no uncertainty about the accuracy of the classifiers. In each period t, each classifier selects its

baseline action with probability $1 - \epsilon_t$ or selects a new random action with probability ϵ_t. When the realized reward is higher than the baseline reward, the classifiers update their baseline actions to the new action.

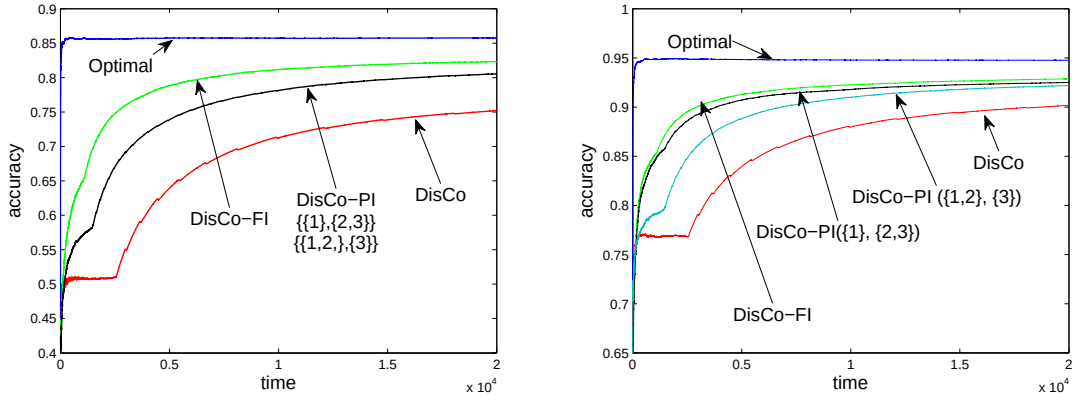
(3) *UCB1*: This is a classic multi-armed bandit algorithm proposed in [ACF02]. As we showed in Proposition 1, there may be problems implementing this centralized algorithm in a distributed setting without message exchange. Nevertheless, for the sake of our simulations we will assume that there are no individual errors in the observation of the global feedback when we implement UCB1, and hence it can be perfectly implemented in our distributed environment.

(4) *Optimal*: In this benchmark, the classifiers choose the optimal joint operating points (trained offline) in all periods.

Figure 4 shows the achieved event classification accuracy over time under both rule 1 and rule 2. All curves are obtained by averaging 50 simulation runs. We also note that agents may receive noisy versions of the outcome (except for UCB1). Under both rules, SE works almost as poorly as the Random benchmark in terms of event detection accuracy. Due to the uncertainty in the detection results, updating the baseline action to a new action with a higher realized reward does not necessarily lead to selecting a better baseline action. Hence, SE is not able to learn the optimal operating points of the classifiers. UCB1 achieves a much higher accuracy than Random and SE algorithms and is able to learn the optimal joint operating points over time. However, the learning speed is slow because the joint arm space is large, i.e. $4^3 = 64$. The proposed DisCo algorithm can also learn the optimal joint action. However, since the joint arm space is large, the classifiers have to stay in the exploration phases for a relatively long time in the initial periods to gain sufficiently high confidence in reward estimates while the exploitation phases are rare and short. Thus, the classification accuracy is low initially. After the initial exploration phases, the classifiers begin to exploit and hence the average accuracy increases rapidly. Since the reward structure satisfies the Fully Informative condition, DisCo-FI rapidly learns the optimal joint action and performs the best among all schemes. Table 4.2 shows the false alarm and miss detection rates under rule 1 by treating one event as the null hypothesis and the remaining events as the alternative

Table 4.2: False Alarm and Miss Detection Rates.

	DisCo	DisCo-FI	UCB1	SE	Random
False Alarm	0.039	0.029	0.050	0.065	0.069
Miss Detection	0.249	0.194	0.356	0.469	0.496



(a) Rule 1 (b) Rule 2

Figure 4.5: Performance comparison for DisCo, DisCo-FI and DisCo-PI.

hypothesis.

4.7.4 Informativeness

Next, we compare the learning performance of the three proposed algorithms. For the DisCo-PI algorithm, we consider two group partitions $\{\{1\},\{2,3\}\}$ and $\{\{1,2\},\{3\}\}$. Figure 5 shows the learning performance over time for DisCo, DisCo-FI and DisCo-PI under Rule 1 and Rule 2. In both cases, DisCo-FI achieves the smallest learning regret and hence the fastest learning speed while the basic DisCo algorithm performs the worst. This is because DisCo-FI fully exploits the problem structure. The performance of the DisCo-PI algorithm is in between that of DisCo-FI and the basic DisCo algorithm. However, different group partitions have different impacts on the performance. Under Rule 1, the two group partitions perform similarly since the impacts of the three classifiers on the final classification result are symmetric. Under Rule 2, the impacts of classifier 2 and classifier 3 are coupled in a more

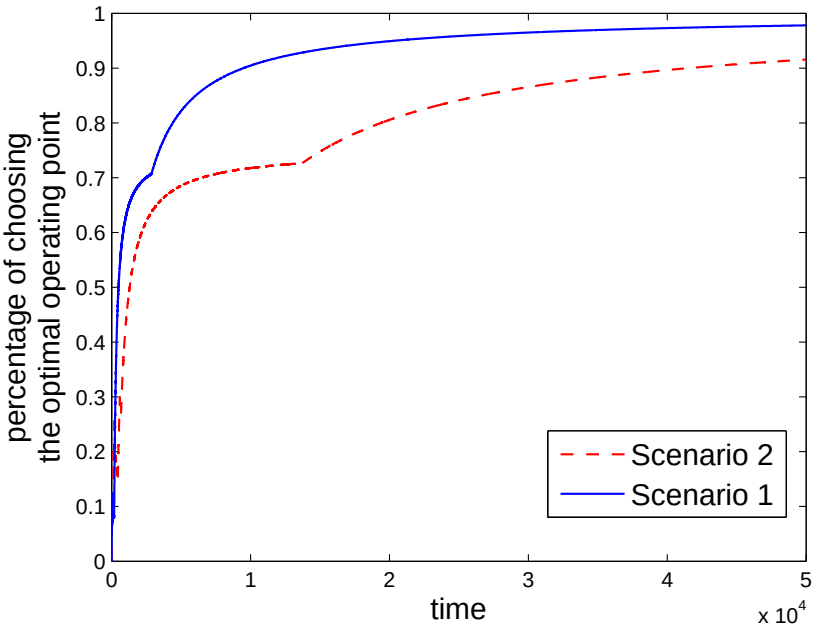
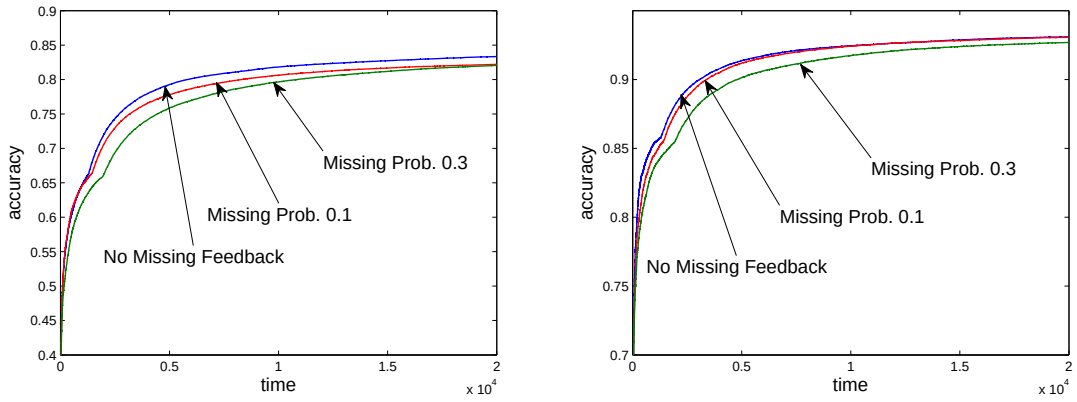


Figure 4.6: Classifier 2 learns its optimal operating point at different speeds under different rules.

complex way. Since the group partition $\{\{1\},\{2,3\}\}$ captures this coupling effect better, it performs better than the group partition $\{\{1,2\},\{3\}\}$. We note that even though that in this simulation DisCo-FI performs the best, in other scenarios where the reward function is only partially informative or even not informative, DisCo-PI and DisCo may perform better.

4.7.5 Impacts of reward function on learning speed

For both synthesis rules, the reward functions are fully informative, and so classifiers can learn their own optimal operating points using only the relative rewards. However, the same classifier will learn its optimal operating point at different speeds under different rules due to the differences in that classifier's impact on the global reward. Note that in the first rule, all the classifiers are processing different tasks of equal importance, whereas in the second rule classifiers 2 and 3 are less critical than classifier 1. Thus the learning speed for classifier 2 will be slower under the second rule because its impact is lower. This learning speed depends on the overall reward difference between the classifier's best operating point and its second-



(a) Rule 1 (b) Rule 2

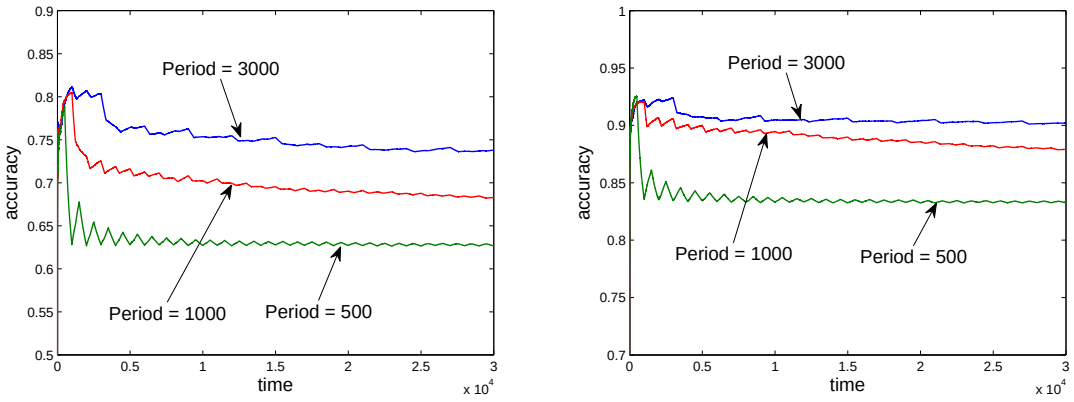
Figure 4.7: Learning performance in scenarios with missing feedback.

best operating point, i.e. Δ_n^{min} . For classifier 2: under Rule 1, $\Delta_2^{min} = \min p_1 \min p_3 \Delta p_2$ and under Rule 2, $\Delta_2^{min} = \min p_1 \min(1 - p_3)\Delta p_2$ with Δp_2 being the accuracy difference of classifier 2's best and second-best operating points. Since p_n is usually much larger than 0.5, Δ_2^{min} of Rule 2 is much smaller than that of Rule 1 and hence, classifier 2 learns its optimal operating point at a much slower speed under Rule 2 than Rule 1. Figure 6 illustrates the percentage of choosing the optimal operating point by classifier 2 under different rules.

4.7.6 Missing and Delayed Feedback

In this set of simulations, we study the impact of missing and delayed global feedback on the learning performance of the proposed algorithm. In Figure 7, we show the accumulating accuracy of the modified DisCo-FI algorithm for three missing feedback scenarios – there is no missing feedback, the missing probability is 0.1 and 0.3. A larger missing probability induces lower classification accuracy for a given time. Nevertheless, the proposed algorithm is not very sensitive to missing feedbacks. Even if the missing probability is relatively large, the degradation of the learning performance is small.

In Figure 8, we show the accumulating accuracy of the modified DisCo-FI algorithm for three delayed feedback scenarios – there is no delay, the maximal delay is 50 slots and 100 slots. Under both synthesis rules, learning is the fastest without feedback delays, and the



(a) Rule 1

(b) Rule 2

Figure 4.8: Learning performance in scenarios with accuracy drift.

larger the delay, the slower the learning speed. However, even with delays, the proposed DisCo-FI algorithm is still able to achieve logarithmic regret.

4.8 Conclusions

In this chapter, we studied a general multi-agent decision making problem in which decentralized agents learn their best actions to maximize the system reward using only noisy observations of the overall reward. The challenging part is that individualized feedback is missing, communication among agents is impossible and the global feedback is subject to individual observation errors. We proposed a class of distributed cooperative learning algorithms that addresses all these problems. These algorithms were proved to be able to achieve logarithmic regret in time. We also proved that by exploiting the informativeness of the reward function, much better regret results can be achieved by our algorithms compared with existing solutions. Through simulations we applied the proposed learning algorithms to Big Data stream mining problems and showed significant performance improvements. Importantly, our theoretical framework can also be applied to learning in other types of multi-agent systems where communication between agents is not possible and agents observe only noisy global feedback.

4.9 Appendix

Proof of Theorem 1

In order to prove $\min_{\pi \in \Pi^c} R(T;\pi) = \min_{\pi \in \Pi^d} R(T;\pi)$, we will show that (i) there exists an optimal algorithm π^* that is deterministic and (ii) any deterministic algorithm can be implemented in the distributed scenario if agents have identical observations of the overall reward realization. If (i) and (ii) are true, then the optimal algorithm π^* belongs to Π^d and therefore $\min_{\pi \in \Pi^c} R(T;\pi) = \min_{\pi \in \Pi^d} R(T;\pi)$. We prove the claims in the following.

(i) Suppose at time $t \leq T$, the reward history is $\mathcal{H}_t$. Let $\text{Reward}(T;\mathcal{H}_t)$ be the expected sum of rewards from time t to T given the history $\mathcal{H}_t$. For an optimal algorithm π^*, the following must be satisfied

$$\text{Reward}(T;\mathcal{H}_t) = \max_{\pi^*(\mathcal{H}_t)} \mathbb{E}[r(\pi^*(\mathcal{H}_t) + \text{Reward}(T;\{\mathcal{H}_t, r(\pi^*(\mathcal{H}_t))\})] \tag{4.13}$$

If $\pi^*(\mathcal{H}_t)$ is a mixed strategy, then it implies that there exist at least two pure joint arms $\boldsymbol{a}'$ and $\boldsymbol{a}''$ such that the algorithm is indifferent between these two joint arms in terms of maximizing the expected sum reward, i.e.

$$\text{Reward}(T;\mathcal{H}_t) = \text{Reward}(T;\mathcal{H}_t,\boldsymbol{a}') = \text{Reward}(T;\mathcal{H}_t,\boldsymbol{a}'') \tag{4.14}$$

Therefore, by setting $\pi^*(\mathcal{H}_t)$ to be any one of the pure joint arms, the algorithm does not lose any expected reward. Since this argument holds for any time t and any history $\mathcal{H}_t$, there must exist a deterministic algorithm that is optimal.

(ii) If an algorithm is deterministic and the overall reward realization can be perfectly observed by all agents, then the reward history is public and identical for all agents. Each agents algorithm depends only on its observed history, and so each agent can correctly infer from the public reward history the arms to be selected by all other agents. This implies that there is no need for message exchange among agents at run-time, because each agent will know the exact arm chosen by any other agent at every time t. Hence, any deterministic algorithm can be implemented in a distributed setting.

Proof of Proposition 1

We prove this by constructing an example showing that the regret achieved by UCB1 is linear. Consider a network with 2 agents and each agent having two arms to select from. The expected overall reward of the 4 joint arms is listed in Table 1.

To simplify the analysis, we will assume that the realization of the overall reward at each time is exactly the expected overall reward at all times. However, agents may observe different noisy versions of the realization. We consider a special case where agent 1 perfectly observes the reward realization without any error in all slots, while agent 2 observes the reward realization with errors in the first 4 slots but without errors in the remaining slots. The error is drawn uniformly from the space $\{-2, 0, 2\}$ and independent across time. In the UCB1 algorithm, agents start by selecting each of the 4 joint arms once and updating their reward estimates. Consider an error sequence $\{-2, 0, 0, 2\}$ in the first 4 slots. Given this error sequence, at the beginning of slot 5, the reward estimates are (the superscript indicates the agent): for agent 1, $\bar{r}^1(\{1,1\}) = 10$, $\bar{r}^1(\{1,2\}) = 0$, $\bar{r}^1(\{2,1\}) = 0$, $\bar{r}^1(\{2,2\}) = 8$; for agent 2, $\bar{r}^1(\{1,1\}) = 8$, $\bar{r}^1(\{1,2\}) = 0$, $\bar{r}^1(\{2,1\}) = 0$, $\bar{r}^1(\{2,2\}) = 10$. According to UCB1, both agents calculate the indices for all joint arms which are: for agent 1, $g^1(\{1,1\}) = 10 + \sqrt{2\ln 5}$,$g^1(\{1,2\}) = 0 + \sqrt{2\ln 5}$, $g^1(\{2,1\}) = 0 + \sqrt{2\ln 5}$, $g^1(\{2,2\}) = 8 + \sqrt{2\ln 5}$; for agent 2, $g^1(\{1,1\}) = 8 + \sqrt{2\ln 5}$, $g^1(\{1,2\}) = 0 + \sqrt{2\ln 5}$, $g^1(\{2,1\}) = 0 + \sqrt{2\ln 5}$, $g^1(\{2,2\}) = 10 + \sqrt{2\ln 5}$. Therefore, agent 1 selects arm 1 and believes that agent 2 will also select arm 1 while agent 2 selects arm 2 and believes that agent 1 will also select arm 2. Since the actual selected joint arm is $\{1,2\}$, the realized reward in slot 5 is 0. At this point, agent 1 updates the reward estimate for the joint arm $\{1,1\}$ and agent 2 updates the reward estimate for the joint arm $\{2,2\}$. Hence, $\bar{r}^1(\{1,1\}) = 5$ and $\bar{r}^2(\{2,2\}) = 5$. We note that the reward estimates and indices are still "symmetric" in the sense $\bar{r}^1(\{1,1\}) = \bar{r}^2(\{2,2\})$, $\bar{r}^1(\{2,2\}) = \bar{r}^2(\{1,1\})$, $g^1(\{1,1\}) = g^2(\{2,2\})$ and $g^1(\{2,2\}) = g^2(\{1,1\})$. It can be easily shown that such "symmetry" will persist for all remaining periods. Suppose at some time agent 1 believes that it needs to select $\{1,2\}$ (or $\{2,1\}$), then agent 2 will also believe that it needs to select arm $\{1,2\}$ (or $\{2,1\}$). Both agents will update the rewards of $\{1,2\}$ (or $\{2,1\}$)

correctly. However, if agent 1 believes it needs to select arm $\{1,1\}$ (or $\{2,2\}$), then agent 2 will believe it needs to select arm $\{2,2\}$ (or $\{1,1\}$). Both agents will update the rewards wrongly but in the same way. Hence, in all time slots after the first 4 slots, the realized rewards are 0. Since the error sequence $\{-2,0,0,2\}$ occurs with a positive probability (2^{-4}), there is a constant gap from the optimal reward. Hence, the regret bound is linear by running such UCB1 algorithm when observing the reward realization is subject to private errors.

Proof of Theorem 2

It is clear that $\bar{r}^n(l) = \bar{r}^n(\boldsymbol{a}^l)$, $\forall n$ where a^l is the joint arm selected in the l^{th} relative index in the exploration phase. Since each joint arm is selected once in each exploration phase, it is equivalent to the case where agents maintain the reward estimates for all joint arms. Moveover, in the exploitation slot, agents select the component arms of the joint arm that maximizes the reward estimate.

First we prove that after P exploration phases, for each agent n, the probability that a non-optimal joint arm $\boldsymbol{a} \neq \boldsymbol{a}^*$ is selected in an exploitation slot is at most $e^{-\frac{P}{2\left(\frac{\Delta_{\boldsymbol{a}}}{D}\right)^2}}$ where $\Delta_{\boldsymbol{a}} = \mu(\boldsymbol{a}^*) - \mu(\boldsymbol{a})$. A non-optimal joint arm $\boldsymbol{a} \neq \boldsymbol{a}^*$ is selected by agent n in an exploitation slot only if $\bar{r}^n(\boldsymbol{a}) \geq \bar{r}^n(\boldsymbol{a}^*)$.

Since

$$P(\bar{r}^n(\boldsymbol{a}) < \bar{r}^n(\boldsymbol{a}^*)) > P(\bar{r}^n(\boldsymbol{a}^*) > \mu(\boldsymbol{a}^*) - 0.5\Delta_{\boldsymbol{a}}) \times P(\bar{r}^n(\boldsymbol{a}) < \mu(\boldsymbol{a}) + 0.5\Delta_{\boldsymbol{a}}) \tag{4.15}$$

we have,

$$\begin{aligned} &P(\bar{r}^n(\boldsymbol{a}) \geq \bar{r}^n(\boldsymbol{a}^*)) = 1 - P(\bar{r}^n(\boldsymbol{a}) < \bar{r}^n(\boldsymbol{a}^*)) \\ &< 1 - P(\bar{r}^n(\boldsymbol{a}^*) > \mu(\boldsymbol{a}^*) - 0.5\Delta_{\boldsymbol{a}}) \times P(\bar{r}^n(\boldsymbol{a}) < \mu(\boldsymbol{a}) + 0.5\Delta_{\boldsymbol{a}}) \\ &< P(\bar{r}^n(\boldsymbol{a}^*) \leq \mu(\boldsymbol{a}^*) - 0.5\Delta_{\boldsymbol{a}}) + P(\bar{r}^n(\boldsymbol{a}) \geq \mu(\boldsymbol{a}) + 0.5\Delta_{\boldsymbol{a}}) \end{aligned} \tag{4.16}$$

Since the reward is bounded by D and the reward estimate is obtained using P realizations, by Hoeffding's inequality,

$$P(\bar{r}^n(\boldsymbol{a}^*) \leq \mu(\boldsymbol{a}^*) - 0.5\Delta_{\boldsymbol{a}}) = P(\bar{r}^n(\boldsymbol{a}) \geq \mu(\boldsymbol{a}) + 0.5\Delta_{\boldsymbol{a}}) \leq e^{-\frac{P}{2}\left(\frac{\Delta^{min}}{D}\right)^2} \tag{4.17}$$

Therefore, $P(\bar{r}^n(\boldsymbol{a}) \geq \bar{r}^n(\boldsymbol{a}^*)) \leq 2e^{-\frac{P}{2}\left(\frac{\Delta^{min}}{D}\right)^2}$. Since there are L_1 sub-optimal joint arms, the probability that agent n selects any one of the sub-optimal joint arm is less than $2L_1e^{-\frac{P}{2}\left(\frac{\Delta^{min}}{D}\right)^2}$. Since there are N agents, the probability that there is at least one agent that selects the joint arm is less than $2L_1Ne^{-\frac{P}{2}\left(\frac{\Delta^{min}}{D}\right)^2}$.

Now we bound the regret of the DisCo algorithm. The regret consists of two parts $R(T) = R_1(T) + R_2(T)$ where $R_1(T)$ is the regret incurred in the exploration phases and $R_2(T)$ is the regret incurred in the exploitation phases up to slot T.

We bound $R_1(T)$ first. Since the algorithm starts with the exploration phase, it ensures that, at any time t, at most $\lceil \zeta(T) \rceil$ exploration phases have been gone through. In each exploration phase, the maximum regret is achieved when the agents select the worst joint arm in every slot and hence, the regret in one exploration phase is bounded by $L_1\Delta^{max}$. Therefore, $R_1(T)$ is at most

$$R_1(T) < \lceil \zeta(T) \rceil L_1\Delta^{max} \leq (\zeta(T)+1)L_1\Delta^{max} < AL_1\Delta^{max}\ln T + L_1\Delta^{max} \tag{4.18}$$

Next we bound $R_2(T)$. We know that, at any time slot $t < T$ when it is an exploitation slot, the probability that a non-optimal joint arm is selected is at most $2L_1Ne^{-\frac{\zeta(t)}{2}\left(\frac{\Delta^{min}}{D}\right)^2}$ since the algorithm ensures that at any exploitation slot at least $\lceil \zeta(t) \rceil$ exploration phases have been gone through. Therefore, the expected regret in any exploitation slot by selecting a non-optimal joint arm is at most

$$2L_1Ne^{-\frac{\zeta(t)}{2}\left(\frac{\Delta^{min}}{D}\right)^2}\Delta^{max} \leq 2L_1Ne^{-\frac{A\ln t}{2}\left(\frac{\Delta^{min}}{D}\right)^2}\Delta^{max} \tag{4.19}$$

Therefore, the expected regret $R_2(T)$ incurred in the exploitation phase is bounded by

$$R_2(T) \leq \sum_{t=0}^{\infty} 2NL_1\Delta^{max}t^{-\frac{A}{2}\left(\frac{\Delta^{min}}{D}\right)^2} \tag{4.20}$$

If we let $A > 2\left(\frac{D}{\Delta^{min}}\right)^2$, then $\sum_{t=0}^{\infty} t^{-\frac{A}{2}\left(\frac{\Delta^{min}}{D}\right)^2}$ is finite. Combining the bounds on $R_1(T)$ and $R_2(T)$ we get the result.

Proof of Theorem 3

First we prove that after P exploration phases, the probability that a non-optimal arm $a_n \neq a_n^*$ is selected by agent n is at most $2e^{-\frac{P}{2}\left(\frac{\Delta_n}{D}\right)^2}$. A non-optimal arm $a_n \neq a_n^*$ is selected only if $\bar{r}^n(a_n) > \bar{r}^n(a_n^*)$. The proposed algorithm ensures that in the n^{th} exploration subphase, agents $i \neq n$ are selecting the same arms while agent n is learning each of its own arms. Let $\boldsymbol{a}_{-n}(p)$ be the set of arms selected by other agents in the n^{th} subphase of the $p^{th}(p \leq P)$ exploration phase. Then the expectation of the reward estimate $\bar{r}^n(a_n)$ is

$$\mathbb{E}[\bar{r}^n(a_n)] = \frac{1}{P}\sum_{p=1}^{P} \mu(a_n, \boldsymbol{a}_{-n}(p)) \tag{4.21}$$

Since $\mu(a_n^*, \boldsymbol{a}_{-n}) - \mu(a_n, \boldsymbol{a}_{-n}) \geq \Delta_n^{min}$, $\forall \boldsymbol{a}_{-n}$ we also have

$$\mathbb{E}[\bar{r}^n(a_n^*)] - \mathbb{E}[\bar{r}^n(a_n)] \geq \tag{4.22}$$

Because the realized reward is bounded, according to Hoeffding's inequality, we have

$$P(\bar{r}^n(a_n) > \bar{r}^n(a_n^*)) \leq 2e^{-\frac{P}{2}\left(\frac{\Delta_n^{min}}{D}\right)^2} \tag{4.23}$$

Therefore, the probability that agent n selects a suboptimal arm is at most $2K_n e^{-\frac{P}{2}\left(\frac{\Delta_n^{min}}{D}\right)^2}$.

Now, we prove the regret bound of the learning algorithm which consists of two parts $R(T) = R_1(T) + R_2(T)$ where $R_1(T)$ is the regret incurred in the exploration phases and $R_2(T)$ is the regret incurred in the exploitation phases up to slot T.

First we bound $R_1(T)$. In each exploration phase, the maximum regret is achieved when the agents select the worst joint arm in every slot and hence, the regret in one exploration phase is bounded by $L_2\Delta^{max}$. Since the algorithm ensures that at any time T, at most $\lceil \zeta(T) \rceil$ exploration phases have been gone through, $R_1(T)$ is at most

$$R_1(T) < \lceil \zeta(T) \rceil L_2\Delta^{max} \leq (\zeta(T)+1)L_2\Delta^{max} = AL_2\Delta^{max}\ln T + L_2\Delta^{max} \tag{4.24}$$

Next we bound . At any time when it is an exploitation period, the expected regret by choosing a non-optimal arm is at most

$$2\sum_{n=1}^{N} K_n e^{-\frac{\zeta(T)}{2}\left(\frac{\Delta_n^{min}}{D}\right)^2}\Delta^{max} \leq 2L_2 e^{-\frac{A\ln T}{2}\left(\frac{\Delta_n^{min}}{D}\right)^2}\Delta^{max} \tag{4.25}$$

Hence, the expected regret in the exploitation slots up to time T is at most

$$R_2(T) = \sum_{t=1}^{T} \sum_{n=1}^{N} 2K_n \Delta^{max} t^{-\frac{A}{2}\left(\frac{\Delta_{FI}^{min}}{D}\right)^2} < 2L_2 \Delta^{max} \sum_{t=1}^{\infty} t^{-\frac{A}{2}\left(\frac{\Delta_{FI}^{min}}{D}\right)^2} \tag{4.26}$$

Because $A \geq 2\left(\frac{D}{\Delta_{FI}^{min}}\right)^2$, $\sum_{t=1}^{\infty} t^{-\frac{A}{2}\left(\frac{\Delta_{FI}^{min}}{D}\right)^2}$ is finite. Combining the bounds on $R_1(T)$ and $R_2(T)$ we get the result.

Proof of Theorem 4

Using the similar techniques in the proofs of Theorem 1 and Theorem 2, we can show that after P exploration phases, the probability that a non-optimal group-joint arm is selected by at least one agent in group g_m is at most $2N_m S_m e^{-\frac{P}{2}\left(\frac{\Delta_m^{min}}{}\right)^2}$. With this, we prove the regret bound of the DisCo-GO algorithm. The regret consists of two parts $R(T) = R_1(T) + R_2(T)$ where $R_1(T)$ is the regret incurred in the exploration phases and $R_2(T)$ is the regret incurred in the exploitation phases up to slot T.

First we bound $R_1(T)$. In each exploration phase, the maximum regret is achieved when the agents select the worst joint arm in every slot and hence, the regret in one exploration phase is bounded by $L_3\Delta^{max}$. Since the algorithm ensures that at any time T, at most $\lceil \zeta(T) \rceil$ exploration phases have been gone through, $R_1(T)$ is at most

$$R_1(T) < \lceil \zeta(T) \rceil L_3 \Delta^{max} \leq A L_3 \Delta^{max} \ln T + L_3 \Delta^{max} \tag{4.27}$$

Next we bound $R_2(T)$. At any time $t < T$ when it is an exploitation slot, the expected regret by choosing a non-optimal arm $\boldsymbol{a}_m \neq \boldsymbol{a}_m^*$ for agent group m is at most

$$2N_m S_m \Delta^{max} e^{-\frac{\zeta(T)}{2}\left(\frac{\Delta_m^{min}}{D}\right)^2} \leq 2N_m S_m \Delta^{max} t^{-\frac{A}{2}\left(\frac{\Delta_{PI}^{min}}{D}\right)^2} \tag{4.28}$$

Hence, the expected regret in the exploitation slots up to time is at most

$$R_2(T) = \sum_{t=1}^{T} \sum_{m=1}^{M} 2N_m \prod_{n \in g_m} K_n \Delta^{max} t^{-\frac{A}{2}\left(\frac{\Delta_{PI}^{min}}{D}\right)^2} \tag{4.29}$$

$$< 2 \sum_{m=1}^{M} N_m \prod_{n \in g_m} K_n \Delta^{max} \sum_{t=1}^{\infty} t^{-\frac{A}{2}\left(\frac{\Delta_{PI}^{min}}{D}\right)^2} \tag{4.30}$$

Because $A \geq 2\left(\frac{D}{\Delta_{PI}^{min}}\right)^2$, $\sum_{t=1}^{\infty} t^{-\frac{A}{2}\left(\frac{\Delta_{PI}^{min}}{D}\right)^2}$ is finite. Combining the bounds on $R_1(T)$ and $R_2(T)$ we get the result.

CHAPTER 5

Content Popularity Forecasting using Social Media

Networked services in the Web 2.0 era focus increasingly on the user participation in producing and interacting with rich media. The role of the Internet itself has evolved from the original use as a communication infrastructure, where users passively receive and consume media content to a social ecosystem, where users equipped with mobile devices constantly generate media data through a variety of sensors (cameras, GPS, accelerometers, etc.) and applications and, subsequently, share this acquired data through social media. Hence, social media is recently being used to provide situational awareness and inform predictions and decisions in a variety of application domains, ranging from live or on-demand event broadcasting, to security and surveillance [Tro12], to health communication [CHB09], to disaster management [SOM10], to economic forecasting [CV12]. In all these applications, forecasting the popularity of the content shared in a social network is vital due to a variety of reasons. For network and cloud service providers, accurate forecasting facilitates prompt and adequate reservation of computation, storage, and bandwidth resources [LWY12], thereby ensuring smooth and robust content delivery at low costs. For advertisers, accurate and timely popularity prediction provides a good revenue indicator, thereby enabling targeted ads to be composed for specific videos and viewer demographics. For content producers and contributors, attracting a high number of views is paramount for attracting potential revenue through micro-payment mechanisms.

While popularity prediction is a long-lasting research topic [SH10] [CKR07] [WTV10] [PAG13], understanding how social networks affect the popularity of the media content and using this understanding to make better forecasts poses significant new challenges. Conventional prediction tools have mostly relied on the history of the past view counts, which

worked well when the popularity solely depended on the inherent attractiveness of the content and the recipients were generally passive. In contrast, social media users are proactive in terms of the content they watch and are heavily influenced by their social media interactions; for instance, the recipient of a certain media content may further forward it or not, depending on not only its attractiveness, but also the situational and contextual conditions in which this content was generated and propagated through social media [LMW13]. For example, the latest measurement on Twitter's Vine, a highly popular short mobile video sharing service, has suggested that the popularity of a short video indeed depends less on the content itself, but more on the contributor's position in the social network [ZWL14]. Hence, being situation-aware, e.g. considering the content initiator's information and the friendship network of the sharers, can clearly improve the accuracy of the popularity forecasts. However, critical new questions need to be answered: which situational information extracted from social media should be used, how to deal with dynamically changing and evolving situational information, and how to use this information efficiently to improve the forecasts?

As social media becomes increasingly more ubiquitous and influential, the video propagation patterns and users' sharing behavior dynamically change and evolve as well. Offline prediction tools [SH10] [GAC10] [HDD11] [LH10] depend on specific training datasets, which may be biased or outdated, and hence may not accurately capture the real-world propagation patterns promoted by social media. Moreover, popularity forecasting is a *multi-stage* rather than a single-stage task since each video may be propagated through a cascaded social network for a relatively long time and thus, the forecast can be made at any time while the video is being propagated. A fast prediction has important economic and technological benefits; however, too early a prediction may lead to a low accuracy that is less useful or even damaging (e.g. investment in videos that will not actually become popular). The timeliness of the prediction has yet to be considered in existing works [CKR07]- [LH10] [WTV10] [PAG13] which solely focus on maximizing the accuracy. Hence, we strongly believe that developing a systematic methodology for accurate and timely popularity forecasting is essential.

In this chapter, we propose for the first time a systematic methodology and associat-

ed online algorithm for forecasting popularity of videos promoted by social media. Our Social-Forecast algorithm is able to make predictions about the popularity of videos while jointly considering the accuracy and the timeliness of the prediction. We explicitly consider the unique situational conditions that affect the video propagated in social media, and demonstrate how this *context information* can be incorporated to improve the accuracy of the forecasts. The unique features of Social-Forecast as well as our key contributions are summarized below:

- We rigorously formulate the online popularity prediction as a multi-stage sequential decision and online learning problem. Our solution, the Social-Forecast algorithm, makes multi-level popularity prediction in an online fashion, requiring no *a priori* training phase or dataset. It exploits the dynamically changing and evolving video propagation patterns through social media to maximize the prediction reward. The algorithm is easily tunable to enable tradeoffs between the accuracy and timeliness of the forecasts as required by various applications, entities and/or deployment scenarios.

- We analytically quantify the regret of Social-Forecast, that is, the performance gap between its expected reward and that of the best prediction policy which can be only obtained by an omniscient oracle having complete knowledge of the video popularity trends. We prove that the regret is sublinear in the number of video arrivals, which implies that the expected prediction reward asymptotically converges to the optimal expected reward. The upper bound on regret also gives a lower bound on the convergence rate to the optimal average reward.

- We validate Social-Forecast's performance through extensive experiments with real-world data traces from RenRen (the largest Facebook-like online social network in China). The results show that significant improvement can be achieved by exploiting the situational and contextual meta-data associated with the video and its propagation through the social media. Specifically, the Social-Forecast algorithm outperforms existing view-based approaches by more than 30% in terms of prediction rewards.

5.1 Related Works

In this section, we review the representative related works from both the application and the theoretical foundation perspectives.

5.1.1 Popularity Prediction for Online Content

Popularity prediction of online content has been extensively studied in the literature. Early works have focused on predicting the future popularity of content (e.g. video) on conventional websites such as YouTube. Various solutions are proposed based on time series models like ARIMA (Autoregressive integrated moving average) [NLL11] [GCM11] [ABB11], regression models [WSW12] [LMS10] [Row11] and classification models [WSW12] [SYK11] [SCN10]. These methods are generally view-based, meaning that the prediction of the future views is solely based on the early views, while disregarding the situational context during propagation. For instance, it was found that a high correlation exists between the number of video views on early days and later days on YouTube [CKR07]. By using the history of views within the past 10 days, the popularity of videos can be predicted up to 30 days ahead [SH10]. While these predictions methods provide satisfactory performance for YouTube-like accesses, their performance is largely unacceptable [LMW13] when applied to predicting popularity in the social media context. This is because in this case the popularity of videos evolves in a significantly different manner which is highly influenced by the situational and contextual characteristics of the social networks in which the video has propagated [LLX12].

Recently, there have been numerous studies aiming to accurately predicting the popularity of content promoted by social media [CHB09] [SOM10] [AH10] [YCK11] [YTL11] [KMG12] [RMZ13]. For instance, a propagation model is proposed in [GAC10] to predict which users are likely to mention which URLs on Twitter. In [HDD11], the retweets prediction on Twitter is modeled as a classification problem, and a variety of context-aware features are investigated. For predicting the popularity of news in Digg, such aspects as website design have been incorporated [LH10], and for predicting the popularity of short messages, the structural characteristics of social media have been used [BSH13]. For video

sharing in social media, our earlier work [LMW13] has identified a series of context-aware factors which influence the propagation patterns.

Our work in this chapter is motivated by these studies, but it is first systematic solution for forecasting the video popularity based on the situational and contextual characteristics of social media. First, existing works are mostly measurement-based and their solutions generally work offline, requiring existing training data sets. Instead, Social-Forecast operates entirely online and does not require any a priori gathered training data set. Second, Social-Forecast is situation-aware and hence it can inherently adapt on-the-fly to the underlying social network structure and user sharing behavior. Last but not least, unlike the early empirical studies which employ only simulations to validate the performance of their predictions, we can rigorously prove performance bounds for Social-Forecast.

Importantly, our Social-Forecast can be easily extended to predict other trends in social media (such as predicting who are the key influencers in social networks, which tweets and news items may become viral, which content may become popular or relevant etc.) by exploiting contextual and situational awareness. For instance, besides popularity, social media has been playing an increasingly important role in predicting present or near future events. Early studies show that the volume and the frequency of Twitter posts can be used to forecast box-office revenues for movies [AH10] and detect earthquakes [SOM10]. Sentiment detection is investigated in [BF10] by exploring characteristics of how tweets are written and meta-information of the words that compose these messages. In [CHB09], Google Trends uses search engine data to forecast near-term values of economic indicators, such as automobile sales, unemployment claims, travel destination planning, and consumer confidence. Social-Forecast can be easily adapted for deployment in these applications as well.

Table 5.1 provides a comprehensive comparison between existing works on popularity prediction and Social-Forecast, highlighting their differences.

5.1.2 Quickest Detection and Contextual Bandits Learning

In our problem formulation, for each video, the algorithm can choose to make a prediction decision using the currently observed context information or wait to make this prediction until the next period, when more context information arrives. This introduces a tradeoff between accuracy and delay which relates to the literature on quickest detection [PH09] [Kri12] [LFP08] which is concerned with the problem of detecting the change in the underlying state (which has already occurred in the past). For example, authors in [LFP08] study how to detect the presence of primary users by taking channel sensing samples in cognitive radio systems. In the considered problem, there is no underlying state; in fact, the state is continuously and dynamically changing, and the problem becomes forecasting how it will evolve and which event will occur in the future. Moreover, many quickest detection solutions assume prior knowledge of the hypotheses [LFP08] while this knowledge is unknown a priori in our problem and needs to be discovered over time to make accurate forecasts.

Our forecasting algorithm is based on the contextual bandits framework [TZS14] [Sli14] [DHK11] [LZ08] [CLR11] but with significant innovations aimed at tackling the unique features of the online prediction problem. First, most of the prior work [Sli14] [DHK11] [LZ08] [CLR11] on contextual bandits is focused on an agent making a single-stage decision based on the provided context information for each incoming instance. In this work, for each incoming video instance, the agent needs to make a sequence of decisions at multiple stages. The context information is stage-dependent and is revealed only when that stage takes place. Importantly, the reward obtained by selecting an action at one stage depends on the actions chosen at other stages and thus, rewards and actions at different stages are coupled. Second, in existing works [TZS14] [Sli14] [DHK11] [LZ08] [CLR11], the estimated rewards of an action can be updated only after the action is selected. In our problem, because the prediction action does not affect the underlying popularity evolution, rewards can be computed and updated even for actions that are not selected. In particular, we update the reward of an action as if it was selected. Therefore, exploration becomes virtual in the sense that explic-

	Situational awareness	Online/offline algorithm	Analytic performance	Multi-stage decision	Timeliness of prediction
Existing works	*No/Partially*	*Offline*	*No*	*No*	*No*
This paper	*Yes*	*Online*	*Yes*	*Yes*	*Yes*

Table 5.1: Comparison with existing works on popularity prediction for online content.

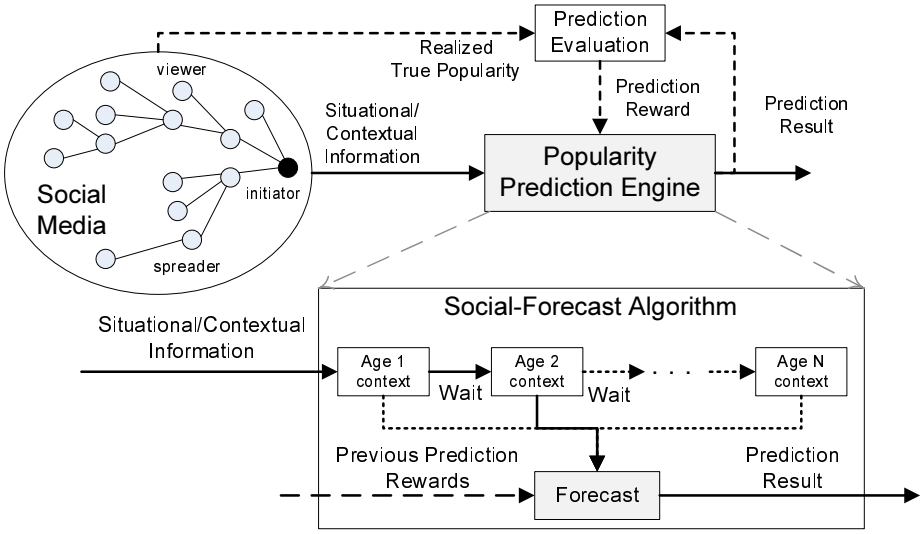


Figure 5.1: System diagram.

it explorations are not needed and hence, in each period, actions with the best estimated rewards can always be selected, thereby improving the learning performance.

5.2 System Model

5.2.1 Sharing Propagation and Popularity Evolution

We consider a generic Web 2.0 information sharing system in which videos are shared by users through social media (see Figure 6.1 for a system diagram). We assign each video with an index $k \in \{1, 2, ..., K\}$ according to the absolute time t_{init}^k when it is initiated[1]. Once a video is initiated, it will be propagated through the social media for some time duration. We assume a discrete time model where a period can be minutes, hours, days, or any suitable time duration. A video is said to have an age of $n \in \{1, 2, ...\}$ periods if

[1]It is easy to assign unique identifiers if multiple videos which are generated/initiated at the same time.

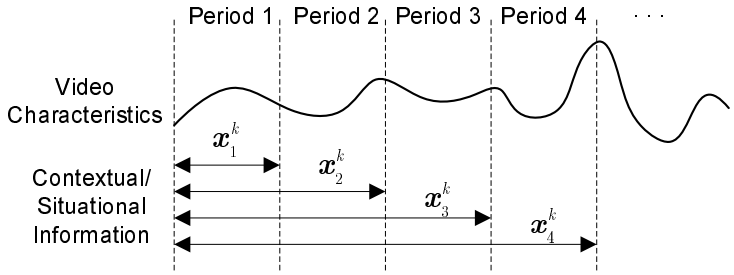


Figure 5.2: An illustration of context information taking the history characteristics.

it has been propagated through the social media for n periods. In each period, the video is further shared and viewed by users depending on the sharing and viewing status of the previous period. The propagation characteristics of video k up to age n are captured by a d_n-dimensional vector $\boldsymbol{x}_n^k \in \mathcal{X}_n$ which includes information such as the total number of views and other situational and contextual information such as the characteristics of the social network over which the video was propagated. The specific characteristics that we use in this work will be discussed in Section VI. In this section, we keep $\boldsymbol{x}_n^k$ in an abstract form and call it succinctly the *context (and situational) information* at age n.

Several points regarding the context information are noteworthy. First, the context space $\mathcal{X}_n$ can be different at different ages n. In particular, $\boldsymbol{x}_n^k$ can include all history information of video k's propagation characteristics up to age n and hence $\boldsymbol{x}_n^k$ includes all information of $\boldsymbol{x}_m^k, \forall m < n$ (See Figure 5.2). Thus the type of contextual/situational information is also age-dependent. Second, $\boldsymbol{x}_n^k$ can be taken from a large space, e.g. a finite space with a large number of values or even an infinite space. For example, some dimensions of $\boldsymbol{x}_n^k$ (e.g. the Sharing Rate used in Section VI) take values from a continuous value space and $\boldsymbol{x}_n^k$ may include all the past propagation characteristics (e.g. $\boldsymbol{x}_m^k \in \boldsymbol{x}_n^k, \forall m < n$). Third, at age n, $\boldsymbol{x}_m^k, \forall m > n$ are not yet revealed since they represent future situational and contextual information which is yet to be realized. Hence, given the context information $\boldsymbol{x}_n^k$ at age n, the future context information $\boldsymbol{x}_m^k, \forall m > n$ are random variables.

We are interested in predicting the future popularity status of the video by the end of a pre-determined age N, and we aim to make the prediction as soon as possible. The choice of N depends on the specific requirements of the content provider, the advertiser and the

web hosts. In this work, we will treat N as given[2]. Thus, the context information for video k during its lifetime of N periods is collected in $\boldsymbol{x}^k = (\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, ..., \boldsymbol{x}_N^k)$. For expositional simplicity, we also define $\boldsymbol{x}_{n^+} = (\boldsymbol{x}_{n+1}, ..., \boldsymbol{x}_N)$, $\boldsymbol{x}_{n^-} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_{n-1})$ and $\boldsymbol{x}_{-n} = (\boldsymbol{x}_{n^-}, \boldsymbol{x}_{n^+})$.

Let $\mathcal{S}$ be the popularity status space, which is assumed to be finite. For instance, $\mathcal{S}$ can be either a binary space {Popular, Unpopular} or a more refined space containing multiple levels of popularity such as {Low Popularity, Medium Popularity, High Popularity} or any such refinement. We let s^k denote the popularity status of video k by the end of age N. Since s^k is realized only at the end of N periods, it is a random variable at all previous ages. However, the conditional distribution of s^k will vary at different ages since they are conditioned on different context information. In many scenarios, the conditional distribution at a higher age n is more informative for the future popularity status since more contextual information has arrived. Nevertheless, our model does not require this assumption to hold.

5.2.2 Prediction Reward

For each video k, at each age $n = 1, ..., N$, we can make a prediction decision $a_n^k \in \mathcal{S} \cup \{\text{Wait}\}$. If $a_n^k \in \mathcal{S}$, we predict a_n^k as the popularity status by age N. If $a_n^k = \text{Wait}$, we choose to wait for the next period context information to decide (i.e. predict a popularity status or wait again). When the prediction is used to make an one-shot decision (e.g. ad investment), introducing a "Wait" option is of significant importance to allow trade-off between accuracy and timeliness. For each video k, at the end of age N, given the decision action vector $\boldsymbol{a}^k$, we define the *age-dependent reward* r_n^k at age n as follows,

$$r_n^k = \begin{cases} U(a_n^k, s^k, n), & \text{if } a_n^k \in \mathcal{S} \\ r_{n+1}^k, & \text{if } a_n^k = \text{Wait} \end{cases} \tag{5.1}$$

where $U(a_n^k, s^k, n)$ is a reward function depending on the accuracy of the prediction (determined by a_n^k and the realized true popularity status s^k) and the timeliness of the prediction (determined by the age n when the prediction is made).

[2]This assumption is generally valid given that the video sharing events have daily and weekly patterns, and the active lifespans of most shared videos through social media are quite limited [LWL12].

The specific form of $U(a_n^k, s^k, n)$ depends on how the reward is derived according to the popularity prediction based on various economical and technological factors. For instance, the reward can the ad revenue derived from placing proper ads for potential popular videos or the cost spent for adequately planning computation, storage, and bandwidth resources to ensure the robust operation of the video streaming services. Even though our framework allows any general form of the reward function, in our experiments (Section VI), we will use a reward function that takes the form of $U(a_n^k, s^k, n) = \theta(a_n^k, s^k) + \lambda\psi(n)$ where $\theta(a_n^k, s^k)$ measures the prediction accuracy, $\psi(n)$ accounts for the prediction timeliness and $\lambda > 0$ is a trade-off parameter that controls the relative importance of accuracy and timeliness.

Let n^* be the first age at which the action is not "Wait" (i.e. the first time a forecast is issued). The *overall prediction reward* is defined as the $r^k = r_{n^*}^k$. According to equation (5.1), when the action is "Wait" at age n, the reward is the same as that at age $n+1$. Thus $r_1^k = r_2^k = ... = r_{n^*}^k$. This suggests that the overall prediction reward is the same as the age-dependent reward at age 1, i.e. $r^k = r_1^k$. For age $n > n^*$, the action a_k^n and the age-dependent reward r_n^k do not affect the realized overall prediction result since a prediction has already been made. However, we still select actions and compute the age-dependent reward since it helps learning the best action and the best reward for this age n which in turn will help decide whether or not we should wait at an early age. Figure 5.3 provides an illustration on how the actions at different ages determine the overall prediction reward.

Remark: The prediction action itself does not generate rewards. It is the action (e.g. online ad investment) taken using the prediction results that is rewarding. In many scenarios, this action can only be taken once and cannot be altered afterwards. This motivates the above overall reward function formulation in which the overall prediction reward is determined by the first non-"Wait" action. Nevertheless, our framework can also be easily extended to account for more general overall reward functions which may depend on all non-"Wait" actions. For instance, the action may be revised when a more accurate later prediction is made. In this case, the reward function $U(a_n^k, s^k, n)$ in (5.1) will depend on not only the current prediction action $a_n^k \in \mathcal{S}$ but also all non-"Wait" actions after age n. We will use the reward function in (5.1) because of its simplicity for the exposition but our

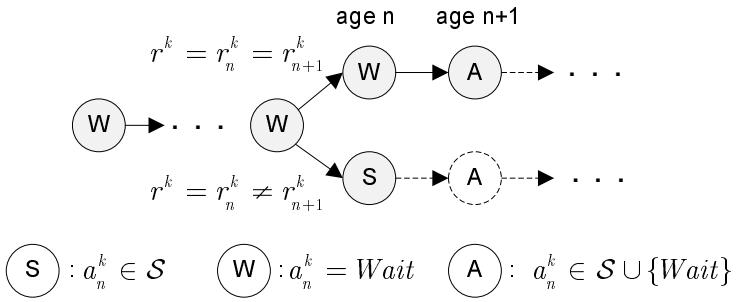


Figure 5.3: An illustration for the multi-stage decision making. The first $n-1$ action is "Wait". If the age-n action is "Wait", then $r_n^k = r_{n+1}^k$ which depends on later actions. If the age-n action is not "Wait", then $r_n^k \neq r_{n+1}^k$ and r^k does not depend on later actions. However, we can still learn the reward of action at age $n+1$ as if all actions before $n+1$ were "Wait".

analysis also holds for general reward functions.

5.2.3 Prediction Policy

In this work, we focus on prediction policies that depend on the current contextual information. Let $\pi_n : \mathcal{X}_n \to \mathcal{S} \cup \{\text{Wait}\}$ denote the prediction policy for a video link of age n and $\pi = (\pi_1, ..., \pi_N)$ be the complete prediction policy. Hence, a prediction policy π prescribes actions for all possible context information at all ages. For expositional simplicity, we also define $\pi_{n^+} = (\pi_{n+1}, ..., \pi_N)$ as the policy vector for ages greater than n, $\pi_{n^-} = (\pi_1, ..., \pi_{n-1})$ as the policy vector for ages smaller than n and $\pi_{-n} = (\pi_{n^-}, \pi_{n^+})$. For a video with context information $\boldsymbol{x}^k$, the prediction policy π determines the prediction action at each age and hence the overall prediction reward, denoted by $r(\boldsymbol{x}|\pi)$, as well as the age-dependent rewards $r_n(\boldsymbol{x}|\pi), \forall n = 1, ..., N$. Let $f(\boldsymbol{x})$ be the probability distribution function of the video context information, which also gives information of the popularity evaluation patterns. The expected prediction reward of a policy π is therefore,

$$V(\pi) = \int_{\boldsymbol{x} \in \mathcal{X}} r(\boldsymbol{x}|\pi) f(\boldsymbol{x}) d\boldsymbol{x} \tag{5.2}$$

Note that the age-n policy π_n will only use the context information $\boldsymbol{x}_n$ rather than $\boldsymbol{x}$ to make predictions since $\boldsymbol{x}_{n^+}$ has not been realized at age n.

Our objective is to determine the optimal policy π^{opt} that maximizes the expected prediction reward, i.e. $\pi^{opt} = \arg\max_{\pi} V(\pi)$. In the following sections, we will propose a systematic methodology and associated algorithms that find the optimal policy for the case when $f(\boldsymbol{x})$ is known or unknown, which are referred to as the complete and incomplete information scenarios, respectively.

5.3 Why Online Learning is Important?

In this section, we consider the optimal policy design problem with the complete information of the context distribution $f(\boldsymbol{x})$ and compute the optimal policy π^{opt}. In the next section in which $f(\boldsymbol{x})$ is unknown, we will learn this optimal policy π^{opt} online and hence, the solution that we derive in this section will serve as the benchmark. Even when having the complete information, determining the optimal prediction policy faces great challenges: first, the prediction reward depends on *all* decision actions at *all* ages; and second, when making the decision at age n, the actions for ages larger than n are not known since the corresponding context information has not been realized yet.

Given policies π_{-n}, we define the expected reward when taking action a_n for $\boldsymbol{x}_n$ as follows,

$$\mu_n(\boldsymbol{x}'_n|\pi_{-n}, a_n) = \int_{\boldsymbol{x}} I_{\boldsymbol{x}_n=\boldsymbol{x}'_n} r_n(\boldsymbol{x}|\pi_{-n}, a_n) f(\boldsymbol{x}) d\boldsymbol{x} \tag{5.3}$$

where $I_{\boldsymbol{x}_n=\boldsymbol{x}'_n}$ is an indicator function which takes value 1 when the age-n context information is $\boldsymbol{x}'_n$ and value 0 otherwise. The optimal $\pi^*(\pi_{-n})$ given π_{-n} thus can be determined by

$$\pi^*_n(\boldsymbol{x}_n|\pi_{-n}) = \arg\max_{a} \mu(\boldsymbol{x}_n|\pi_{-n}, a), \forall \boldsymbol{x}_n \tag{5.4}$$

and in which we break ties deterministically. Equation (5.4) defines a best response function from a policy to a new policy $F : \Pi \to \Pi$ where Π is the space of all policies. In order to compute the optimal policy π^{opt}, we iteratively use the best response function in (5.4) using the output policy computed in the previous iteration as the input for the new iteration. Note that a computation iteration is different from a time period. "Period" is used to describe the time unit of the discrete time model of the video propagation. A period can be a minute, an hour or any suitable time duration. In each period, the sharing and viewing

statistics of a *specific* video may change. "Iteration" is used for the (offline) computation method for the optimal policy (which prescribes actions for *all* possible context information in *all* periods). Given the complete statistical information (i.e. the video propagation characteristics distribution $f(\boldsymbol{x})$) of videos, a new policy is computed using best response update in each iteration.

We prove the convergence and optimality of this best response update as follows.

Lemma 4. *$\pi_n^*(\boldsymbol{x}_n|\pi_{-n})$ is independent of $\pi_m, \forall m < n$, i.e. $\pi_n^*(\boldsymbol{x}_n|\pi_{-n}) = \pi_n^*(\boldsymbol{x}_n|\pi_{n^+})$.*

Proof. By the definition of age-dependent reward, the prediction actions before age n does not affect the age-n reward. Hence, the optimal policy depends only on the actions after age n. □

Lemma 1 shows that the optimal policy π_n at age n is fully determined by the policies for ages larger than n but does not depend on the policies for ages less than n. Using this result, we can show the best response algorithm converges to the optimal policy within a finite number of computation iterations.

Theorem 6. *Starting with any initial policy π^0, the best response update converges to a unique point π^* in N computation iterations. Moreover, $\pi^* = \pi^{opt}$.*

Proof. Given the context distribution $f(\boldsymbol{x})$ which also implies the popularity evolution, the optimal age-N policy can be determined in the first iteration. Since we break ties deterministically when rewards are the same, the policy is unique. Given this, in the second iteration, the optimal age-$(N-1)$ policy can be determined according to (5.4) and is also unique. By induction, the best response update determines the unique optimal age-n policy after $N+1-n$ iterations. Therefore, the complete policy is found in N iterations and this policy maximizes the overall prediction reward. □

Theorem 1 proves that we can compute the optimal prediction policy using a simple iterative algorithm as long as we have complete knowledge of the popularity evolution distribution. In practice, this information is unknown and extremely difficult to obtain, if not

possible. One way to estimate this information is based on a training set. Since the context space is usually very large (which usually involves infinite number of values), a very large volume of training set is required to obtain a reasonably good estimation. Moreover, existing training sets may be biased and outdated as social media evolves. Hence, prediction policies developed using existing training sets may be highly inefficient [SB98]. In the following section, we develop learning algorithms to learn the optimal policy in an online fashion, requiring no initial knowledge of the popularity evolution patterns.

5.4 Learning the Optimal Forecasting Policy with Incomplete Information

In this section, we develop a learning algorithm to determine the optimal prediction policy without any prior knowledge of the underlying context distribution $f(\boldsymbol{x})$. In the considered scenario, videos arrive to the system in sequence[3] and we will make popularity prediction based on past experiences by exploiting the similarity information of videos.

Since we have shown in the last section that we can determine the complete policy π using a simple iterative algorithm, we now focus mainly on learning π_n for one age by fixing the policies π_{-n} for other ages. Importantly, we will provide not only asymptotic convergence results but also prediction performance bounds during the learning process.

5.4.1 Learning Regret

In this subsection, we define the performance metric of our learning algorithm. Let σ_n be a learning algorithm of π_n which takes action $\sigma_n^k(\boldsymbol{x}_n^k)$ at instance k. We will use learning regret to evaluate the performance of a learning algorithm. Since we focus on π_n, we will use simplified notations in this section by neglecting π_{-n}. However, keep in mind that the age-n prediction reward depends on actions at all later ages a_{n^+} besides a_n when $a_n = \text{Wait}$. Let $\mu_n(\boldsymbol{x}_n|a_n)$ denote the expected reward when age-n context information is $\boldsymbol{x}_n$ and the

[3]To simplify our analysis, we will assume that one video arrives at one time. Nevertheless, our framework can be easily extended to scenarios where multiple videos arrive at the same time.

algorithm takes the action $a_n \in \mathcal{S} \cup \{\text{Wait}\}$.

The optimal action given a context $\boldsymbol{x}_n$ is therefore, $a^*(\boldsymbol{x}_n) = \arg\max_{a_n} \mu_n(\boldsymbol{x}_n|a_n)$ (with ties broken deterministically) and the optimal expected reward is $\mu_n^*(\boldsymbol{x}_n) = \mu_n(\boldsymbol{x}_n|a_n^*)$. Let $\Delta = \max_{\boldsymbol{x}_n \in \mathcal{X}_n}\{\mu_n^*(\boldsymbol{x}_n) - \mu_n(\boldsymbol{x}_n|a_n \neq a_n^*)\}$ be the maximum reward difference between the optimal action and the non-optimal action over all context $\boldsymbol{x}_n \in \mathcal{X}_n$. Finally, we let $r_n(\boldsymbol{x}_n^k|\sigma_n^k)$ be the realized age-n reward for video k by using the learning algorithm σ. The expected regret by adopting a learning algorithm σ_n is defined as

$$R_n(K) = \mathbb{E}\{\sum_{k=1}^{K} \mu_n^*(\boldsymbol{x}_n^k) - \sum_{k=1}^{K} r_n(\boldsymbol{x}_n^k|\sigma_n^k)\} \tag{5.5}$$

Our online learning algorithm will estimate the prediction rewards by selecting different actions and then choose the actions with best estimates based on past experience. The reward estimates of $a_n^k \in \mathcal{S}$ implicitly capture the likelihood of different popularity levels. The reward estimate of $a_n^k = \text{Wait}$ captures the reward of the best prediction strategy if the prediction is made at a later age. Thus our algorithm not only decides which prediction is the best at each age but also when to make the best prediction in order to maximize the prediction reward. One way to do this is to record the reward estimates without using the context/situational information. However, this could be very inefficient since for different contexts, the optimal actions can be very different. Another way is to maintain the reward estimates for each individual context $\boldsymbol{x}_n$ and select the action only based on these estimates. However, since the context space $\mathcal{X}_n$ can be very large, for a finite number K of video instances, the number of videos with the same context $\boldsymbol{x}_n$ is very small. Hence it is difficult to select the best action with high confidence. Our learning algorithm will exploit the similarity information of contexts, partition the context space into smaller subspaces and learn the optimal action within each subspace. The key challenge is how and when to partition the subspace in an efficient way. Next, we propose an algorithm that adaptively partitions the context space according the arrival process of contexts.

5.4.2 Online Popularity Prediction with Adaptive Partition

In this subsection, we propose the online prediction algorithm with adaptive partition (Adaptive-Partition) that adaptively partitions the context space according to the context arrivals. This will be the key module of the Social-Forecast algorithm. For analysis simplicity, we normalize the context space to be $\mathcal{X}_n = [0,1]^d$. We call a d-dimensional hypercube which has sides of length 2^{-l} a level l hypercube. Denote the partition of $\mathcal{X}_n$ generated by level l hypercubes by $\mathcal{P}_l$. We have $|\mathcal{P}_l| = 2^{ld}$. Let $\mathcal{P} := \cup_{l=0}^{\infty}\mathcal{P}_l$ denote the set of all possible hypercubes. Note that $\mathcal{P}_0$ contains only a single hypercube which is $\mathcal{X}_n$ itself. For each instance arrival, the algorithm keeps a set of hypercubes that cover the context space which are mutually exclusive. We call these hypercubes *active* hypercubes, and denote the set of active hypercubes at instance k by $\mathcal{A}_k$. Clearly, we have $\cup_{C\in\mathcal{A}_k} = \mathcal{X}_n$. Denote the active hypercube that contains $\boldsymbol{x}_n^k$ by C_k. Let $M_{C_k}(k)$ be the number of times context arrives to hypercube C_k by instance k. Once activated, a level l hypercube C will stay active until the first instance k such that $M_{C_k}(k) \geq A2^{pl}$ where $p > 0$ and $A > 0$ are algorithm design parameters. When a hypercube C_k of level l becomes inactive, the hypercubes of level $l+1$ that constitute C_k, denoted by $\mathcal{P}_{l+1}(C_k)$, are then activated.

When a context $\boldsymbol{x}_n^k$ arrives, we first check to which active hypercube $C_k \in \mathcal{A}_k$ it belongs. Then we choose the action with the highest reward estimate $a_n = \arg\max_a \bar{r}_{a,C_k}(k)$, where $\bar{r}_{a,C_k}(k)$ is the sample mean of the rewards collected from action a in C_k which is an activated hypercube at instance k. When the prediction reward is realized for instance k (i.e. at the end of age N), we perform a *virtual update* for the reward estimates for all actions (see Figure 5.4). The reason why we can perform such a virtual update for actions which are not selected is because the context transition over time is independent of our prediction actions and hence, the reward by choosing any action can still be computed even though it is not realized.

Algorithm 1 provides a formal description for the Adaptive-Partition algorithm. Figure 5.6 illustrates the adaptive partition process of Adaptive-Partition algorithm. The intuition of our algorithm is as follows. Our algorithm learns the optimal action (whether to make

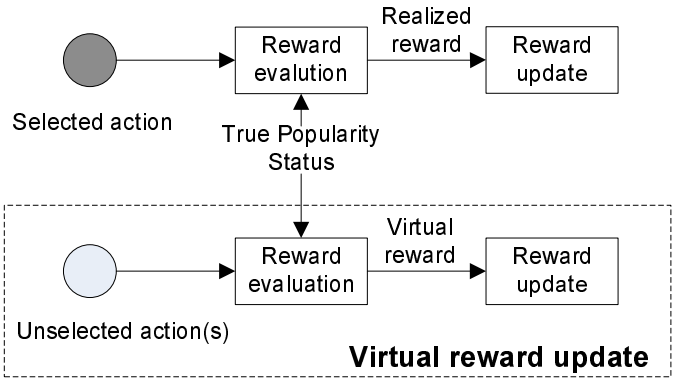


Figure 5.4: Illustration for virtual reward update in Adaptive Partition.

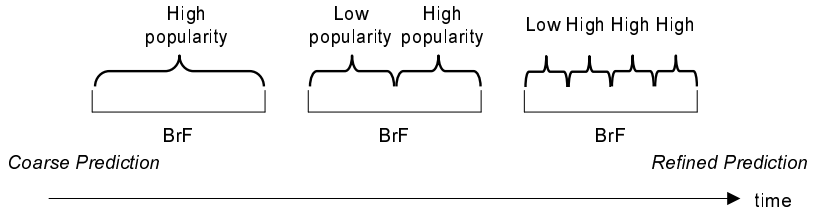


Figure 5.5: Learning is refined by context space partitioning.

a prediction or wait and which prediction to make) using sample mean reward estimates of different actions. The reward estimates are updated every time a new video instance comes and its popularity evolution pattern is realized. According to the law of large numbers, the reward estimates tend to be accurate as many videos have been seen. However, since there are a large number of different video evolution patterns, it is inefficient to maintain reward estimates for each pattern due to the small number of video instances for each individual pattern. Our algorithm exploits the similarity of video popularity evolution patterns to speed up learning. Specifically, initially we maintain reward estimates for the entire context space (i.e. by treating different patterns equally). These reward estimates are coarse but can be quickly updated since all video instances can be used. As we gather more video instances, the context space is gradually partitioned (i.e. by treating different patterns differently). As the partition becomes more and more refined, the reward estimates for each context subspace (i.e. cluster of patterns) become more and more accurate. Figure 5.5 illustrates the process of learning refinement assuming that the context only includes the BrF.

Next, we bound the regret by running the Adaptive-Partition algorithm. We make a

Algorithm 1 Adaptive-Partition Algorithm

Initialize $\mathcal{A}_1 = P_0$, $M_C(0) = 0$, $\bar{r}_{a,C}(0) = 0, \forall a, \forall C \in \mathcal{P}$.

for each video instance k **do**

Determine $C \in \mathcal{A}_k$ such that $\boldsymbol{x}_n^k \in C$.

Select $a_n = \arg\max_a \bar{r}_{a,C}(k)$.

After the prediction reward is realized, update $\bar{r}_{a,C}(k+1)$ for all a.

Set $M_C(k) \leftarrow M_C(k-1) + 1$.

if $M_C(k) \geq A2^{pl}$ **then**

Set $\mathcal{A}_{k+1} = (\mathcal{A}_k \backslash C) \cup \mathcal{P}_{l+1}(C)$

end if

end for

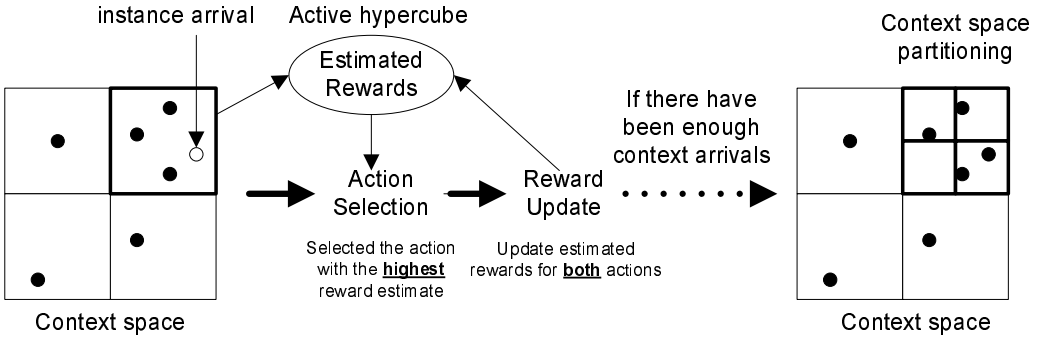


Figure 5.6: The context space partitioning of the Adaptive-Partition algorithm.

widely adopted assumption [Sli14] [DHK11] [LZ08] that the expected reward of an action is similar for similar contextual and situational information; we formalize this in terms of (uniform) Lipschitz condition.

Assumption. *(Lipschitz) For each $a_n \in \mathcal{S} \cup \{Wait\}$, there exists $L > 0, \alpha > 0$ such that for all $\boldsymbol{x}_n, \boldsymbol{x}'_n \in \mathcal{X}_n$, we have $|\mu(\boldsymbol{x}_n|a_n) - \mu(\boldsymbol{x}'_n|a_n)| \leq L\|\boldsymbol{x}_n, \boldsymbol{x}'_n\|^\alpha$.*

In order to get the regret bound of the Adaptive-Partition algorithm, we need to consider how many hypercubes of each level is formed by the algorithm up to instance K. The number of such hypercubes explicitly depends on the context arrival process. Therefore, we investigate the regret for different context arrival scenarios.

Definition. *We call the context arrival process* ***the worst-case arrival process*** *if it is uniformly distributed inside the context space, with minimum distance between any two context samples being* $K^{-1/d}$*, and* ***the best-case arrival process*** *if* $\boldsymbol{x}^k \in C, \forall k$ *for some level* $\lceil (\log_2(K)/p \rceil + 1$ *hypercube* C*.*

In Theorem 2, we determine the finite time, uniform regret bound for the Adaptive-Partition algorithm. The complete regret analysis and proofs can be found in the appendix.

Theorem 7.
- *For the worst case arrival process, if* $p = \frac{3\alpha+\sqrt{9\alpha^2+8\alpha d}}{2}$*, then* $R_n(K) = O(K^{\frac{d+\alpha/2+\sqrt{9\alpha^2+8\alpha d}/2}{d+3\alpha/2+\sqrt{9\alpha^2+8\alpha d}/2}})$.
- *For the best case arrival process, if* $p = 3\alpha$*, then* $R_n(K) = O(K^{2/3})$.

Proof. See Appendix. □

The regret bounds proved in Theorem 2 are sublinear in K which guarantee convergence in terms of the average reward, i.e. $\lim_{K\to\infty} \mathbb{E}[R_n(K)]/K = 0$. Thus our online prediction algorithm makes the optimal predictions as sufficiently many videos instances have been seen. More importantly, the regret bound tells how much reward would be lost by running our learning algorithm for any finite number K of videos arrivals. Hence, it provides a rigorous characterization on the learning speed of the algorithm.

5.4.3 Learning the Complete Policy π

In the previous subsection, we proposed the Adaptive-Partition algorithm to learn the optimal policy $\pi_n^*(\pi_{-n})$ by fixing π_{-n}. We now present in Algorithm 2 the Social-Forecast algorithm that learns the complete policy.

Social-Forecast learns all age-dependent policies $\pi_n, \forall n$ simultaneously. For a given age n, since π_{-n} is not fixed to be the optimal policy π_{-n}^{opt} during the learning process, the learned policy π_n may not be the global optimal π_n^{opt}. However, as we have shown in Section IV, in order to determine π_n^{opt}, only the policies for ages greater than n, i.e. $\pi_{n^+}^{opt}$ need to be determined. Thus even though we are learning $\pi_n, \forall n$ simultaneously, the learning problem

Algorithm 2 Social-Forecast Algorithm

for each video instance k **do**

  for each age $n = 1$ to N **do**

    Get context information $\boldsymbol{x}_n^k$.

    Select a_n^k according to Adaptive-Partition.

    Perform context partition using Adaptive-Partition.

  end for

  Popularity status s^k is realized.

  for each age $n = 1$ to N **do**

    Compute the age-dependent reward r_n^k.

    Update reward estimates using Adaptive-Partition.

  end for

end for

of π_N is not affected and hence, π_N^{opt} will be learned with high probability after a sufficient number of video arrivals. Once π_N^{opt} is learned with high probability, π_{N-1}^{opt} can also be learned with high probability after an additional number of video arrivals. By this induction, such a simultaneous learning algorithm can still learn the global optimal complete policy with high probability. In the experiments we will show the performance of this algorithm in practice.

5.4.4 Complexity of Social-Forecast

For each age of one video instance arrival, Social-Forecast needs to do one comparison operation and one update operation on the estimated reward of each forecast action. It also needs to update the counting of context arrivals to the current context subspace and perform context space partitioning if necessary. In sum, the time complexity has the order $O(|\mathcal{S}|N)$ for each video instance and $O(|\mathcal{S}|NK)$ for K video arrivals. Since the maximum age N of interest and the popularity status space is given, the time complexity is linear in the number of video arrivals K. The Social-Forecast algorithm maintains for each *active* context subspace reward estimates of all forecast actions. Each partitioning creates $2^d - 1$ more

active context subspaces and the number of partitioning is at most K/A. Thus the space complexity for K video arrivals is at most $O(2^d NK/A)$. Since the context space dimension d and the algorithm parameter A are given and fixed, the space complexity is at most linear in the number of video arrivals K.

5.5 Experiments

In this section we evaluate the performance of the proposed Social-Forecast algorithm. We will first examine the unique propagation characteristics of videos shared through social media. Then we will use these as the context (and situational) information for our proposed online prediction algorithm. Our experiments are based on the dataset that tracks the propagation process of videos shared on RenRen (`www.renren.com`), which is one of the largest Facebook-like online social networks in China. We set one period to be 2 hours and are interested in predicting the video popularity by 100 periods (8.3 days) after its initiation. In most of our experiments, we will consider a binary popularity status space {Popular, Unpopular} where "Popular" is defined for videos whose total number of views exceeds 10000. However, we also conduct experiments on a more refined popularity status space in Section VI(F).

Since our algorithm does not rely on specific assumptions on the selected reward function, we use two different prediction reward functions in our experiment in order to show the generality of our method. The first prediction reward function takes a linear form of accuracy and timeliness, namely $U(a_n^k, s^k, n) = \theta(a_n^k, s^k) + \lambda\psi(n)$ where $\theta(a_n^k, s^k)$ represents the accuracy of the prediction, $\psi(n)$ is the timeliness of the prediction and λ is a trade-off parameter. In particular, ψ is simply taken as $\psi(n) = N - n$. The second prediction reward function takes a discounted form of accuracy, namely $U(a_n^k, s^k, n) = \delta^n \theta(a_n^k, s^k)$ where $\delta \in [0, 1)$ is a discounted factor. For the case of binary popularity status space, the accuracy

reward function θ is chosen as follows

$$\theta(a_n^k, s^k) = \begin{cases} 1, & \text{if } a_n^k = s^k = \text{Unpopular} \\ w, & \text{if } a_n^k = s^k = \text{Popular} \\ 0, & \text{if } a_n^k \neq s^k \end{cases} \tag{5.6}$$

where $w > 0$ is fixed reward for correctly predicting popular videos and hence controls the relative importance of true positive and true negative. Note that we use these specific reward functions in this experiment but other reward functions can easily be adopted in our algorithm.

5.5.1 Video propagation characteristics

A RenRen user can post a link to a video taken by him/herself or from an external video sharing website such as Youtube. The user, referred to as an *initiators* [LMW13], then starts the sharing process. The friends of these initiators can find this video in their "News Feed". Some of them may watch this video and some may re-share the video to their own friends. We call the users who watched the shared video *viewers* and those who re-shared the video *spreaders*. Since spreaders generally watched the video before re-shared it, most of them are also viewers. In the experiment, we will use two characteristics of videos promoted by social media as the context (and situational) information for our algorithm. The first is the initiator's *Branching Factor (BrF)*, which is the number of users who are directly following the initiator and viewed the video shared by initiator. The second is the *Share Rate (ShR)*, which is the ratio of the viewers that re-share the video after watching it. Figure 5.7 shows the evolution of the number of views, the BrF and the ShR for three representative videos over 100 periods. Among these three videos, video 1 is an unpopular video while video 2 and video 3 are popular videos, which become popular at age 37 and age 51, respectively. We analyze the differences between popular and unpopular videos as follows.

- *Video 1 vs Video 2.* The ShRs of both videos are similar. The BrF of video 2 is much larger than that of video 1. This indicates that video 2 may be initiated by users with a large number of friends, e.g. celebrities and pubic accounts. Thus, videos with larger

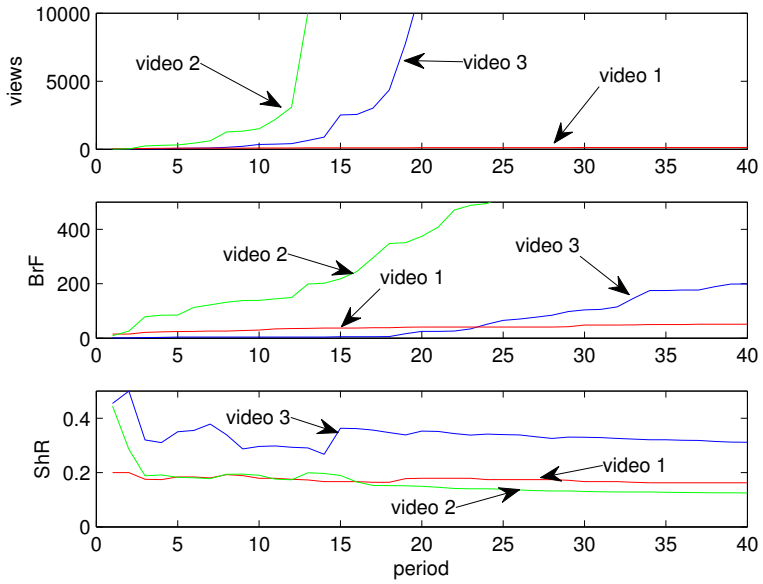


Figure 5.7: Popularity evolution of 3 representative videos.

BrF potentially will achieve popularity in the future.

- *Video 1 vs Video 3.* The BrFs of both videos are low (at least before video 3 becomes popular). Video 3 has a much larger ShR than video 1. This indicates that video 3 is being shared with high probability and thus, videos with larger ShR will potentially become popular in the future.

The above analysis shows that BrF and ShR are good situational metrics for videos promoted by social media. Therefore we will use these two metrics in addition to the total and per-period numbers of views as the context information for our proposed online prediction algorithms. The RenRen raw dataset records the information of each viewing action on RenRen. In particular, each data entry includes the URL of the video, the viewer id, the sharer id, the two-hop sharer id, the initiator and the time stamp when the view occurs. We pre-processed the raw dataset and extract for each video URL the viewing and sharing behavior over time such as BrF and ShR. Nevertheless, our algorithms are general enough to take other situational metrics to further improve the prediction performance, e.g. the type of the videos, the number of spreaders, other metrics representing the propagation topology etc.

5.5.2 Benchmarks

We will compare the performance of our online prediction algorithm with three benchmarks.

- **Szabo and Huberman (SH)**. The first benchmark is a conventional view-based prediction algorithm based on [SH10]. It uses training sets to establish log-linear correlations between the early number of views and the later number of views. Since this algorithm does not explicitly consider timeliness in prediction, we will investigate different versions that make predictions at different ages. Intuitively, the time when the prediction is made has opposite affects on the prediction accuracy and timeliness. A later prediction predicts the video with higher confidence but is less timely.

- **Pinto, Almeida and Goncalves (PAG)**. This benchmark is also a view-based prediction algorithm [PAG13]. Unlike SH which uses only the total view count by a reference time to predict future popularity, PAG incorporates the per-period view counts up to the reference date. Again, since this algorithm does not explicitly consider timeliness in prediction, we will also investigate different versions that make predictions at different ages.

- **Correlation using context information (CC)**. The above two benchmarks do not use siutational/contextual information as our proposed algorithm does. To enable fair comparison, we develop a modified prediction algorithm based on the ideas of SH and PAG by taking into consideration also the situational/contextual information. Specifically, the algorithm establishes (log-)linear correlations between the early number of views together with the context information (i.e. BrF and ShR) and the later number of views.

- **Perfect Prediction**. The last benchmark provides the best prediction results: for each unpopular video, it predicts unpopular at age 1; for each popular video, it predicts popular at age 1. Since this benchmark generates the highest possible prediction reward, we normalize the rewards achieved by other algorithms with respect to this reward.

5.5.3 Performance comparison

In this subsection, we compare the prediction performance of our proposed algorithm with the benchmarks. This set of experiments are carried out on a set of 5000 video links. The videos were initiated in sequence and thus, initially we do not have any knowledge of the videos or video popularity evolution patterns. For the SH (or PAG, CC) algorithm, we use three versions, labeled as SH-5 (or PAG-5, CC-5), SH-10 (or PAG-10, CC-10), SH-15 (or PAG-15, CC-15), in which the prediction is made at age 5, 10, 15, respectively.

Table 5.2 records the normalized prediction rewards (column 2 to 4) and the prediction accuracy (column 5) obtained by our proposed algorithm and the benchmarks for $\lambda = 0.01$ and $w = 1, 2, 3$ given the reward function $U(a_n^k, s^k, n) = \theta(a_n^k, s^k) + \lambda\psi(n)$. Table 5.3 records the normalized prediction rewards obtained by our proposed algorithm and the benchmarks for $w = 1$ and $\lambda = 0.01, 0.015, 0.02$. The trade-off parameter λ for accuracy and timeliness is set to be small because the lifetime N is large. We have the following observations:

- The accuracies of all three benchmarks are increasing in the reference age when the forecast is made. It implies that having more information is helpful for the prediction. The prediction rewards of the benchmarks are relatively insensitive to the time when the forecast is issued. This is because even though accuracy improves when the reference age is large, the prediction timeliness decreases. These two effects almost balance out in our experiments.

- The proposed algorithm Social-Forecast generates significantly higher prediction rewards than all benchmark algorithms. Its performance is not sensitive to the specific value of w which implies that it is able to predict both popular and unpopular videos very accurately and in a timely manner.

Table 5.4 shows the corresponding results for $\delta = 0.99$ given $U(a_n^k, s^k, n) = \delta^n\theta(a_n^k, s^k)$. We obtain similar observations even though a different reward function is used.

We then vary the popularity threshold. Table 5.5 reports the prediction rewards and accuracies for different thresholds 10000, 30000, 50000 for SH-10, PAG-10, CC-10 and

Table 5.2: Comparison of normalized prediction reward with varying w ($U(a_n^k, s^k, n) = \theta(a_n^k, s^k) + \lambda\psi(n)$)

	$w = 1$	$w = 2$	$w = 3$	accuracy
SH-5	0.82	0.82	0.81	0.80
SH-10	0.80	0.80	0.80	0.83
SH-15	0.78	0.80	0.81	0.84
PAG-5	0.71	0.77	0.80	0.64
PAG-10	0.79	0.82	0.84	0.81
PAG-15	0.78	0.82	0.85	0.85
CC-5	0.82	0.81	0.81	0.79
CC-10	0.79	0.80	0.81	0.80
CC-15	0.82	0.83	0.84	0.89
Social-Forecast	0.92	0.93	0.94	0.94

Table 5.3: Comparison of normalized prediction reward with varying λ ($U(a_n^k, s^k, n) = \theta(a_n^k, s^k) + \lambda\psi(n)$)

	$\lambda = 0.01$	$\lambda = 0.015$	$\lambda = 0.02$
SH-5	0.82	0.82	0.83
SH-10	0.80	0.79	0.79
SH-15	0.78	0.76	0.75
PAG-5	0.71	0.73	0.74
PAG-10	0.79	0.79	0.78
PAG-15	0.78	0.76	0.75
CC-5	0.82	0.82	0.83
CC-10	0.79	0.79	0.78
CC-15	0.82	0.79	0.78
Social-Forecast	0.92	0.93	0.94

Table 5.4: Comparison of normalized prediction reward with varying w ($U(a_n^k, s^k, n) = \delta^n \theta(a_n^k, s^k)$)

	$w = 1$	$w = 2$	$w = 3$	accuracy
SH-5	0.76	0.76	0.76	0.80
SH-10	0.75	0.74	0.73	0.83
SH-15	0.73	0.74	0.74	0.84
PAG-5	0.61	0.69	0.75	0.64
PAG-10	0.73	0.76	0.77	0.81
PAG-15	0.73	0.76	0.78	0.85
CC-5	0.76	0.76	0.76	0.79
CC-10	0.74	0.74	0.75	0.80
CC-15	0.77	0.77	0.77	0.89
Social-Forecast	0.92	0.91	0.92	0.92

Table 5.5: Comparison of normalized prediction reward and accuracy with varying popularity threshold (in each entry, the first number is the reward, the second number is the accuracy)

	SH-10	PAG-10	CC-10	Social-Forecast
1e4	0.80, 0.83	0.79, 0.81	0.79, 0.80	0.92, 0.94
3e4	0.81, 0.82	0.86, 0.90	0.83, 0.90	0.94, 0.96
5e4	0.84, 0.87	0.85, 0.89	0.81, 0.88	0.95, 0.96

Social-Forecast by fixing $\lambda = 0.01$ and $w = 1$ given the reward function $U(a_n^k, s^k, n) = \theta(a_n^k, s^k) + \lambda\psi(n)$. As the popularity threshold increases, the rewards and accuracies obtained by the Social-Forecast algorithm and SH-10 and PAG-10 all increase. In particular, the PAG algorithm has a significant increase in the prediction accuracy. This suggests that these benchmark algorithms have better accuracy in videos with a large number of views. However, the proposed Social-Forecast significantly outperforms the benchmarks in all categories.

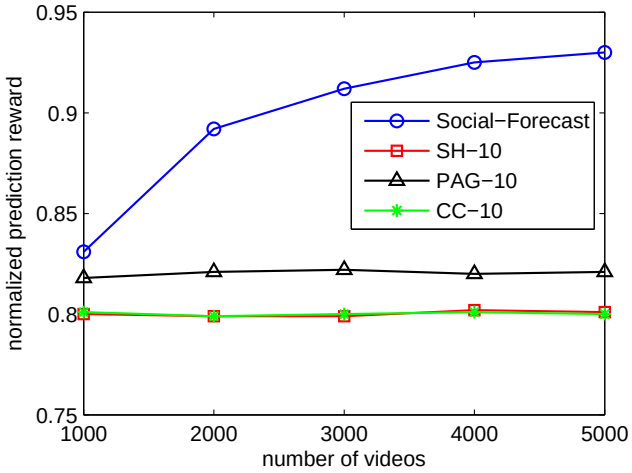


Figure 5.8: Prediction performance during the learning process.

5.5.4 Learning performance

Our proposed Social-Forecast algorithm is an online algorithm and does not require any prior knowledge of the video popularity evolution patterns. Hence, it is important to investigate the prediction performance during the learning process. Our analytic results have already provided sublinear bounds on the prediction performance for any given number of video instances which guarantee the convergence to the optimal prediction policy. Now, we show how much prediction reward that we can achieve during the learning process in experiments. Figure 5.8 shows the normalized prediction reward of Social-Forecast, SH-10, PAG-10 and CC-10 as the number of video instances increases for $\lambda = 0.10$ and $w = 2$. As more video instances arrive, our algorithm learns better the optimal prediction policy and hence, the prediction reward improves with the number of video instances. In particular, the proposed prediction algorithm is able to achieve more than 85% of the best possible reward even with a relatively small number of video instances. On the other hand, the normalized prediction rewards of the benchmark algorithms stay nearly invariant since they are trained offline and do not adapt to the new arriving videos.

Table 5.6: Comparison of normalized prediction reward for ternary popularity levels.

	$\lambda = 0.01$	$\lambda = 0.02$	$\lambda = 0.03$	accuracy
SH-5	0.71	0.75	0.78	0.63
SH-10	0.72	0.72	0.74	0.74
SH-15	0.70	0.68	0.68	0.76
PAG-5	0.66	0.71	0.75	0.57
PAG-10	0.74	0.74	0.74	0.73
PAG-15	0.73	0.70	0.69	0.76
CC-5	0.73	0.77	0.79	0.66
CC-10	0.73	0.73	0.74	0.72
CC-15	0.73	0.71	0.70	0.79
Social-Forecast	0.92	0.94	0.86	0.90

5.5.5 More refined popularity prediction

In the previous experiments, we considered a binary popularity status space. Nevertheless, our proposed popularity prediction methodology and associated algorithm can also be applied to predict popularity in a more refined space. In this experiment, we consider a refined popularity status space {High Popularity, Medium Popularity, Low Popularity} where "High Popularity" is defined for videos with more than TH_1 views, "Medium Popularity" for videos with views between $TH_2 < TH_1$ and TH_1, and "Low Popularity" for videos with views below TH_2. Table 5.6 illustrates the normalized rewards and accuracy obtained by different algorithms for $\lambda = 0.01, 0.02, 0.03$ and $TH_1 = 10000$, $TH_2 = 5000$. Table 5.7 reports the normalized rewards and accuracy of Social-Forecast, SH-10, PAG-10 and CC-10 by varying the High popularity threshold TH_1 given $\lambda = 0.02, w = 1$. It can been seen that the rewards obtained by all algorithms decrease compared with the binary popularity status case since prediction becomes more difficult. However, the performance improvement of Social-Forecast against the benchmark solutions becomes even larger. This suggests that our algorithm, which explicitly considers the contextual information associated with the social network, is able to achieve a higher performance gain against the benchmark approaches for more refined popularity prediction.

Table 5.7: Comparison of normalized prediction reward and accuracy with varying popularity threshold (in each entry, the first number is the reward, the second number is the accuracy)

	SH-10	PAG-10	CC-10	Social-Forecast
1e4	0.72, 0.74	0.74, 0.73	0.74, 0.72	0.94, 0.90
3e4	0.75, 0.75	0.79, 0.81	0.79, 0.81	0.91, 0.94
5e4	0.76, 0.74	0.77, 0.78	0.75, 0.75	0.89, 0.92

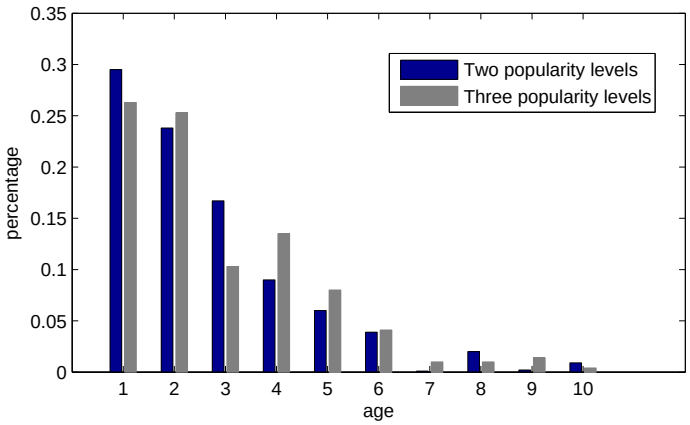


Figure 5.9: Distribution of ages at which the forecasts are made.

5.5.6 Prediction timeliness

Finally, we investigate at which age the forecast is actually made by our proposed Pop-Forecast algorithm. Figure 5.9 shows the percentage of the forecasts made at ages between 1 to 10 for the cases of binary and ternary popularity levels. As we can see, most of our forecasts are made at early ages of the video propagation, yet the accuracy is still very high by incorporating the contextual information of the social network in which the video is propagated.

5.6 Conclusions

In this work, we have proposed a novel, systematic and highly-efficient online popularity forecasting algorithm for videos promoted by social media. We have shown that by incor-

porating situational and contextual information, significantly better prediction performance can be achieved than existing approaches which disregard this information and only consider the number of times that videos have been viewed so far. The proposed Social-Forecast algorithm does not require prior knowledge of popularity evolution or a training set and hence can operate easily and successfully in online, dynamically-changing environments such as social media. We have systematically proven sublinear regret bounds on the performance loss incurred by our algorithm due to online learning. Thus Social-Forecast guarantees both short-term performance as well as its asymptotic convergence to the optimal performance in the long term.

This paper considered a single learner who observes the propagation patterns of videos promoted by one social media. One important future work direction is to extend to scenarios where there are multiple distributed learners (e.g. multiple advertisers, content producers and web hosts) who have access to multiple different social medias or different sections of one social media. In such scenarios, significant improvement is expected by enabling cooperative learning among the distributed learners [TZS14]. The challenges in these scenarios are how to design efficient cooperative learning algorithms with low communication complexity [XTZ15] and, when the distributed learners are self-interested and have conflicting goals, how to incentivize them to participate in the cooperative learning process using, e.g. rating mechanisms [XSS14] [XS14]. Finally, while this paper has studied the specific problem of online prediction of video popularity based on contextual and situational information, our methodology and associated algorithm can be easily adapted to predict other trends in social media (such as identifying key influencers in social networks, the potential for becoming viral of contents or tweets, identifying popular or relevant content, providing recommendations for social TV etc.).

5.7 Appendix

In this appendix, we analyze the learning regret of the Adaptive-Partition algorithm. The notations are summarized in Table 5.8. To facilitate the analysis, we artificially create two

Table 5.8: Notation Table

Notation	Description
$M_C(k)$	number of arrivals to C by k
$D(k)$	deterministic control function, $D(k) = k^z \log k$
$\mathcal{E}_{a,C}(k)$	set of rewards collected from action a by k for C
$a^*(C)$	optimal action for the center context of C
$\bar{\mu}_{a,C}$	maximum expected reward for contexts in C by taking a
$\underline{\mu}_{a,C}$	minimum expected reward for contexts in C by taking a
$\mathcal{L}_{C,l,B}$	suboptimal action set for C with level l given parameter B
A, p	algorithm parameters
L, α	Lipschitz condition parameters
$\mathcal{W}_C(k)$	event that the algorithm virtually exploits in C at k
$\mathcal{V}_{a,C}(k)$	event that a suboptimal action a is chosen in C at k
H_k	a positive number measuring the estimation gap

learning steps in the algorithms: for each instance k, it belongs to either a *virtual exploration* step or a *virtual exploitation* step. Let $M_C(k)$ be the number of context arrivals in C by video instance k. Given a context $\boldsymbol{x}_n^k \in C$, which step the instance k belongs to depends on $M_C(k)$ and a deterministic function $D(k)$. If $M_C(k) \leq D(k)$, then it is in a virtual exploration step; otherwise, it is in a virtual exploitation step. Notice that these steps are only used in the analysis; in the implementation of the algorithm, these different steps do not exist and are not needed.

We introduce some notations here. Let $\mathcal{E}_{a,C}(k)$ be the set of rewards collected from action a by instance k for hypercube C. For each hypercube C let $a^*(C)$ be the action which is optimal for the center context of that hypercube, and let $\bar{\mu}_{a,C} := \sup_{\boldsymbol{x}\in C} \mu(\boldsymbol{x}|a)$ and $\underline{\mu}_{a,C} := \inf_{\boldsymbol{x}\in C} \mu(\boldsymbol{x}|a)$. For a level l hypercube C, the set of suboptimal action is given by

$$\mathcal{L}_{C,l,B} := \{a : \underline{\mu}_{a^*,C} - \bar{\mu}_{a,C} > BLd^{\alpha/2}2^{-l\alpha}\} \tag{5.7}$$

colorwhere $B > 0$ is a constant that will be determined later.

The regret can be written as a sum of three components:

$$R(K) = \mathbb{E}[R_e(K)] + \mathbb{E}[R_s(K)] + \mathbb{E}[R_n(K)] \tag{5.8}$$

where $R_e(K)$ is the regret due to virtual exploration steps by instance K, $R_s(K)$ is the regret due to sub-optimal action selection in virtual exploitation steps by instance K and $R_n(K)$ is the regret due to near-optimal action selections in virtual exploitation steps by instance K. The following series of lemmas bound each of these terms separately.

We start with a simple lemma which gives an upper bound on the highest level hypercube that is active at any instance k.

Lemma 5. *All the active hypercubes $\mathcal{A}_k$ at instance k have at most a level of $(\log_2 k)/p + 1$.*

Proof. Let $l+1$ be the level of the highest level active hypercube. Since there are totally k instances, we must have $\sum_{j=1}^{l} A2^{pj} < k$, otherwise the highest level active hypercube will be less than $l+1$. Summing up the left-hand side, we have for $k/A > 1$,

$$A\frac{2^{p(l+1)-1}}{2^p - 1} < k \Rightarrow 2^{pl} < \frac{k}{A} \Rightarrow l < \frac{\log_2(k)}{p} \tag{5.9}$$

□

The next three lemmas bound the regrets for any level l hypercube.

Lemma 6. *If $D(k) = k^z \log k$. Then, for any level l hypercube the regret due to virtual explorations by instance k is bounded above by $k^z \log k + 1$.*

Proof. Since the instance k belongs to a virtual exploration step if and only if $M_C(k) \leq D(k)$, up to instance K, there can be at most $\lceil k^z \log k \rceil$ virtual exploration steps for one hypercube. Therefore, the regret is bounded by $k^z \log k + 1$. □

Lemma 7. *Let $B = \frac{2}{Ld^{\alpha/2}2^{-\alpha}} + 2$. If $p > 0, 2\alpha/p \leq z < 1$, $D(k) = k^z \log k$, then for any level l hypercube C, the regret due to choosing suboptimal actions in virtual exploitation steps, i.e. $\mathbb{E}[R_{C,s}(K)]$, is bounded above by $2\beta_2$.*

Proof. Let Ω denote the space of all possible outcomes, and w be a sample path of the reward realization. The event that the algorithm virtually exploits in C at instance k occurs when exploitation condition holds and the current context falls in an active hypercube C and thus is given by

$$\mathcal{W}_C(k) := \{w : M_C(k) > D(k), \boldsymbol{x}_n^k \in C, C \in \mathcal{A}_k\}$$

We will bound the probability that the algorithm chooses a suboptimal arm in an virtual exploitation step in C, and then bound the expected number of times a suboptimal action is chosen by the algorithm. Recall that loss in every step is at most 1. Let $\mathcal{V}_{a,C}(k)$ be the event that a suboptimal action is chosen. Then

$$\mathbb{E}[R_{C,s}(K)] \leq \sum_{k=1}^{K} \sum_{a \in \mathcal{L}_{C,l,B}} P(\mathcal{V}_{a,C}(k), \mathcal{W}_C(k))$$

For any a, we have

$$\begin{aligned}
&\{\mathcal{V}_{a,C}(k), \mathcal{W}_C(k)\} \\
\subset &\{\bar{r}_{a,C}(k) \geq \bar{\mu}_{a,C} + H_k, \mathcal{W}_C(k)\} \\
&\cup \{\bar{r}_{a^*,C}(k) \leq \underline{\mu}_{a^*,C} - H_k, \mathcal{W}_C(k)\} \\
&\cup \{\bar{r}_{a,C}(k) \geq \bar{r}_{a^*,C}(k), \bar{r}_{a,C}(k) < \bar{\mu}_{a,C} + H_k, \\
&\bar{r}_{a^*,C}(k) > \underline{\mu}_{a^*,C} - H_k, \mathcal{W}_C(k)\}
\end{aligned}$$

for some positive number $H_k > 0$ that controls the reward estimate gap which will be determined later. This implies

$$\begin{aligned}
&P(\mathcal{V}_{a,C}(k), \mathcal{W}_C(k)) \\
\leq &P(\bar{r}_{a,C}^{best}(M_C(k)) \geq \bar{\mu}_{a,C} + H_k + Ld^{\alpha/2}2^{-l\alpha}, \mathcal{W}_C(k)) \\
+ &P(\bar{r}_{a^*,C}^{worst}(M_C(k)) \leq \underline{\mu}_{a^*,C} - H_k - Ld^{\alpha/2}2^{-l\alpha}, \mathcal{W}_C(k)) \\
+ &P(\bar{r}_{a,C}^{best}(M_C(k)) \geq \bar{r}_{a^*,C}^{worst}(M_C(k)), \\
&\quad \bar{r}_{a,C}^{best}(M_C(k)) < \bar{\mu}_{a,C} + H_k, \\
&\quad \bar{r}_{a^*,C}^{worst}(M_C(k)) > \underline{\mu}_{a^*,C} - H_k, \mathcal{W}_C(k))
\end{aligned}$$

Consider the last term in the above equation, we want to make it zero so that we can focus only on the first two terms. If $2H_k \leq (B-2)Ld^{\alpha/2}2^{-l\alpha}$, then the three events in the last term cannot happen simultaneously. Thus, if we let $H_k = k^{-z/2}$, $z \geq 2\alpha/p$ and $B = \frac{2}{Ld^{\alpha/2}2^{-\alpha}} + 2$, then the last probability is 0. For the first two terms, by using a Chernoff-Hoeffding bound, for any $a \in \mathcal{L}_{C,l,B}$, since on the event $\mathcal{W}_C(k)$, $M_C(k) \geq k^z \log k$, we have

$$P(\bar{r}_{a,C}^{best}(M_C(k)) \geq \bar{\mu}_{a,C} + H_k, \mathcal{W}_C(k)) \leq e^{-2(H_k)^2 k^z \log k} \leq e^{-2\log k} \leq \frac{1}{k^2}$$

and

$$P(\bar{r}_{a^*,C}^{worst}(M_C(k)) \leq \underline{\mu}_{a^*,C} - H_k, \mathcal{W}_C(k)) \leq e^{-2(H_k)^2 k^z \log k} \leq e^{-2\log k} \leq \frac{1}{k^2}$$

Finally, the regret due to virtual exploitation is bounded by $\sum_{k=1}^{K} \frac{1}{k^2} < \beta_2$. Therefore, $\mathbb{E}[R_{C,s}(K)] \leq 2\beta_2$. □

Lemma 8. *Let $B = \frac{2}{Ld^{\alpha/2}2^{-\alpha}} + 2$. If $p > 0, 2\alpha/p \leq z < 1$, $D(k) = k^z \log k$, then for any level l hypercube C, the regret due to choosing near optimal actions in virtual exploitation steps, i.e. $\mathbb{E}[R_{C,n}(K)]$, is bounded above by $2ABLd^{\alpha/2}2^{(p-\alpha)l}$.*

Proof. The one-step regret of any near optimal action a is bounded by $2BLd^{\alpha/2}2^{-l\alpha}$ according to the definition of near optimal actions. Since C remains active for at most $A2^{pl}$ context arrivals, we have

$$\mathbb{E}[R_{C,n}(K)] \leq 2ABLd^{\alpha/2}2^{(p-\alpha)l} \tag{5.10}$$

□

Now we are ready to prove Theorem 2.

Proof. We let $B = \frac{2}{Ld^{\alpha/2}2^{-\alpha}} + 2$.

Consider the worst-case. It can be shown that in the worst case the highest level hypercube has level at most $1 + \log_{2^{p+d}} K$. The total number of hypercubes is bounded by

$$\sum_{l=0}^{1+\log_{2^{p+d}} K} 2^{dl} \leq 2^{2d} K^{\frac{d}{d+p}} \tag{5.11}$$

We can calculate the regret from choosing near optimal action as

$$\mathbb{E}[R_n(K)] \leq 2ABLd^{\alpha/2} \sum_{l=0}^{1+\log_{2^{p+d}} K} 2^{(p-\alpha)l} \leq 2ABLd^{\alpha/2} 2^{2(d+p-\alpha)} K^{\frac{d+p-\alpha}{d+p}} \tag{5.12}$$

Since the number of hypercubes have the order $O(K^{\frac{d}{d+p}})$, regret due to virtual explorations is $O(K^{\frac{d}{d+p}+z} \log K)$, while regret due to suboptimal selection is $O(K^{\frac{d}{d+p}+z})$, for $z \geq \frac{2\alpha}{p}$. These three terms are balanced when $z = 2\alpha/p$ and $\frac{d+p-\alpha}{d+p} = \frac{d}{d+p} + z$. Solving for p we get

$$p = \frac{3\alpha + \sqrt{9\alpha^2 + 8\alpha d}}{2} \tag{5.13}$$

Substituting these parameters and summing up all the terms we get the regret bound.

Consider the best case, the number of activated hypercubes is upper bounded by $\log_2 K/p+1$, and by the property of context arrivals all the activated hypercubes have different levels. We calculate the regret from choosing near optimal arm as

$$\mathbb{E}[R_n(K)] \leq 2ABLd^{\alpha/2} \sum_{l=0}^{1+\log_2 K/p} 2^{p-\alpha} l \leq 2ABLd^{\alpha/2} \frac{2^{2(p-\alpha)}}{2^{p-\alpha}} K^{\frac{p-\alpha}{p}} \tag{5.14}$$

The terms are balanced by setting $z = 2\alpha/p$, $p = 3\alpha$. □

CHAPTER 6

The Design and Dynamics of Up-or-Out Evaluation

In many networks (e.g. organizations, firms, societies etc.), if an individual does not achieve a certain reputation/rank within a certain period of time, then he/she is retained from the network. This so called "up-or-out" evaluation system is practiced in many professions, although perhaps under distinctive names [Ria95] [RTL93] [Sio98] [GDL06]. For instance, in academia [Sio98], newly hired professors must impress their department with their accomplishments to be awarded tenure; those not awarded tenure within a fixed time may be terminated. In law firms [GDL06], associated lawyers who fail to achieve partner status within ten years of being hired are required to leave. Accounting, engineering and military are also examples of professions that exhibit characteristics of polices of the same type.

The goal of deploying an "up-or-out" evaluation system is to eliminate individuals of low quality and keep individuals of high quality so that the productivity of the whole network can be optimized. As an individual is working, he/she produces valuable outcomes and accumulates reputation according to these outcomes. Individuals of higher quality are likely to accumulate higher reputation (or achieve higher ranks) than individuals of lower quality after the same trial period and hence, they are more likely to survive in the "up-or-out" evaluation system. Reputation/rank in this way serves as a signal of an individual's quality. However reputation is not a perfect signal due to two major reasons. The obvious reason is that each single outcome is a noisy reflection of an individual's quality. Therefore, reputation is also a noisy reflection of quality. A perhaps less obvious but equally important reason is that the outcome produced by one individual may also encode the quality of other individuals in the network due to social interactions such as collaborations. Therefore, the reputation of an individual depends on not only his/her own quality but also the quality of other individuals

in the network as well as the intensity at which he/she interacts with them. Since whether an individual can pass the evaluation and stay in the network afterwards depends on how much reputation he/she can accumulate, individuals' reputations and the average quality of individuals in the network co-evolve.

In this paper, we build a population model to study the "up-or-out" evaluation system and explicitly consider the impact of noise and social interaction on the achievable productivity of individuals in the network. The "up-or-out" evaluation system that we consider consists of two parameters: how long the evaluation period is and how much reputation an individual needs to pass the evaluation. We seek to determine the optimal parameters that maximize the total productivity of the society subject to a total population constraint due to a resource constraint. Our model is stylized. It makes first steps towards understanding the up-or-out evaluation from the population point of view because (1) it explicitly models the stochastic and dynamic process of entry and exit of individuals; (2) it allows any general prior distribution of individuals' qualities; (3) various deployment scenarios such as admission rate control can be modeled and analyzed; (4) more sophisticated network effects such as cumulative advantage can be easily extended to. Our main results are as follows:

- We prove that given an evaluation system, there always exists at least one steady state in which the total population and the social quality (i.e. average quality of individuals in the society) become invariant of time. Hence, system design becomes computationally possible.

- Time-to-evaluation is set by the noise level in learning individuals' qualities with higher noise level requiring long trial period. Moreover, noise reduces the achievable total productivity by the optimal evaluation design.

- Social interaction such as collaboration reduces the average quality of individuals in the network since it effectively adds more noise to the evaluation. However, its impact on the total productivity depends on the value of collaboration. There are cases when having some level of collaboration improves the total productivity but having too much reduces instead.

- Cumulative advantage prevents the quality of an individual from being accurately learning even with an extended trail periods.

- When there are heterogeneous types of agents, a higher valuation of intra-type collaboration results in a more homogeneous population demographics while a higher valuation of inter-type collaboration promotes a more heterogeneous demographics.

Up-or-out rules have been empirically studied in [Wal05] [ZB92]. There is also much work in microeconomics literature that builds theoretical models to explain the emergence of up-or-out policies. One strand of literature builds up-or-out systems exogenously into their models. The model developed by Kahn and Huberman [KH88] explains that the rationale for up-or-out policies is provided by a bilateral moral-hazard problem due to the two-sided uncertainty between the firm and the worker. Waldman [Wal90] found that the retention decision serves a signal to other firms of employee ability, and thus helps reduce the information asymmetry between the firms. Rajan and Zingales [RZ00] found flat hierarchies will prevail in human-capital-intensive industries and will have up-or-out promotion systems. O'Flaherty and Siow [OS92] derived the up-or-out rules endogenously. They show that under the optimal solution, the firm will promote a junior worker when the posterior probability that the worker is of high quality rises above some standard and will dismiss the work when the posterior probability falls below the ability prior. Most of this microeconomics literature has a focus on the principal-agent interaction and studies the agent decision problem. The present paper has a very different focus: we study how to design the optimal "up-or-out" system using a population model. Existing work neglects some important aspects that are critical for a systematic design of the up-or-out policy on the population level: workers have much more diverse qualities than the often assumed binary levels; workers enter and exit and hence, the population is dynamically changing; agents interact with other agents and the social interaction complicates the learning about individual agents. They are explicitly modeled and analyzed in the present paper.

Our paper has a specific emphasis on the impact of the social interaction such as collaboration on the social productivity. In academia, there is a long-standing assumption that

research collaboration has a positive effect on publishing productivity. However, Lee and Bozeman [LB05] found that the net impacts of collaboration are less clear. Landry and Amara [LA98] shows that collaboration may undermine productivity using arguments based on connection costs. Bozeman and Corley [BC04] found that while collaboration may enhance the productivity of some parties, it may also reduce the productivity of others such as experienced researchers. The model in the present paper also captures the effect that the return of collaboration is individual-dependent. More importantly, it in addition shows that in the presence of an "up-or-out" evaluation system collaboration reduces the average quality of individuals in the network. This reduction in social productivity is because collaboration makes it more difficult for the system to accurately distinguish individuals of different qualities in noisy environment. However, the impact of collaboration on the achievable total productivity depends on the inherent value of collaboration.

The rest of this paper is organized as follows. In Section II, we present the model. In Section III, we prove the existence of and convergence to the steady states given an evaluation system design. Section IV provides the optimal design and discusses the impact of noise and social interaction. Section V presents two extensions. Finally, we conclude in Section VI.

6.1 Model

6.1.1 Quality and Reputation

We consider an infinite horizon continuous-time model with a continuum of individuals. Each individual enters the system (e.g. a scientific society, a company, an organization etc.) with an intrinsic quality q, which models his own productivity. We assume that there is no moral hazard problem in the model so the productivity only depends on the individuals' quality but not effect levels. The quality of individuals follow a prior probability distribution $f(q)$ on the support $[0, \infty)$. We denote $F(q)$ as the cumulative distribution. To simplify notations for our later analysis, we define $g(q) = qf(q)$ and $G(q) = \int_{q'=0}^{q} g(q')dq'$. Clearly, the mean of the intrinsic quality is $\bar{q} = G(\infty)$.

Our analysis applies to any general prior quality distribution. For illustration purpose, our simulations will use a specific family of distributions, namely the chi-squared distributions, which has the following form,

$$f(q) = \frac{1}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})} q^{\frac{k}{2}-1} e^{-\frac{q}{2}} \tag{6.1}$$

where the k is the parameter of the chi-squared distribution. Chi-squred distribution is one of the most widely used probability distribution in inferential statistics, which can be used for goodness of fit of an observed distribution to a theoretical one.

Individuals in the system work and produce valuable outputs. They work both individually and collaboratively with other individuals in the system. At any time t, an individual spends α fraction of time on collaborative work and $1-\alpha$ fraction of time on individual work. The parameter α models the intensity at which individuals interact/collaborate with other individuals. A larger α implies that the system has a higher level of social interactions. For example, in academia, researchers conduct both individual and collaborative research and write both single-authored papers and co-authored papers. In some scientific fields, such as biology and chemistry, research projects often involve multiple researchers due to high workload and the need for diverse expertise. However, in some other scientific fields, such as mathematics and economics, research projects often involve very few or, in many cases, only a single researcher.

The value of an individual work is determined by the individual's own quality q; the value of a collaborative work linearly depends on the quality of all collaborating individuals with a scaling parameter $\gamma > 0$. We assume that the collaboration/interaction is uniformly random. Hence, the expected value of a collaborative work equals the average quality of individuals in the system multiplied by γ. We call this average quality of individuals currently in the system the *social quality* and denote it by $Q(t)$ at time t [1]

Individuals accumulate reputations based on the working outputs by themselves. We denote $\theta(t)$ as the reputation of an individual of quality q at time t, which evolves according

[1]Strictly speaking, the value of a collaborative work depends on both the individual's own quality and the social quality, i.e. $aq+(1-a)Q(t)$ where a is the contribution of a single individual. However, we can regard $\alpha(1-a)$ as the new weight α representing the fraction of time that an individual spends on collaborative works. For notational simplicity, we adopt this simplified modeling choice.

to the following dynamics,

$$d\theta(t) = [(1-\alpha)q + \alpha\gamma Q(t)]dt + \sigma dB_t \tag{6.2}$$

In the above equation, (1) $\gamma > 0$ is a parameter that normalizes the value of a collaborative work with respect to an individual work with $\gamma > 1$ representing that collaboration is inherently more valuable than working along. (2) The working outputs are noisy reflections of the qualities. In a continuous time system, noise is often modeled using a Brownian motion diffusion. Hence, B_t is the standard Brownian motion process and $\sigma > 0$ is the noise level.

6.1.2 Entry, Exit and "Up-or-Out" Evaluation

The population in the system is dynamic since individuals enter and exit from the system. Individuals enter the society at a rate of λ_b mass per unit time, which means that the total mass of individuals entering the community in Δt time is $\lambda_b\Delta$. Individuals exit from the system according to an exogenous process due to, e.g., graduation, retirement, switching professions etc.. We model this using a Poisson arrival process with a rate λ_d starting at the time of entry of the individual, and at the first arrival instance the individual exits.

The system has a resource budget (e.g. money, job opportunities) and hence, there is an upper bound $\bar{P}$ on the total population that the system can support at any time. To meet the total population constraint, the system may have to expel some individuals from the system. The question is which individuals should be expelled so that the productivity of individuals remaining in the system is maximized. For this, the system designer designs an "up-or-out" evaluation rule based on reputations, which is formally defined as follows.

Definition 7. *An "up-or-out" evaluation rule consists of two components: the time-to-evaluation T and the reputation threshold Θ. An individual pass the evaluation if and only if $\theta(t_0 + T) \geq \Theta$ where t_0 is the entry time of the individual.*

Figure 6.1 illustrates an representative individual who fails to pass the evaluation and hence is expelled from the system. The "up-or-out" evaluation rule is simple since it contains only two elements. However a careful design is still needed for the best performance (i.e.

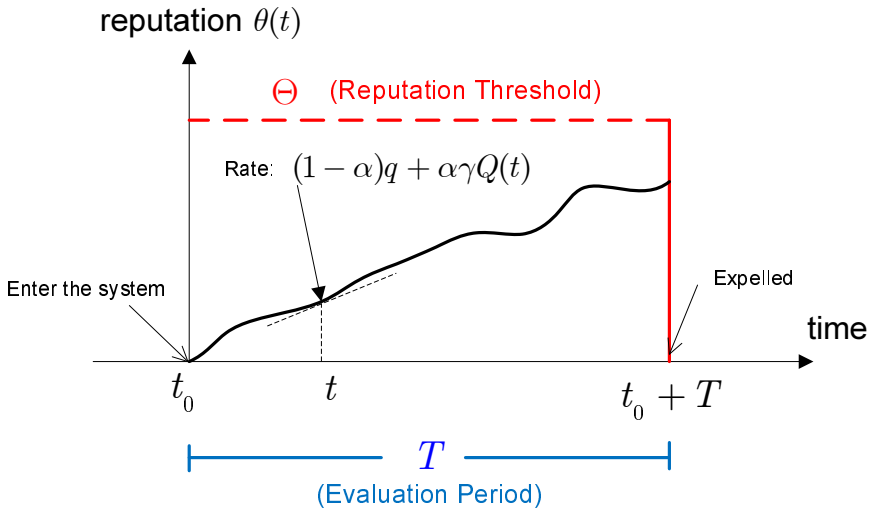


Figure 6.1: Illustration of entry, exit, and reputation evolution of an individual.

highest productivity) of the system. In the next sections, we will investigate how to design the optimal evaluation rule and how the different characteristics of the system influence the design as well as the achievable performance.

6.2 Steady State

The system at any time t can be completely characterized by the population mass distribution $p(q, \theta, t_0|t), \forall q, \theta, t_0$, which represents the population mass of individuals in the system who have quality q, reputation θ and entered the system at time t_0. This population mass distribution evolves over time as individuals of different qualities enter, exit and are expelled from the system under the evaluation rule. Let $p(q|t) = \int_{\theta=0}^{\infty} \int_{t_0=0}^{t} p(q, \theta, t_0|t) dt_0 d\theta$ denote the population mass of individuals of quality q. Then the total population mass at time t can be computed as $P(t) = \int_{q=0}^{\infty} p(q|t) dq$ and the social quality is $Q(t) = \frac{1}{P(t)} \int_{q=0}^{\infty} qp(q|t) dq$. The social productivity is the sum of working productivity of all individuals in the system, i.e.

$$W(t) = \int_{q=0}^{\infty} [(1-\alpha)q + \alpha\gamma Q(t)]p(q)dq = [(1-\alpha) + \alpha\gamma]P(t)Q(t) \tag{6.3}$$

The social quality affects the reputation that individuals can accumulate and hence affects which individuals can pass the evaluation. This introduces an endogenous coupling in the

system dynamics. To enable tractable analysis, we are interested in the steady state behavior of the system.

Definition 8. *The system is in steady state if $P(t), Q(t)$ are time invariant.*

When a society is in steady state, we can neglect the time subscript in the population mass variables. We note that since the considered model is continuous-time and has a continuum of population, it is possible that the system has a steady state even with noise. If the model is finite-time or has a finite population, there will be non-vanishing fluctuations in the population mass distribution.

With the above notations, we present the considered system design more formally as follows,

$$\begin{aligned} &\text{maximize}_{T,\Theta,(\lambda_b)} && W \\ &\text{subject to} && P \leq \bar{P} \end{aligned} \tag{6.4}$$

We consider two typical scenarios which require different evaluation system design.

- **Design without Admission Control**: in this scenario, the individual arrival rates are treated as exogenous and hence, the parameters T and Θ of the evaluation rule are the only design parameters.
- **Design with Admission Control**: in this scenario, the designer can also perform an admission rate control on the individuals entering the system by setting the arrival rate λ_b.

Next, we prove that the system indeed permits steady state(s) for any given evaluation rule. We also prove under certain conditions that the system converges to the steady state from any initial state. The existence proof also informs the optimal system design in the next section.

We start with Lemma 1, which states that, given any system design, there is a specific quality level that divides individuals into two categories.

Lemma 9. *Suppose the system is in steady state with social quality Q, then there exists a quality threshold $q_{th} = \frac{\Theta/T - \alpha\gamma Q}{1-\alpha}$ such that*

- *If $\sigma = 0$, then all individuals of quality $q \geq q_{th}$ pass the evaluation and individuals of quality $q < q_{th}$ fail.*
- *If $\sigma > 0$, then all individuals of quality $q \geq q_{th}$ pass the evaluation with probability higher than 0.5 and individuals of quality $q < q_{th}$ fail with probability higher than 0.5.*

Lemma 2 computes the total population and social quality if the quality threshold q_{th} is known.

Lemma 10. *Suppose the system is in steady state and the quality threshold is q_{th}, then*

- *If $\sigma = 0$, then $P(q_{th}) = \frac{\lambda_b}{\lambda_d}(1 - e^{-\lambda_d T} F(q_{th}))$ and $Q(q_{th}) = \frac{\bar{q} - e^{-\lambda_d T} G(q_{th})}{1 - e^{-\lambda_d T} F(q_{th})}$*
- *If $\sigma > 0$, then $P(q_{th}) = \frac{\lambda_b}{\lambda_d}(1 - e^{-\lambda_d T} \int_{q=0}^{\infty} \rho_q(q_{th}) f(q) dq)$ and $Q(q_{th}) = \frac{\bar{q} - e^{-\lambda_d T} \int_{q=0}^{\infty} \rho_q(q_{th}) g(q) dq}{1 - e^{-\lambda_d T} \int_{q=0}^{\infty} \rho_q(q_{th}) f(q) dq}$ where $\rho_q(q_{th}) = \Phi\left(\frac{(1-\alpha)\sqrt{T}}{\sigma}(q_{th} - q)\right)$ and $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.*

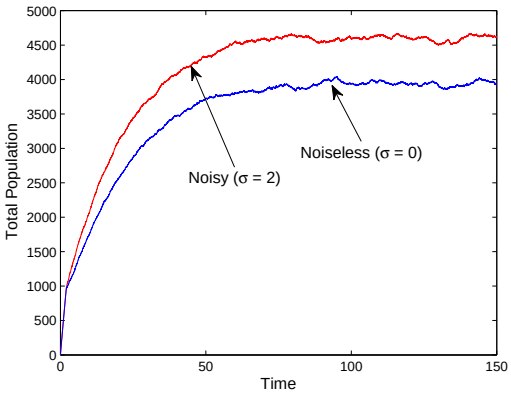
Note that in Lemma 2 the quality threshold q_{th} depends on the social quality Q (according to Lemma 1). Therefore, $Q(q_{th})$ is also a function of itself. The existence of a steady state requires that there is a solution to the following equation:

$$\frac{\Theta/T - (1-\alpha) q_{th}}{\alpha \gamma} = Q(q_{th}) \tag{6.5}$$

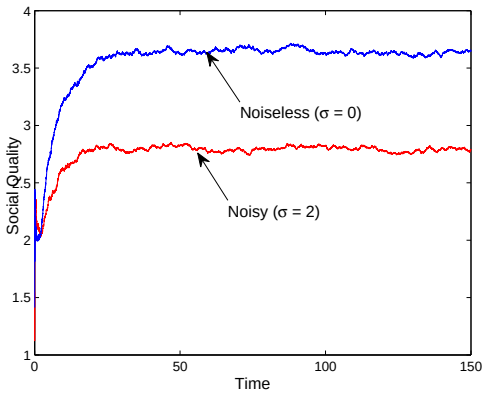
Theorem 8. *Given any system design, there exists at least one possible steady state.*

Theorem 1 proves the existence of steady states given any system design. This applies to both cases with and without noise. However, it does not exclude the possibility that there could be multiple steady states given a system design depending on the initial state of the system. Neither does it guarantee the convergence of the system to the steady state from any initial state. In the next two propositions, we establish uniqueness and convergence under certain conditions for the system without reputation noise. Convergence for the noisy case is much more difficult to prove if not impossible. We investigate this through simulations.

Proposition 8. *Suppose $\sigma = 0$. For a given system design, if $g(q) < \frac{(1-\alpha)(1-e^{-\lambda_d T})^2}{\alpha \gamma e^{-\lambda_d T}}$, then there exists a unique steady state of the system.*



(a) Total Population.



(b) Social Quality

Figure 6.2: Convergence of the system with and without noise.

Let $\beta_1 = \sup_{q_1>q_2} \frac{|G(q_1)F(q_2)-G(q_2)F(q_1)|}{q_1-q_2}$ and $\beta_2 = \sup_{q_1>q_2} \frac{|\bar{q}(F(q_1)-F(q_2))-(G(q_1)-G(q_2))|}{q_1-q_2}$ be two constants characterizing the smoothness of the quality distribution function $f(q)$, which are independent of the system design.

Proposition 9. *Suppose $\sigma = 0$. For a given system design, if $\frac{\alpha\gamma(e^{-2\lambda_d T}\beta_1+e^{-\lambda_d T}\beta_2)}{(1-\alpha)(1-e^{-\lambda_d T})^2} < 1$, the the system converges to the unique steady state from any initial state.*

Proposition 1 and 2 state that if the prior quality distribution $f(q)$ is well-behaved, namely it is smooth and vanishes rapidly enough, then the system converges to the unique steady state.

In Figure 6.2, we show the convergence of the system in terms of the total population and the social quality for scenarios with and without noise. Even though the same evaluation system (i.e. the same T, Θ and λ_b) is deployed, the system converges to different steady states depending on the noise level of the reputation updating. As we can see from the figure, when the noise is higher, the system tends to allow more individuals of low qualities to pass the evaluation and hence the total population is larger while the social quality is lower. In order to operate the system at the desired level, a careful evaluation rule design is thus needed and should take into account the amount of noise in the reputation update.

6.3 Optimal Design

In this section, we provide the optimal design of the evaluation system with and without admission control aimed at maximizing the social productivity in steady states. Then we will perform comparative statics to show how the social interaction and the noise level affect the system design and the achievable social productivity.

Before we proceed, we notice that given an evaluation time T, the reputation threshold Θ can be uniquely determined by the quality threshold through the equation

$$\Theta = [(1-\alpha)q_{th} + \alpha\gamma Q(q_{th})]T \tag{6.6}$$

Therefore, instead of designing the reputation threshold Θ directly, it is more convenient to first determine the optimal quality threshold q_{th} and use (6.6) to find the optimal Θ. In what follows, we analyze the design in terms of T and q_{th} but the conversion from q_{th} to Θ is straightforward. We first present an impossibility result, namely no evaluation system improves the social quality if there is infinite amount of noise.

Theorem 9. *For given* T, Θ, λ_b, $\lim_{\sigma\to\infty} Q(\sigma) \to \bar{q}$.

If there is a significant amount of noise, then the working output is randomly determined which reveals nothing about the intrinsic qualities of individuals. Therefore, no evaluation system is able to learn the true qualities of individuals and differentiate them to improve the social quality. Thus, the evaluation system design is only useful when the noise level is not too large.

6.3.1 Design without Admission Control

When there is no admission control, the design parameters are T and q_{th} (and hence Θ). For a given T, since both the total population $P(q_{th})$ and the social productivity $W(q_{th})$ are decreasing in q_{th} due to the fact that all individuals' qualities are non-negative, W is also increasing in P. Therefore, in the optimal design, we must have $P(T, q_{th}) = \bar{P}$. The optimal

design is thus formally presented as follows

$$(T^*, q_{th}^*) = \arg\max_{T, q_{th}} Q(T, q_{th}) \text{ s.t. } P(T, q_{th}) = \bar{P} \tag{6.7}$$

We first show how the quality threshold should be set depending on the time-to-evaluation if we want to achieve a certain total population.

Proposition 10. *Let $q_{th}(T)$ be the quality threshold given the evaluation time T such that $P(T, q_{th}(T)) = \bar{P}$. Then T must satisfy $T < -\frac{\log(1-\frac{\lambda_d}{\lambda_b}\bar{P})}{\lambda_d}$. Moreover,*

- *if $\sigma = 0$, $q_{th}(T)$ is increasing in T.*
- *if $\sigma > 0$, $q_{th}(T)$ is increasing in $T > \frac{1}{4\lambda_d}$.*

Proposition 3 states that there is an upper bound on the evaluation time in order to satisfy the population constraint. Moreover, if the evaluation time is longer, then the quality threshold should be set higher to eliminate more people from those who still remain in the network by the evaluation time. However, if there is noise in reputation update, this monotonicity property only holds when T is sufficiently large. In simulations (see Section V), we observed that when T is small, q_{th} decreases with T in order to have the same total population.

The next proposition characterizes the optimal time-to-evaluation.

Proposition 11. *Given a population constraint $\bar{P}$, in a system without admission control,*

- *if $\sigma = 0$, $W(T)$ is decreasing in T.*
- *if $\sigma > 0$, there exists $T^* > 0$, such that $\lim_{T\to 0} W(T) < W(T^*)$.*

Proposition 4 shows a significant difference in system design between scenarios with and without noise. If there is no noise in reputation update, the true quality of an individual are revealed immediately after she starts to produce output. Therefore, the evaluation should be performed immediately so that only individuals of sufficiently high qualities stay in the system and produce high value output. Even though by Proposition 3 the quality threshold

becomes higher if the evaluation is postponed and hence, individuals of even higher quality can pass the evaluation, the decrease in performance by allowing individuals of low quality stay in the system for a longer time outweighs the performance improvement by letting individuals of higher quality pass the evaluation. Therefore, it is better to evaluate as soon as possible the individuals when there is no noise.

When there is noise in reputation update, the design becomes significantly different. Since in early periods, the evaluation system is not able to accurately differentiate individuals of different qualities, evaluating individuals very soon could result in too many individuals of high quality expelled from the system and too many individuals of low quality pass the evaluation. Only after the system gathers sufficient much information about individuals' true quality should the evaluation be performed in order to achieve a high social productivity. The next proposition shows that this delay in evaluation due to noise results in inevitable loss in social productivity.

Proposition 12. *Given a population constraint $\bar{P}$, let $W^*(\sigma)$ be the optimal social productivity that can be achieved at noise level σ, then $W^*(\sigma)$ is decreasing in σ.*

6.3.2 Design with Admission Control

If the system designer can in addition control the arrival rate of new individuals, then the social productivity can be further improved. Since the population constraint can always been met by adjusting the arrival rate λ_b, the optimal design problem becomes an unconstrained optimization problem which aims at maximizing the social quality:

$$(T^*, q_{th}^*) = \arg\max_{T, q_{th}} Q(T, q_{th}) \tag{6.8}$$

The optimal arrival rate λ_b^* can be determined afterwards according to $\lambda_b^* = \frac{\lambda_b}{\bar{P}}(1-e^{-\lambda_d T^*}F(q_{th}^*))$ in the case without noise and $\lambda_b^* = \frac{\lambda_b}{\bar{P}}(1 - e^{-\lambda_d T^*}\int_{q=0}^{\infty}\rho_q(q_{th}^*)f(q)dq)$ in the case with noise.

The next proposition characterizes the impact of choosing different q_{th} on the achievable social quality when there is no noise.

Proposition 13. *If $\sigma = 0$, then the following statements about $Q(q_{th})$ are true*

1. $\lim_{q_{th}\to\infty} Q(q_{th}) = Q(0) = \bar{q}$.

2. *There exists a unique* $q^*_{th} > \bar{q}$ *such that* $Q(q_{th})$ *is increasing in* $[0, q^*_{th}]$ *and decreasing in* $[q^*_{th}, \infty)$.

Proposition 6 implies that for a given time-to-evaluation T, there is a unique quality threshold q^*_{th} that maximizes the social quality. Moreover, the threshold is not too small (at least larger than $\bar{q}$) and not too long. If the threshold is too small, then too many individuals of low quality pass the evaluation which degrades the average quality. If the threshold is too large, then too many individuals of high quality cannot pass the evaluation which also degrades the average quality. When there is noise in reputation update, the impact of q_{th} on $Q(q_{th})$ can be more complicated. However, we observe similar optimal quality threshold exists from simulations with the chi-squared distribution.

How to choose the optimal time-to-evaluation T? The next proposition again shows that if there is no noise, evaluation should be performed as soon as possible and if there is noise, it is better to wait for a certain time to perform evaluation.

Proposition 14. *Given a population constraint* $\bar{P}$*, in a system with admission control*

- *if* $\sigma = 0$, $W(T)$ *is decreasing in* T.
- *if* $\sigma > 0$, *there exists* $T^* > 0$, *such that* $\lim_{T\to 0} W(T) < W(T^*)$.

6.3.3 Simulations

Figure 6.3 illustrates the achievable normalized total population and social quality by varying the design parameters T and q_{th} when there is no noise, i.e. $\sigma = 0$.

- **Design without Admission Control**. As we can see from Figure 6.3a, in order to achieve a certain target total population, the quality threshold q_{th} has to increase with the evaluation time duration T. Moreover, given a population constraint, increasing the evaluation time always decreases the social quality (hence the social productivity). This implies that in the evaluation system should evaluate individuals as soon as possible.

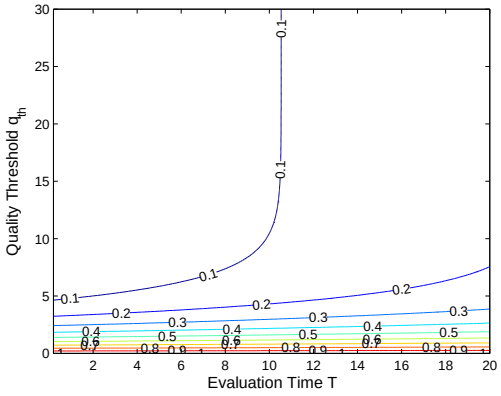
- **Design with Admission Control**. In this case, the system can freely choose q_{th} and T to maximize the social quality. Figure 6.3b shows that given a evaluation T, there is an optimal quality threshold q_{th}^* that is neither too small nor too large. It also shows that the maximal social quality is again achieved at $T = 0$, implying the evaluation should be carried out immediately. Moreover, the social quality can go to infinity as $T \to 0$ provided that the system can admit as many individuals as possible.

Figure 6.4 illustrates the achievable normalized total population and social quality by varying the design parameters T and q_{th} when the noise level is $\sigma = 2$. Significant differences are observed from the scenario without noise:

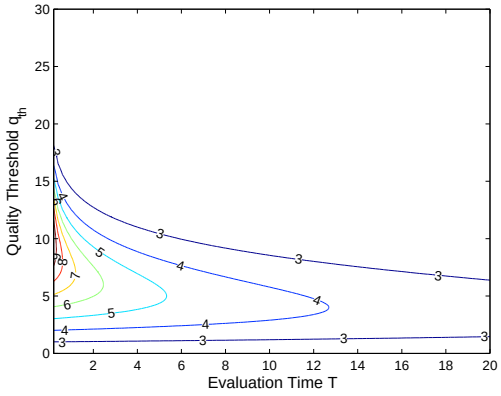
- **Design without Admission Control**. Figure 6.4a show that given a population constraint, it no longer holds that the quality threshold q_{th} increases with T. Instead, When T becomes close to 0, a very large q_{th} need to be used if the population constraint is small. Moreover, choosing a small T does not improve the social productivity as opposed to the case without noise. This is because when T is very small, reputation is too noisy to tell the true quality of an individual.

- **Design with Admission Control**. Figure 6.4b shows that given the evaluation time T, the optimal quality threshold is not too low or too high. However, due to similar reasons as before, the optimal evaluation rule evaluates individuals after a certain time. Moreover, the optimal design cannot achieve infinite social quality as opposed to the case without noise, thereby confirming that noise brings inevitable loss in social productivity.

Figure 6.5 shows the impact of the noise level on the evaluation rule design and the resulting performance with admission control. As shown in Figure 6.5b, the evaluation time has to be delayed more to accurately distinguish individuals of different qualities.

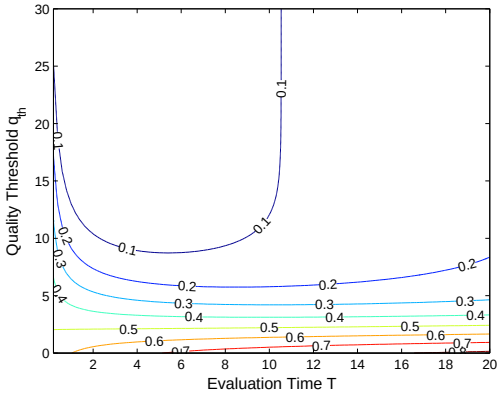


(a) Total Population.

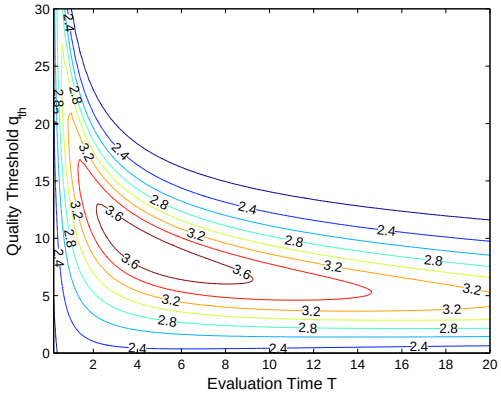


(b) Social Quality

Figure 6.3: Evaluation System Design without Noise. ($k = 2$)

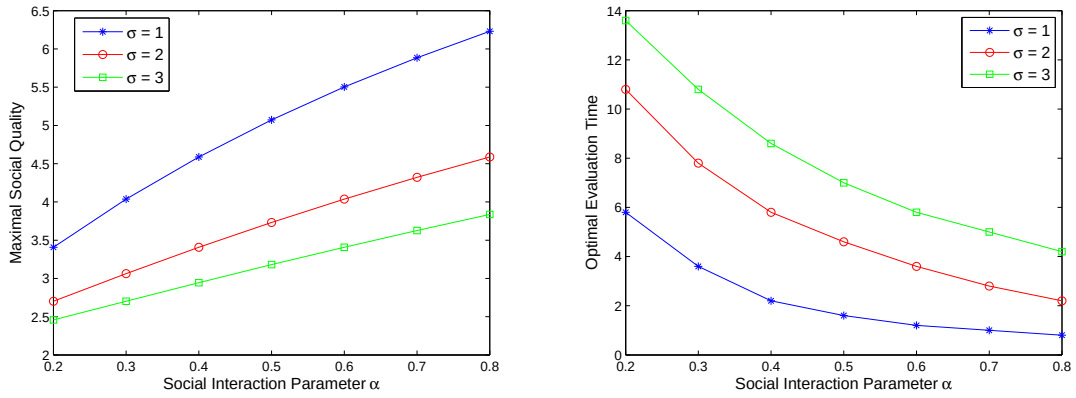


(a) Total Population.



(b) Social Quality

Figure 6.4: Evaluation System Design with Noise. ($k = 2$)



(a) Maximal Social Quality (b) Optimal Evaluation Time

Figure 6.5: Impact of noise and social interaction.

6.4 Roles of Social Interaction

The intensity at which individuals interact/collaborate with each other represents an important characteristic of a system. Given an evaluation rule, how does the changing in the interaction intensity the resulting population in the system? Does more interaction beneficial improve or harm the achievable total productivity of the network? These are the questions that we want to answer in this section.

Proposition 15. *Suppose $\sigma = 0$. Given an evaluation system T, Θ, λ_b, if $\gamma Q(\frac{T}{\Theta}) < \frac{\Theta}{T}$, then q_{th} increases with α; if $\gamma Q(\frac{T}{\Theta}) > \frac{\Theta}{T}$, then q_{th} decreases with α; if $\gamma Q(\frac{T}{\Theta}) = \frac{\Theta}{T}$, then q_{th} is independent of α.*

This above result shows that increasing the social interaction intensity may increase or decrease the fraction of individuals who can pass the evaluation depending on the specific evaluation system design and the prior quality distribution. In particular, there are evaluation systems that are robust to the change in social interaction intensities, meaning that the quality threshold, hence the total population and social productivity do not change with the social interaction intensity. In such evaluation systems, $q_{th} = \frac{\Theta}{T}$ and hence the value of individual work of individual of the threshold quality equals the value of a collaborative work.

Is social interaction beneficial to individuals and to the system as a whole? According to the reputation dynamics, the expected reputation of an individual of quality q is $[(1-\alpha)q + \alpha\gamma Q]T$. Thus individuals of quality lower than γQ will benefit by increasing the interaction intensity since they will achieve a higher reputation. Conversely, individuals of quality higher than γQ will have their reputation decreased if the social interaction intensity is higher. On the network level, the role of social interaction is less clear. Recall that the total productivity in steady state is

$$W = [(1-\alpha) + \alpha\gamma]QP \tag{6.9}$$

In this equation, $(1-\alpha) + \alpha\gamma$ represents the average value of a work (either individual or collaborative). If $\gamma > 1$, namely collaboration is inherently more valuable than working alone, then more intense social interaction increase the average value of a work. However, the total productivity also depends on the social quality that can be achieved by the evaluation system. In the next proposition, we prove that Q is indeed decreasing in the social interaction intensity α.

Proposition 16. *Given a population constraint $\bar{P}$, let $Q^*(\alpha)$ be the social quality that can be achieved by a system without admission control, then*

- *if $\sigma = 0$, then $Q^*(\alpha)$ is independent of α.*
- *if $\sigma > 0$, then $Q^*(\alpha)$ is decreasing in α.*

Proposition 9 implies that when there is no noise, social interaction does not change the optimal achievable social productivity. However, when there is noise, increasing the social interaction intensity decreases the achievable social quality. This is because when there is reputation updating noise, having more social interactions effectively introduces more noise in distinguishing the true quality of individuals, thereby reducing the social productivity.

The result of proposition 9 suggests that the social interaction is a double-edged sword: on one hand, more intense social interaction increases the inherent value of a work; on the other hand, more intense social interaction reduces the efficacy of the evaluation. The

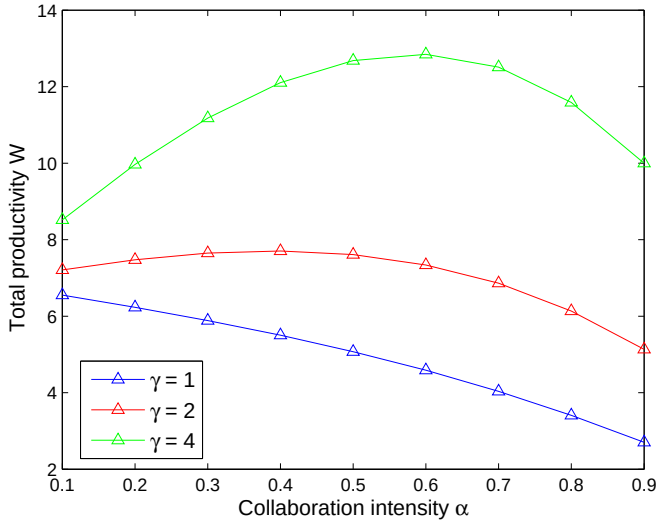


Figure 6.6: Impact of social interaction on the total productivity ($\gamma = 4$).

net value of collaboration depends on the inherent value of the collaborative work. In Figure 6.6, we show the achievable total productivity as a function of the social interaction intensity depending on the inherent value of a collaborative work. As we can see, if the value of collaboration is small, then more intense social interaction only leads to a lower total productivity. If collaboration is sufficiently valuable, then have some level of social interaction increases the total productivity. However, again, too much social interaction reduces the total productivity.

6.5 Discussions and Extensions

In the previous sections, we adopted some simplified modelling choices. In this section, we discuss these simplifications and provide some extensions.

6.5.1 Cumulative Advantage

Cumulative advantage effect, also known as the Matthew effect [Mer68], is a sociological phenomenon where eminent individuals get more advantages than less famous individuals in terms of resource, credit or opportunity. There are multiple ways to model cumulative

advantage in the current context. We discuss a few in what follows.

Non-uniform credit distribution. We have implicitly assumed that individuals regardless of its current reputation obtain the same credit from a collaborative work. One type of cumulative advantage is that individuals of higher reputation are given more credit than individuals of lower reputation. In this case, the individual's reputation dynamics becomes

$$d\theta = [(1-\alpha)q + \alpha\frac{\theta}{\bar{\theta}}\gamma Q]dt + \sigma dB_t \tag{6.10}$$

where $\bar{\theta}(t)$ is the average reputation of all individuals in the system. The reputation derived from collaborative works linearly depends on the individual's current reputation and hence, the higher reputation, the more credit. This credit is normalized using the average reputation so that the total credit of a collaborative work equals its value.

Preferential attachment [JNB03]. We assumed that the social interaction intensity is homogeneous across individuals, which implies that all individuals have the same opportunity to collaborate. A second type of cumulative advantage is that individuals of higher reputations tend to have more opportunities to collaborate and hence, the social interaction intensity is reputation dependent. In this case, the reputation dynamics become

$$d\theta = [(1-\alpha_\theta)q + \alpha_\theta\gamma Q(t)]dt + \sigma dB_t \tag{6.11}$$

where α_θ is the reputation-dependent interaction intensity which is increasing in θ.

Non-uniform matching. We also assumed that individuals in the system are randomly matched to collaborate. The result of this assumption is that the value of a collaborative work linearly depends on the average quality of individuals in the system. A third type of cumulative advantage is that individuals of similar reputations tend to collaborate more often. In this case, the value of collaborative work is reputation-dependent. For instance,

$$d\theta = [(1-\alpha)q + \alpha\gamma Q_\theta(t)]dt + \sigma dB_t \tag{6.12}$$

where $Q_\theta(t)$ is the average quality of individuals of reputation θ. It is obvious to see that Q_θ is increasing in θ since high quality individuals are more likely to have high reputation.

When there is no noise in the reputation dynamics (i.e. $\sigma = 0$), it can be easily seen that in all three types of cumulative advantage, higher quality individuals will have higher

reputation at the evaluation time T. Since the reputation preserves the order of quality, each quality threshold is mapped to a unique reputation threshold. Therefore, the design in terms of the quality threshold remains the same as before. In particular, the design is solved by $(q_{th}^*, T^*) = \arg\max_{q_{th},T} W(q_{th}, T)$. Using the optimal quality threshold q_{th}^*, the social quality Q and the reputation-dependent social quality Q_θ can be computed. Then, the corresponding reputation threshold can computed $\Theta^* = \mathbb{E}(\theta|q_{th}^*)$.

When there is noise, analysis can be much more complicated and design can be much more difficult. We illustrate this using the first type of cumulative advantage, i.e. non-uniform credit distribution. In this case, we show that there is a non-diminishing probability that high quality individuals will fail the evaluation.

Proposition 17. *With the first type of cumulative advantage, individuals of quality $q > q_{th}$ has a non diminishing probability of failing the evaluation even if the time-to-evaluation is sufficiently long.*

This proposition implies that the true quality of an individual cannot be learned arbitrarily accurately due to the cumulative advantage effect. Therefore, design becomes much more difficult in the case without cumulative advantage.

In Figure 6.7, we show the impact of cumulative advantage on evaluation system design and the achievable social quality for two values of T. Given the time-to-evaluation T, varying the reputation threshold leads to different social quality. For the same time-to-evaluation T, the system without the cumulative advantage effect requires a higher reputation threshold to achieve the optimal social quality. Importantly, the best achievable social quality is lower then the cumulative advantage is present. This is consistent with Proposition 10 that with cumulative advantage, individuals qualities are more difficult to learn due to the early randomness in reputation accumulation.

6.5.2 Heterogenous types of individuals

The working output of individuals may not only depend on their qualities but also their types. In this subsection, we introduce types of individuals. Let Π be a finite set of types.

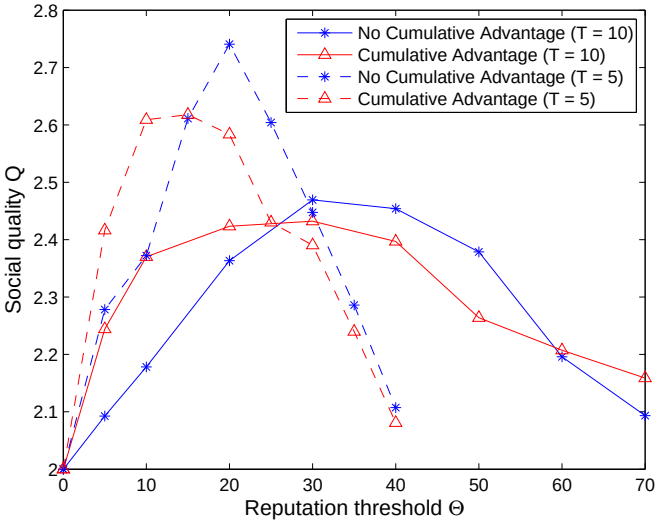


Figure 6.7: Impact of cumulative advantage on social quality.

The value of a collaborative work between type $\pi_i \in \Pi$ and type $\pi_i \in \Pi$ is denoted by γ_{ij}. In a system where collaboration is randomly matched, the intensity at which an individual of any type collaborates with an individual of type j is $\alpha P_j/P$. Therefore, the reputation dynamics of an individual of type i and quality q is as follows:

$$d\theta = [(1-\alpha)q + \alpha \sum_j \frac{P_j}{P} \lambda_{ij} Q_j]dt + \sigma dB_t \tag{6.13}$$

To simplify the analysis, let us assume that all types have the same prior quality distribution and $\lambda_{ii} = \lambda_1, \forall i$ and $\lambda_{ij} = \lambda_2, \forall i \neq j$. Therefore, the value of a collaboration work depends only on whether the collaborating parties have the same types or not. In this case, the reputation dynamics becomes

$$d\theta = [(1-\alpha)q + \frac{\alpha}{P}(\gamma_1 P_i Q_i + \gamma_2 \sum_{j \neq i} P_j Q_j)]dt + \sigma dB_t \tag{6.14}$$

Proposition 18. *If $\gamma_1 < \gamma_2$, then the only possible steady state is symmetric, i.e. $P_i = P_j, Q_i = Q_j, \forall i \neq j$.*

The above proposition implies that if inter-type collaboration is more valuable than intra-type collaboration, then the system allows the most diverse population profile, i.e. there is not a type of individuals that is more dominant than another. However, when $\gamma_1 > \gamma_2$,

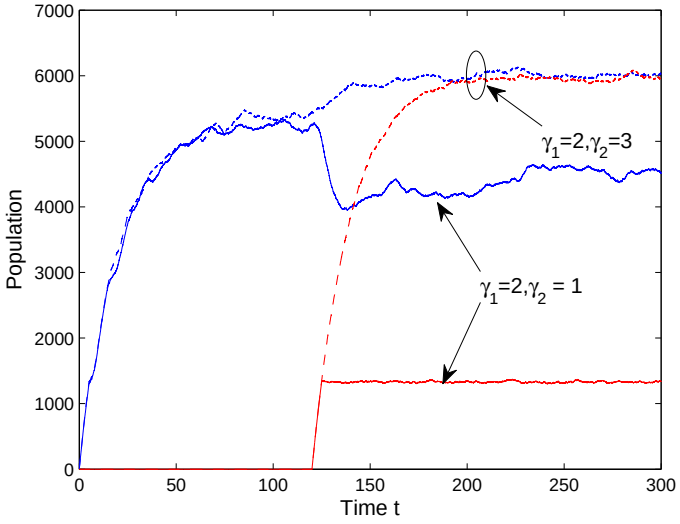


Figure 6.8: Population of different types of individuals.

i.e. intra-type collaboration is more valuable than inter-type collaboration, there could exist a dominant type that has a much larger population than others. In Figure 6.8, we carry out a simulation for two types of individuals and show the effects of different valuation of inter-type and intra-type collaborations. In this simulation, only one type of individuals enter into the system for $t = 0$ to $t = 120$ and the system converges to a steady state. Starting from time $t = 120$, a second type of individuals begin to enter and hence two types of individuals co-exist in the system. Depending on the value of inter-type and intra-type collaborations, the system moves to different steady states. When intra-collaboration is more valuable than inter-type collaboration ($\lambda_1 > \lambda_2$), the first type of individuals who enter the system earlier dominate in terms of population as shown by the solid curves. When intra-type collaboration is less valuable than inter-type collaboration ($\lambda_1 < \lambda_2$), both types of individuals tend to have the same population as shown by the dashed curves. This is consistent with our analysis and implies that a higher valuation of inter-type collaboration promotes a more diverse population.

6.6 Conclusions

In this paper, we developed a population model for the prevailing up-or-out evaluation system that tries to control the number and qualities of individuals in a network. Our model is simple yet is able to produce important insights on many aspects of the system: the optimal design, the role of collaboration on productivity, the effect of cumulative advantage on learning and the co-existence of heterogenous types of individuals. In order to enable tractable analysis from the population perspective, the current model neglects the agent decision component in which the working output depends not only on the qualities of individuals but also their effort levels. Understanding the joint impact of quality and efforts on the performance of the evaluation system is, though much more challenging, a future research topic of significant importance.

6.7 Appendix

Proof of Lemma 1

Suppose the system is already in a steady state. (1) When $\sigma = 0$, the reputation of an individual of quality q grows linear over time and at time t_0+T the reputation is $[(1-\alpha)q+\alpha\gamma Q]T$. By the "up-or-out" rule, only individuals of quality $q \geq q_{th}$ will have reputation higher than or equal to Θ and hence can pass the evaluation.

(2) When $\sigma > 0$, i.e. there is noise in reputation update, the reputation of an individual of quality q at time $t_0 + T$ is

$$\theta(t_0 + T) = [(1 - \alpha)q + \alpha\gamma Q]T + \sigma(B_{t_0+T} - B_{t_0}) \tag{6.15}$$

Since B_t is a Brownian motion, $B_{t_0+T} - B_{t_0}$ follows the normal distribution $\mathcal{N}(0, T)$. Therefore, $\theta(t_0 + T)$ is normally distribution with mean $[(1 - \alpha)q + \alpha\gamma Q]T$ and variance $\sigma^2 T$. Therefore, the probability that this individual of quality q fails the evaluation is

$$\rho_q(Q) = \Phi\left(\frac{\Theta - [(1 - \alpha)q + \alpha\gamma Q]T}{\sqrt{T}\sigma}\right) \tag{6.16}$$

This probability is greater than or equal to 0.5 if $q < q_{th}$.

Proof of Lemma 2

(1) We analyze the population mass of individuals of different qualities. Let $p(q,\tau)$ be the population mass of individuals who have quality q and have stayed in the system for τ time. After a small amount of time dt, $1-e^{-\lambda_d dt}$ fraction of them exit from the system and the remaining $e^{-\lambda_d dt}$ fraction have an increased staying time $t+dt$. Therefore, we have the following relation in steady state

$$p(q,\tau+dt) = e^{-\lambda_d dt} p(q,\tau), \forall \tau \tag{6.17}$$

This leads to the general solution $p(q,\tau) = Ce^{-\lambda_d \tau}$ where C is a constant. Thus $p(q) = \int_{\tau=0}^{\infty} p(q,\tau) = C/\lambda_d$. Since entering and exiting individuals should balance in steady state, $f(q)\lambda_b dt = p(q)(1-e^{-\lambda_d dt})$. As $dt \to 0$, this becomes, $f(q)\lambda_b = p(q)\lambda_d$. Therefore, $C = f(q)\lambda_b$ and $p(q) = \frac{\lambda_b}{\lambda_d} f(q)$. Individuals of $q < q_{th}$ can stay up to T time and hence $p(q,\tau) = 0, \forall \tau > T$. Therefore $p(q) = \int_{\tau=0}^{T} f(q)\lambda_b e^{-\lambda_d \tau} d\tau = \frac{\lambda_b}{\lambda_d} f(q)(1-e^{-\lambda_d T})$. Using $p(q)$, we can compute P and Q as follows

$$P = \int_{q=q_{th}}^{\infty} \frac{\lambda_b}{\lambda_d} f(q) dq + \int_{q=0}^{q_{th}} \frac{\lambda_b}{\lambda_d} f(q)(1-e^{-\lambda_d T}) dq = \frac{\lambda_b}{\lambda_d}(1 - e^{-\lambda_d T} F(q_{th})) \tag{6.18}$$

Similarly, we can derive $Q = \frac{\bar{q} - e^{-\lambda_d T} G(q_{th})}{1 - e^{-\lambda_d T} F(q_{th})}$.

(2) The case with reputation update noise is similar as above. The only difference is that for all q, there is a probability that the individual may be expelled from the system at $\tau = T$ and hence,

$$p(q) = \int_{t=0}^{T} f(q)\lambda_b e^{-\lambda_d \tau} + \int_{t=T}^{\infty} (1-\rho_q) f(q)\lambda_b e^{-\lambda_d \tau} = \frac{\lambda_b}{\lambda_d}(1 - e^{-\lambda_d T}\rho_q) f(q) \tag{6.19}$$

Therefore,

$$P = \int_{q=0}^{\infty} (\frac{\lambda_b}{\lambda_d}(1 - e^{-\lambda_d T}\rho_q) f(q)) = \frac{\lambda_b}{\lambda_d}(1 - e^{-\lambda_d T} \int_{q=0}^{\infty} \rho_q f(q) dq) \tag{6.20}$$

Similarly, we can derive $Q = \frac{\bar{q} - e^{-\lambda_d T} \int_{q=0}^{\infty} \rho_q g(q) dq}{1 - e^{-\lambda_d T} \int_{q=0}^{\infty} \rho_q f(q) dq}$.

Proof of Theorem 1

(1) If $\sigma = 0$, then $q_{th} \geq 0$. Notice that $Q(q_{th})$ is bounded in $[\bar{q}, \frac{\bar{q}}{1-e^{-\lambda_d T}})$, as $q_{th} \to \infty$, the left-hand-side (LHS) is less than the right-hand side (RHS). If $Q(0) = \bar{q} < \frac{\Theta/T}{(\alpha\gamma}$, then when

$q_{th} = 0$, LHS > RHS. Since functions on both sides are continuous in q_{th}, there must be a solution where $q_{th} > 0$. If $\bar{q} < \frac{\Theta/T}{\alpha\gamma}$, then it means that all individuals regardless of their qualities will pass the evaluation. In this case, there is also a steady state where $Q = \bar{q}$.

(2) If $\sigma > 0$, then q_{th} can also take negative values. Again $Q(q_{th})$ is bounded in $[\bar{q}, \frac{\bar{q}}{1-e^{-\lambda_d T}})$. Moreover, as $q_{th} \to \infty$, LHS > RHS; as $q_{th} \to -\infty$, LHS < RHS. Therefore, there must be a solution for LHS = RHS. This proves the existence of a steady state.

Proof of Proposition 1

To show there is a unique solution to (6.5), consider the function $Y(q_{th}) = \frac{\Theta/T-(1-\alpha)q_{th}}{\alpha\gamma} - Q(q_{th})$. If $Y(q_{th})$ is monotone in q_{th}, then there is a unique solution of $Y(q_{th}) = 0$. Thus, we study the derivative of $Y(q_{th})$, which is

$$-\frac{(1-\alpha)}{\alpha\gamma} - e^{-\lambda_d T}\frac{-g(q_{th})(1-e^{-\lambda_d T}F(q_{th}) + (\bar{q} - e^{-\lambda_d T}G(q_{th}))f(q_{th})}{(1-e^{-\lambda_d T}F(q_{th}))^2} \tag{6.21}$$

For q_{th} such that $-g(q_{th})(1-e^{-\lambda_d T}F(q_{th})+(\bar{q}-e^{-\lambda_d T}G(q_{th}))f(q_{th}) \geq$, the derivative is always negative. For q such that $-g(q_{th})(1-e^{-\lambda_d T}F(q_{th}) + (\bar{q} - e^{-\lambda_d T}G(q_{th}))f(q_{th}) < 0$,

$$\frac{g(q_{th})(1-e^{-\lambda_d T}F(q_{th}) - (\bar{q} - e^{-\lambda_d T}G(q_{th}))f(q_{th})}{(1-e^{-\lambda_d T}F(q_{th}))^2} < \frac{g(q_{th})}{(1-e^{-\lambda_T})^2} \tag{6.22}$$

Since $g(q) < \frac{(1-\alpha)(1-e^{-\lambda_d T})^2}{\alpha\gamma e^{-\lambda_d T}}, \forall q$, the derivative of $Y(q_{th})$ is also negative. Thus $Y(q_{th})$ is monotonically decreasing and therefore there is a unique solution to $Y(q_{th}) = 0$.

Proof of Proposition 2

By converting (6.5) we can get a fixed point equation in terms of q_{th}, i.e.

$$q_{th} = \frac{1}{1-\alpha}\left(\frac{\Theta}{T} - \alpha\gamma\frac{\bar{q} - e^{-\lambda_d T}G(q_{th})}{1-e^{-\lambda_d T}F(q_{th})}\right) = \Pi(q_{th}) \tag{6.23}$$

Convergence requires that for any $q_1 > q_2$,

$$|\Pi(q_1) - \Pi(q_2)| < q_1 - q_2 \tag{6.24}$$

This is equivalent to

$$\left|\frac{\bar{q} - e^{-\lambda_d T}G(q_1)}{1-e^{-\lambda_d T}F(q_1)} - \frac{\bar{q} - e^{-\lambda_d T}G(q_2)}{1-e^{-\lambda_d T}F(q_2)}\right| < \frac{(1-\alpha)(q_1-q_2)}{\alpha\gamma} \tag{6.25}$$

The left-hand side is

$$\frac{|e^{-2\lambda_d T}(G(q_1)F(q_2)-G(q_2)F(q_1)))+e^{-\lambda_d T}(\bar{q}F(q_1)-G(q_1)-(\bar{q}F(q_2)-G(q_2))|}{(1-e^{-\lambda_d T}F(q_1))(1-e^{-\lambda_d T}F(q_2))} \tag{6.26}$$

Using the triangle inequality, the left-hand side is less than

$$\frac{(e^{-2\lambda_d T}\beta_1+e^{-\lambda_d T}\beta_2)}{(1-e^{-\lambda_d T})^2}(q_1-q_2) \tag{6.27}$$

Hence, if we have $\frac{\alpha\gamma(e^{-2\lambda_d T}\beta_1+e^{-\lambda_d T}\beta_2)}{(1-\alpha)(1-e^{-\lambda_d T})^2} < 1$, then (6.24) is satisfied. Therefore, the system converges to the steady state.

Proof of Proposition 3

Since the total population satisfies $P > \frac{\lambda_b}{\lambda_b}(1-e^{-\lambda_d T}), \forall q_{th}$, a feasible evaluation T must be less than $-\frac{\log(1-\frac{\lambda_d}{\lambda_b}\bar{P})}{\lambda_d}$.

For the case $\sigma = 0$, it is obvious $q_{th}(T)$ is increasing in T since $e^{-\lambda_d T}$ is decreasing in T and $F(q_{th})$ is increasing in q_{th}.

For the case $\sigma > 0$, for a fixed T, $P(q_{th})$ is always decreasing in q_{th}. For a fixed q_{th}, we study the first-order derivative of $P(T)$,

$$\begin{aligned}\frac{dP(T)}{dT} = e^{-\lambda_d}\int_{q=0}^{\infty} \quad & [\lambda_d\Phi(\tfrac{\gamma(1-\alpha)\sqrt{T}}{\sigma}(q_{th}-q)) \\ & -\phi(\tfrac{\gamma(1-\alpha)\sqrt{T}}{\sigma}(q_{th}-q))\tfrac{\gamma(1-\alpha)}{\sigma}(q_{th}-q)\tfrac{1}{2}T^{-1/2}]f(q)dq\end{aligned} \tag{6.28}$$

Let $h(q) = \lambda_d\Phi(\frac{\gamma(1-\alpha)\sqrt{T}}{\sigma}(q_{th}-q)) - \phi(\frac{\gamma(1-\alpha)\sqrt{T}}{\sigma}(q_{th}-q))\frac{\gamma(1-\alpha)}{\sigma}(q_{th}-q)\frac{1}{2}T^{-1/2}$. For $q > q_{th}, h(q) > 0$. For $q < q_{th}$, since $\Phi(\frac{\gamma(1-\alpha)\sqrt{T}}{\sigma}(q_{th}-q)) > 2\phi(\frac{\gamma(1-\alpha)\sqrt{T}}{\sigma}(q_{th}-q))\frac{\gamma(1-\alpha)\sqrt{T}}{\sigma}(q_{th}-q)$, if $\frac{1}{4\lambda_d T} < 1$, then $h(q) > 0$. Hence if $T > \frac{1}{4\lambda_d}$, then $P(T)$ is increasing in T. Therefore, $q_{th}(T)$ has to be increasing in T for the fixed $\bar{P}$.

Proof of Proposition 4

(1) Consider $\sigma = 0$. To show $W(T)$ is decreasing in T, we only need to show $e^{-\lambda_d T}G(q_{th}(T))$ is increasing in T. Because $e^{-\lambda_d T}F(q_{th}(T)) = 1-\bar{P}$ is a constant, we can alternatively show that $\frac{G(q_{th}(T))}{F(q_{th}(T))}$ is increasing in T. Because $\frac{G(q_{th})}{F(q_{th})}$ is increasing in q_{th} and from Proposition 3 we know that $q_{th}(T)$ is increasing in T, we get the claimed result.

(2) Consider $\sigma > 0$. Using similar arguments as for the case with $\sigma = 0$, we can show that maximizing the social productivity $W(T, q_{th}(T))$ is equivalent to minimize $\frac{\int_{q=0}^{\infty} \rho_q(q_{th}(T)) q f(q) dq}{\int_{q=0}^{\infty} \rho_q(q_{th}(T)) f(q) dq} \triangleq U(T)$. Since $\rho_q(q_{th})$ is a decreasing function of q, it is easy to see that $U(T) \leq q$. We will show that as $T \to 0$, $U(T) \to \bar{q}$. This will lead to the claimed result.

Let $q_{th}(T)$ such that $\Phi(\frac{\gamma(1-\alpha)}{\sigma}\sqrt{T} q_{th}(T))) = 1 - \bar{P}$. As $T \to 0$,

$$P(T) = 1 - e^{-\lambda_d T} \int_{q=0}^{\infty} \Phi(\frac{\gamma(1-\alpha)}{\sigma}\sqrt{T}(q_{th}(T) - q))) f(q) dq \to \bar{P} \tag{6.29}$$

Hence, the population constraint is satisfied. Moreover, $\forall q$, as $T \to 0$, $\Phi(\frac{\gamma(1-\alpha)}{\sigma}\sqrt{T}(q_{th}(T) - q))) \to 1 - \bar{P}$ which is a constant. Therefore, $\lim_{T \to 0} U(T) \to \frac{\int_{q=0}^{\infty} q f(q) dq}{\int_{q=0}^{\infty} f(q) dq} = \bar{q}$.

We have shown that $T \to 0$ leads to the lowest possible social productivity and hence $\lim_{T \to 0} W(T) < W(T^*)$ for some $T^* > 0$.

Proof of Proposition 5

Consider two noise levels $\sigma_1 < \sigma_2$. Fix T and let $q_{th,1}, q_{th,2}$ be the quality thresholds such that the population constraint is met for σ_1, σ_2, respectively. For any quality q, the difference in the probability that the individuals of quality q fail is reflected by

$$\frac{\gamma(1-\alpha)\sqrt{T}}{\sigma_1}(q_{th,1} - q) - \frac{\gamma(1-\alpha)\sqrt{T}}{\sigma_2}(q_{th,2} - q) = \gamma(1-\alpha)\sqrt{T}\frac{q_{th,1}\sigma_2 - q_{th,2}\sigma_1 - (\sigma_2 - \sigma_1)q}{\sigma_1 \sigma_2} \tag{6.30}$$

This leads to $q_{th,1}\sigma_2 - q_{th,2}\sigma_1 > 0$; otherwise all qualities will have an higher probability to pass the evaluation for σ_1 and hence, the total population will be more. Therefore, there is a quality threshold $\hat{q}$ such that for all qualities $q > \hat{q}$, the failure probability is higher for σ_2 than σ_1 and for qualities $q < \hat{q}$, the failure probability is lower for σ_2 than σ_1. This means that the social quality is higher when the noise level is σ_1.

Proof of Proposition 6

(1) It is obvious that $Q(0) = \bar{q}$. As $q_{th} \to \infty$, $\frac{G(q_{th})}{F(q_{th})} \to \bar{q}$. Therefore it is also easy to verify that $\lim_{q_{th} \to \infty} Q(q_{th}) \to \bar{q}$.

(2) The first-order derivative of $Q(q_{th})$ is

$$\frac{dQ(q_{th})}{dq_{th}} = e^{-\lambda_d T} \frac{-g(q_{th})(1 - e^{-\lambda_d T} F(q_{th})) + f(q_{th})(\bar{q} - e^{-\lambda_d T} G(q_{th}))}{(1 - e^{-\lambda_d T} F(q_{th}))^2} \tag{6.31}$$

For the first-order condition $\frac{Q(q_{th})}{q_{th}} = 0$ to hold, we must have $-g(q_{th})(1 - e^{-\lambda_d T} F(q_{th})) + f(q_{th})(\bar{q} - e^{-\lambda_d T} G(q_{th}))$, which can be simplified to

$$L(q_{th}) \triangleq (\bar{q} - q_{th}) - e^{-\lambda_d T}(G(q_{th}) - q_{th} F(q_{th})) = 0 \tag{6.32}$$

If this equation has a unique solution, then there exists a unique maximizer q^*_{th} for $Q(q_{th})$. We prove this below.

When $q_{th} = 0$, we have $L(0) = \bar{q} > 0$. Since $L(q_{th}) < \bar{q} - (1 - e^{-\lambda_d T}) q_{th}$, as $q_{th} \to \infty$, $L(q_{th}) < 0$. Therefore, there must exist at least one solution to $L(q_{th}) = 0$. The first-order derivative of $L(q_{th})$ is always negative, i.e.

$$\frac{dL(q_{th})}{dq_{th}} = -1 + e^{-\lambda_d T} F(q_{th}) < 0 \tag{6.33}$$

Therefore, there is a unique solution of $L(q_{th}) = 0$. Thus, we have proved the existence of q^*_{th}. To show $q^*_{th} > \bar{q}$, simply notice $L(\bar{q}) > 0$. This completes the proof.

Proof of Proposition 7

(1) For a fixed q_{th}, we have

$$\frac{dQ(T)}{dT} = \frac{\lambda_d e^{-\lambda_d T}(G(q_{th}) - \bar{q} F(q_{th})}{(1 - e^{-\lambda_d T} F(q_{th}))^2} < 0 \tag{6.34}$$

and hence, $Q(T|q_{th})$ is decreasing in T. Since $Q^*(T) = \max_{q_{th}} Q(T|q_{th})$, it is obvious that $Q^*(T)$ is also decreasing in T.

(2) For a fixed q_{th}, we have for any q, $\Phi(\frac{\gamma(1-\alpha)}{\sigma}(q_{th} - q)) \to \Phi(0) = 1/2$ which is a constant. Therefore $\lim_{T \to 0} Q(T|q_{th}) \to \bar{q}$. Since this holds for all q_{th}, we get the claimed result.

Proof of Proposition 8

First we note that $Q(q_{th})$ is independent of α when $\sigma = 0$. Hence, only the shape of $H(q_{th}) = \frac{\Theta/T - (1-\alpha) q_{th}}{\alpha \gamma}$ changes with α. Notice there is an invariant point of $H(q_{th})$ which

is $H(\frac{\Theta}{T}) = \frac{\Theta}{\gamma T}$ independent of α. If $Q(\frac{\Theta}{T}) < H(\frac{\Theta}{T})$, then the intersection/steady state if greater than $\frac{\Theta}{T}$. In this case, increasing α decreases the intersection. The other cases can be similarly analyzed.

Proof of Proposition 9

For $\sigma = 0$, notice that $Q(q_{th})$ is independent of α. For $\sigma > 0$, the proof is similar to the proof of Proposition 5 by considering the change in α instead of σ.

Proof of Proposition 10

The solution of the reputation dynamics is

$$\theta_t = \frac{(1-\alpha)q\bar{\theta}}{-\alpha\gamma Q}(1 - e^{\frac{\alpha\gamma Q}{\theta}t}) + e^{\frac{\alpha\gamma Q}{\theta}t}\int_{s=0}^{t} \sigma e^{-\frac{\alpha\gamma Q}{\theta}s} dB_s \tag{6.35}$$

Therefore, the reputation of an individual of quality q at the evaluation time T is normally distributed with mean

$$\theta_{\mathbb{T}} = \frac{(1-\alpha)q\bar{\theta}}{\alpha\gamma Q}(e^{\frac{\alpha\gamma Q}{\theta}T} - 1) \tag{6.36}$$

and variance

$$var(\theta_T) = \frac{\sigma^2\bar{\theta}}{2\alpha\gamma Q}(e^{2\frac{\alpha\gamma Q}{\theta}T} - 1) \tag{6.37}$$

Thus, the probability that the individual fails the evaluation is computed by

$$\Phi\left(\frac{\frac{(1-\alpha)\bar{\theta}}{\alpha\gamma Q}(e^{\frac{\alpha\gamma Q}{\theta}T} - 1)(q_{th} - q)}{\sigma\sqrt{\frac{\bar{\theta}}{2\alpha\gamma Q}(e^{2\frac{\alpha\gamma Q}{\theta}T} - 1)}}\right) \tag{6.38}$$

As $T \to \infty$, $e^{\frac{\alpha\gamma Q}{\theta}T} \gg 1$ and hence the above becomes

$$\Phi\left(\frac{\frac{(1-\alpha)\bar{\theta}}{\alpha\gamma Q}(q_{th} - q)}{\sigma\sqrt{\frac{\bar{\theta}}{2\alpha\gamma Q}}}\right) = \Phi\left(\sqrt{\frac{2\bar{\theta}}{\alpha\gamma Q}}\frac{(1-\alpha)}{\sigma}(q - q_{th})\right) \tag{6.39}$$

Consider an individual of quality q. If there is no evaluation system, then the average reputation that he can obtain is

$$\int_{t=0}^{\infty} e^{-\lambda_d t}((1-\alpha)q + \alpha\gamma Q)t \tag{6.40}$$

which is bounded. Hence $\bar{\theta}$ is bounded as $T \to \infty$. Therefore, the failing probability does not vanish.

Proof of Proposition 11

We prove by contradiction. Suppose there exists two types i and j such that $P_i \neq P_j$. Because the entering rate is the same for all types and all types are evaluated at the same time T, the cut-off qualities of both types must be different, i.e. $q_{th,i} \neq q_{th,j}$. Suppose $q_{th,i} > q_{th,j}$. Because $P_i Q_i = \frac{\lambda_b}{\lambda_d}(\bar{q} - e^{-\lambda_d T} G(q_{th,i})$ and the prior quality distribution is the same for all types, we have $P_i Q_i < P_j Q_j$. Since the same reputation threshold is used for all types, we have

$$\begin{aligned} &(1-\alpha) q_{th,i} + \tfrac{\alpha}{P}(\gamma_1 P_i Q_i + \gamma_2 P_j Q_j + \gamma \sum_{k \neq i,j} P_k Q_k) \\ &= (1-\alpha) q_{th,j} + \tfrac{\alpha}{P}(\gamma_1 P_j Q_j + \gamma_2 P_i Q_i + \gamma \sum_{k \neq i,j} P_k Q_k) \end{aligned} \tag{6.41}$$

This leads to

$$(1-\alpha)(q_{th,i} - q_{th,j}) = \frac{\alpha}{P}(\gamma_1 - \gamma_2)(P_j Q_j - P_i Q_i) \tag{6.42}$$

Because $\gamma_1 < \gamma_2$, the right-hand side of the above equation is negative. Therefore $q_{th,i} < q_{th,j}$ which leads to a contradiction. Therefore, $q_{th,i} = q_{th,j}, \forall i, j$ and hence $P_i = P_j, Q_i = Q_j, \forall i, j$.

CHAPTER 7

Concluding Remarks

In this dissertation, I have studied three important problems for the efficient and robust operation of networks consisting of strategic agents: how to incentivize agents to take socially optimal actions, how do agents learn efficiently in networks and how to design social norm mechanisms to perform adverse selection in networks. I have provided systematic solutions to solve all the above problems and have demonstrated significant performance gains over existing solutions.

In the first part of this dissertation, I developed two incentive mechanisms depending on the specific network interaction environments and rigorously analyzed their equilibrium behavior and performance. The first is based on the exchange of tokens/fiat money and the second is based on reputation/rating systems. Both mechanisms are developed in the framework of the repeated game theory but with many innovations. In the second part, I developed two online learning algorithms for agents to efficiently extract knowledge from real-time information flows and make decisions aimed at maximizing system performance. The first is a multi-agent learning algorithm and the second is a contextual learning algorithm both of which are developed using the theory of multi-armed bandits. These algorithms are simple to implement, yet can adapt to the dynamically changing environments and most importantly, their performance can be rigorously proved. In the final part, I studied the design and dynamics of the "up-or-out" evaluation system using a population model. It provides theoretical supports for the current practice of "up-or-out" evaluation adopted in many professions and informs the optimal design of such a mechanism. Moreover, I also show the role of collaboration among agents on the achievable productivity of the network.

Several future directions are mentioned here. First, the problems studied have focused

on certain features of the system while making simplifying assumptions on the others. It is of great importance to study more general settings. Second, other dimensions of dynamics and heterogeneity, such as types of agents and changing rates of interaction, can make the model more realistic, though more challenging. Third, networks in this dissertation have been treated mostly as given. However, with strategic agents, networks can evolve endogenously as agents learn and make decisions on forming new links and severing existing links between agents. This may further complicated the problem but also promises many research opportunities. This dissertation does not solve all problems in networks of strategic agents. However, I hope that it makes important steps towards better understanding of and designing such networks.

References

[AB02] Réka Albert and Albert-László Barabási. "Statistical mechanics of complex networks." *Reviews of modern physics*, **74**(1):47, 2002.

[ABB11] Giuseppe Amodeo, Roi Blanco, and Ulf Brefeld. "Hybrid models for future event prediction." In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1981–1984. ACM, 2011.

[ACF02] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. "Finite-time analysis of the multiarmed bandit problem." *Machine learning*, **47**(2-3):235–256, 2002.

[AH00] Eytan Adar and Bernardo A Huberman. "Free riding on gnutella." *First Monday*, **5**(10), 2000.

[AH10] Sitaram Asur and Bernardo A Huberman. "Predicting the future with social media." In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pp. 492–499. IEEE, 2010.

[Alo99] Carlos Alós-Ferrer. "Dynamical systems with a continuum of randomly matched agents." *Journal of Economic Theory*, **86**(2):245–267, 1999.

[AMT11] Animashree Anandkumar, Nithin Michael, Ao Kevin Tang, and Ananthram Swami. "Distributed algorithms for learning and cognitive medium access with logarithmic regret." *Selected Areas in Communications, IEEE Journal on*, **29**(4):731–745, 2011.

[Aue03] Peter Auer. "Using confidence bounds for exploitation-exploration trade-offs." *The Journal of Machine Learning Research*, **3**:397–422, 2003.

[AVW87] Venkatachalam Anantharam, Pravin Varaiya, and Jean Walrand. "Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-Part I: IID rewards." *Automatic Control, IEEE Transactions on*, **32**(11):968–976, 1987.

[Axe81] Robert Axelrod. "The emergence of cooperation among egoists." *American Political Science Review*, **75**(02):306–318, 1981.

[BC04] Barry Bozeman and Elizabeth Corley. "Scientists collaboration strategies: implications for scientific and technical human capital." *Research Policy*, **33**(4):599–616, 2004.

[BCW07] Aleksander Berentsen, Gabriele Camera, and Christopher Waller. "Money, credit and banking." *Journal of Economic Theory*, **135**(1):171–195, 2007.

[Ber00] Aleksander Berentsen. "Money inventories in search equilibrium." *Journal of money, credit and banking*, pp. 168–178, 2000.

[Ber02] Aleksander Berentsen. "On the Distribution of Money Holdings in a Random-Matching Model*." *International Economic Review*, **43**(3):945–954, 2002.

[BF10] Luciano Barbosa and Junlan Feng. "Robust sentiment detection on twitter from biased and noisy data." In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 36–44. Association for Computational Linguistics, 2010.

[BH01] Levente Buttyan and Jean-Pierre Hubaux. "Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks." Technical report, EPFL, 2001.

[BH03] Levente Buttyán and Jean-Pierre Hubaux. "Stimulating cooperation in self-organizing mobile ad hoc networks." *Mobile Networks and Applications*, **8**(5):579–592, 2003.

[BO06] Dirk Bergemann and Deran Ozmen. "Optimal pricing with recommender systems." In *Proceedings of the 7th ACM conference on Electronic commerce*, pp. 43–51. ACM, 2006.

[BP02] Sulin Ba and Paul A Pavlou. "Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior." *MIS quarterly*, pp. 243–268, 2002.

[BSH13] Peng Bao, Hua-Wei Shen, Junming Huang, and Xue-Qi Cheng. "Popularity prediction in microblogging network: a case study on sina weibo." In *Proceedings of the 22nd international conference on World Wide Web companion*, pp. 177–178. International World Wide Web Conferences Steering Committee, 2013.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[CC99] Gabriele Camera and Dean Corbae. "Money and price dispersion." *International Economic Review*, **40**(4):985–1008, 1999.

[CHB09] Wen-Ying Sylvia Chou, Yvonne M Hunt, Ellen Burke Beckjord, Richard P Moser, and Bradford W Hesse. "Social media use in the United States: implications for health communication." *Journal of medical Internet research*, **11**(4), 2009.

[Ciu10] Augusto Ciuffoletti. "Secure token passing at application level." *Future Generation Computer Systems*, **26**(7):1026–1031, 2010.

[CKR07] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system." In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 1–14. ACM, 2007.

[CL12] Nicolo Cesa-Bianchi and Gábor Lugosi. "Combinatorial bandits." *Journal of Computer and System Sciences*, **78**(5):1404–1422, 2012.

[CLR01] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, et al. *Introduction to algorithms*, volume 2. MIT press Cambridge, 2001.

[CLR11] Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. "Contextual bandits with linear payoff functions." In *International Conference on Artificial Intelligence and Statistics*, pp. 208–214, 2011.

[CV12] Hyunyoung Choi and Hal Varian. "Predicting the present with google trends." *Economic Record*, **88**(s1):2–9, 2012.

[CW99a] Ricardo de O Cavalcanti and Neil Wallace. "Inside and outside money as alternative media of exchange." *Journal of Money, Credit and Banking*, pp. 443–457, 1999.

[CW99b] Ricardo de O Cavalcanti and Neil Wallace. "A model of private bank-note issue." *Review of Economic Dynamics*, **2**(1):104–136, 1999.

[CWY13] Wei Chen, Yajun Wang, and Yang Yuan. "Combinatorial multi-armed bandit: General framework and applications." In *Proceedings of the 30th International Conference on Machine Learning*, pp. 151–159, 2013.

[Del05] Chrysanthos Dellarocas. "Reputation mechanism design in online trading environments with pure moral hazard." *Information Systems Research*, **16**(2):209–230, 2005.

[DHK08] Varsha Dani, Thomas P Hayes, and Sham M Kakade. "Stochastic Linear Optimization under Bandit Feedback." In *COLT*, pp. 355–366, 2008.

[DHK11] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. "Efficient optimal learning for contextual bandits." *arXiv preprint arXiv:1106.2369*, 2011.

[DS77] Avinash K Dixit and Joseph E Stiglitz. "Monopolistic competition and optimum product diversity." *The American Economic Review*, pp. 297–308, 1977.

[DS07] Darrell Duffie and Yeneng Sun. "Existence of independent random matching." *The Annals of Applied Probability*, pp. 386–419, 2007.

[DS13] Raphaël Ducasse and Mihaela van der Schaar. "Finding it now: Construction and configuration of networked classifiers in real-time stream mining systems." In *Handbook of Signal Processing Systems*, pp. 97–134. Springer, 2013.

[DTS10] Raphael Ducasse, Deepak S Turaga, and Mihaela van der Schaar. "Adaptive topologic optimization for large-scale stream mining." *Selected Topics in Signal Processing, IEEE Journal of*, **4**(3):620–636, 2010.

[EG13] Matthew Elliott and Benjamin Golub. "A network approach to public goods." In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pp. 377–378. ACM, 2013.

[Ell00] Glenn Ellison. "Basins of attraction, long-run stochastic stability, and the speed of step-by-step evolution." *The Review of Economic Studies*, **67**(1):17–45, 2000.

[FHK06] Eric J Friedman, Joseph Y Halpern, and Ian Kash. "Efficiency and Nash equilibria in a scrip system for P2P networks." In *Proceedings of the 7th ACM conference on Electronic commerce*, pp. 140–149. ACM, 2006.

[FS10] Brian Foo and Mihaela van der Schaar. "A distributed approach for optimizing cascaded classifier topologies in real-time stream mining systems." *Image Processing, IEEE Transactions on*, **19**(11):3035–3048, 2010.

[FST04] Daniel R Figueiredo, Jonathan K Shapiro, and Don Towsley. "Payment-based incentives for anonymous peer-to-peer systems." *Computer Science Department, University of Massachusetts*, 2004.

[FTV07] Fangwen Fu, Deepak S Turaga, Olivier Verscheure, Mihaela van der Schaar, and Lisa Amini. "Configuring competing classifier chains in distributed stream mining systems." *Selected Topics in Signal Processing, IEEE Journal of*, **1**(4):548–563, 2007.

[FTW05] Ming Fan, Yong Tan, and Andrew B Whinston. "Evaluation and design of online cooperative feedback mechanisms for reputation management." *Knowledge and Data Engineering, IEEE Transactions on*, **17**(2):244–254, 2005.

[FV09] Brian Foo and Mihaela Van Der Schaar. "A rules-based approach for configuring chains of classifiers in real-time stream mining systems." *EURASIP Journal on Advances in Signal Processing*, **2009**:40, 2009.

[GAC10] Wojciech Galuba, Karl Aberer, Dipanjan Chakraborty, Zoran Despotovic, and Wolfgang Kellerer. "Outtweeting the twitterers-predicting information cascades in microblogs." In *Proceedings of the 3rd conference on Online social networks*, pp. 3–3, 2010.

[GCM11] GONCA Gursun, Mark Crovella, and Ibrahim Matta. "Describing and forecasting video access patterns." In *INFOCOM, 2011 Proceedings IEEE*, pp. 16–20. IEEE, 2011.

[GDL06] Michel Grossetti, Marie-Laure Djelic, and Emmanuel Lazega. "The Collegial phenomenon. The social mechanisms of cooperation among peers in a corporate law partnership." *Sociologie du travail*, **48**:88–109, 2006.

[Git79] John C Gittins. "Bandit processes and dynamic allocation indices." *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 148–177, 1979.

[GKJ12] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations." *IEEE/ACM Transactions on Networking (TON)*, **20**(5):1466–1478, 2012.

[HDD11] Liangjie Hong, Ovidiu Dan, and Brian D Davison. "Predicting popular messages in twitter." In *Proceedings of the 20th international conference companion on World wide web*, pp. 57–58. ACM, 2011.

[HKW09] Tai-wei Hu, John Kennan, and Neil Wallace. "Coalition-Proof Trade and the Friedman Rule in the Lagos-Wright Model." *Journal of Political Economy*, **117**(1), 2009.

[JBC13] Yu-Gang Jiang, Subhabrata Bhattacharya, Shih-Fu Chang, and Mubarak Shah. "High-level event recognition in unconstrained videos." *International Journal of Multimedia Information Retrieval*, **2**(2):73–101, 2013.

[JNB03] Hawoong Jeong, Zoltan Néda, and Albert-László Barabási. "Measuring preferential attachment in evolving networks." *EPL (Europhysics Letters)*, **61**(4):567, 2003.

[JRT12] Matthew O Jackson, Tomas Rodriguez-Barraquer, and Xu Tan. "Social capital and social quilts: Network patterns of favor exchange." *The American Economic Review*, **102**(5):1857–1897, 2012.

[Kan92] Michihiro Kandori. "Social norms and community enforcement." *The Review of Economic Studies*, **59**(1):63–80, 1992.

[KFH07] Ian A Kash, Eric J Friedman, and Joseph Y Halpern. "Optimizing scrip systems: Efficiency, crashes, hoarders, and altruists." In *Proceedings of the 8th ACM conference on Electronic commerce*, pp. 305–315. ACM, 2007.

[KH88] Charles Kahn and Gur Huberman. "Two-sided uncertainty and" up-or-out" contracts." *Journal of Labor Economics*, pp. 423–444, 1988.

[KMG12] Farshad Kooti, Winter A Mason, Krishna P Gummadi, and Meeyoung Cha. "Predicting emerging social conventions in online social networks." In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 445–454. ACM, 2012.

[Koc02] Narayana Kocherlakota. "The two-money theorem." *International Economic Review*, pp. 333–346, 2002.

[KP13] Vikram Krishnamurthy and H Vincent Poor. "Social learning and Bayesian games in multiagent signal processing: How do local and global decision makers interact?" *Signal Processing Magazine, IEEE*, **30**(3):43–57, 2013.

[Kri12] Vikram Krishnamurthy. "Quickest detection POMDPs with social learning: Interaction of local and global decision makers." *Information Theory, IEEE Transactions on*, **58**(8):5563–5587, 2012.

[KSG03] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. "The eigentrust algorithm for reputation management in p2p networks." In *Proceedings of the 12th international conference on World Wide Web*, pp. 640–651. ACM, 2003.

[KW89] Nobuhiro Kiyotaki and Randall Wright. "On money as a medium of exchange." *The Journal of Political Economy*, pp. 927–954, 1989.

[LA98] Réjean Landry and Nabil Amara. "The impact of transaction costs on the institutional structuration of collaborative academic research." *Research Policy*, **27**(9):901–913, 1998.

[LB05] Sooho Lee and Barry Bozeman. "The impact of research collaboration on scientific productivity." *Social studies of science*, **35**(5):673–702, 2005.

[LFP08] Lifeng Lai, Yijia Fan, and H Vincent Poor. "Quickest detection in cognitive radio: A sequential change detection framework." In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pp. 1–5. IEEE, 2008.

[LH10] Kristina Lerman and Tad Hogg. "Using a model of social dynamics to predict popularity of news." In *Proceedings of the 19th international conference on World wide web*, pp. 621–630. ACM, 2010.

[LLX12] Haitao Li, Jiangchuan Liu, Ke Xu, and Song Wen. "Understanding video propagation in online social networks." In *Proceedings of the 2012 IEEE 20th International Workshop on Quality of Service*, p. 21. IEEE Press, 2012.

[LLZ13] Haoyang Liu, Keqin Liu, and Qing Zhao. "Learning in a changing world: Restless multiarmed bandit with unknown dynamics." *Information Theory, IEEE Transactions on*, **59**(3):1902–1916, 2013.

[LMS10] Jong Gun Lee, Sue Moon, and Kavé Salamatian. "An approach to model and predict the popularity of online contents with explanatory factors." In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pp. 623–630. IEEE, 2010.

[LMW13] Haitao Li, Xiaoqiang Ma, Feng Wang, Jiangchuan Liu, and Ke Xu. "On popularity prediction of videos shared in online social networks." In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pp. 169–178. ACM, 2013.

[LR85] Tze Leung Lai and Herbert Robbins. "Asymptotically efficient adaptive allocation rules." *Advances in applied mathematics*, **6**(1):4–22, 1985.

[LW94] Nick Littlestone and Manfred K Warmuth. "The weighted majority algorithm." *Information and computation*, **108**(2):212–261, 1994.

[LWL12] Haitao Li, Haiyang Wang, Jiangchuan Liu, and Ke Xu. "Video sharing in online social networks: measurement and analysis." In *Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video*, pp. 83–88. ACM, 2012.

[LWY12] Hongqiang Harry Liu, Ye Wang, Yang Richard Yang, Hao Wang, and Chen Tian. "Optimizing cost and performance for content multihoming." In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pp. 371–382. ACM, 2012.

[LZ08] John Langford and Tong Zhang. "The epoch-greedy algorithm for multi-armed bandits with side information." In *Advances in neural information processing systems*, pp. 817–824, 2008.

[MA09] David A Miller and S Nageeb Ali. "Enforcing Cooperation in Networked Societies." In *2009 Meeting Papers*. Society for Economic Dynamics, 2009.

[Mer68] Robert K Merton. "The Matthew effect in science." *Science*, **159**(3810):56–63, 1968.

[MJS06] Fabio Milan, Juan José Jaramillo, and R Srikant. "Achieving cooperation in multihop wireless networks of selfish nodes." In *Proceeding from the 2006 workshop on Game theory for communications and networks*, p. 3. ACM, 2006.

[MPX13] Nicholas Mastronarde, Viral Patel, Jie Xu, and Mihaela van der Schaar. "Learning relaying strategies in cellular D2D networks with token-based incentives." In *Globecom Workshops (GC Wkshps), 2013 IEEE*, pp. 163–169. IEEE, 2013.

[MS06] George J Mailath and Larry Samuelson. *Repeated games and reputations*, volume 2. Oxford university press Oxford, 2006.

[MV95] Jeffrey K. MacKie-Mason and Hal R. Varian. "Pricing congestible network resources." *Selected Areas in Communications, IEEE Journal on*, **13**(7):1141–1149, 1995.

[NLL11] Di Niu, Zimu Liu, Baochun Li, and Shuqiao Zhao. "Demand forecast and performance prediction in peer-assisted on-demand streaming systems." In *INFOCOM, 2011 Proceedings IEEE*, pp. 421–425. IEEE, 2011.

[OS92] Brendan O'Flaherty and Aloysius Siow. "On the job screening, up or out rules, and firm growth." *Canadian Journal of Economics*, pp. 346–368, 1992.

[Ost08] Elinor Ostrom. "Tragedy of the commons." *The New Palgrave Dictionary of Economics*, pp. 360–362, 2008.

[PAG13] Henrique Pinto, Jussara M Almeida, and Marcos A Goncalves. "Using early view patterns to predict the popularity of youtube videos." In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 365–374. ACM, 2013.

[PC07] Daniel P Palomar and Mung Chiang. "Alternative distributed algorithms for network utility maximization: Framework and applications." *Automatic Control, IEEE Transactions on*, **52**(12):2254–2269, 2007.

[PH09] H Vincent Poor and Olympia Hadjiliadis. *Quickest detection*, volume 40. Cambridge University Press Cambridge, 2009.

[PM06] Vinay Pai and Alexander E Mohr. "Improving robustness of peer-to-peer streaming with incentives." In *Workshop on the Economics of Networks, Systems and Computation*, 2006.

[Pod10] Konrad Podczeck. "On existence of rich Fubini extensions." *Economic Theory*, **45**(1-2):1–22, 2010.

[PP12] Konrad Podczeck and Daniela Puzzello. "Independent random matching." *Economic Theory*, **50**(1):1–29, 2012.

[PS10] Jaeok Park and Mihaela van der Schaar. "A game theoretic analysis of incentives in content production and sharing over peer-to-peer networks." *Selected Topics in Signal Processing, IEEE Journal of*, **4**(4):704–717, 2010.

[Ria95] Ahmed Riahi-Belkaoui. *The cultural shaping of accounting*. Greenwood Publishing Group, 1995.

[RMZ13] Suman Deb Roy, Tao Mei, Wenjun Zeng, and Shipeng Li. "Towards cross-domain learning for social video popularity prediction." *Multimedia, IEEE Transactions on*, **15**(6):1255–1267, 2013.

[Row11] Matthew Rowe. "Forecasting audience increase on youtube." In *International Workshop on User Profile Data on the Social Semantic Web*, 2011.

[RT10] Paat Rusmevichientong and John N Tsitsiklis. "Linearly parameterized bandits." *Mathematics of Operations Research*, **35**(2):395–411, 2010.

[RTL93] Bernard Rostker, Harry Thie, James Lacy, Jennifer Kawata, and Susanna Purnell. "The Defense Oficer Personnel ManagementAct of 1980: A Retrospective Assessment, Santa Monica, Calif.: RAND." Technical report, R-4246-PMP, 1993.

[RZ00] Raghuram G Rajan and Luigi Zingales. "The firm as a dedicated hierarchy: a theory of the origin and growth of firms." Technical report, National bureau of economic research, 2000.

[RZ02] Paul Resnick and Richard Zeckhauser. "Trust among strangers in internet transactions: Empirical analysis of ebays reputation system." *The Economics of the Internet and E-commerce*, **11**(2):23–25, 2002.

[SB98] Peter Sollich and David Barber. "Online learning from finite training sets and robustness to input bias." *Neural computation*, **10**(8):2201–2217, 1998.

[SBH13] Balazs Szorenyi, Róbert Busa-Fekete, István Hegedüs, Róbert Ormándi, Márk Jelasity, and Balázs Kégl. "Gossip-based distributed stochastic bandit algorithms." In *30th International Conference on Machine Learning (ICML 2013)*, volume 28, pp. 19–27. Acm Press, 2013.

[SCN10] Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. "How useful are your comments?: analyzing and predicting youtube comments and comment ratings." In *Proceedings of the 19th international conference on World wide web*, pp. 891–900. ACM, 2010.

[SH10] Gabor Szabo and Bernardo A Huberman. "Predicting the popularity of online content." *Communications of the ACM*, **53**(8):80–88, 2010.

[SHC03] Mehul A Shah, Joseph M Hellerstein, Sirish Chandrasekaran, and Michael J Franklin. "Flux: An adaptive partitioning operator for continuous query systems." In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pp. 25–36. IEEE, 2003.

[Sio98] Aloysius Siow. "Tenure and other unusual personnel practices in academia." *Journal of Law, Economics and Organization*, **14**(1):152–73, 1998.

[Sli14] Aleksandrs Slivkins. "Contextual bandits with similarity information." *The Journal of Machine Learning Research*, **15**(1):2533–2568, 2014.

[SMK13] Panagiotis Sidiropoulos, Vasileios Mezaris, and Ioannis Kompatsiaris. "Enhancing video concept detection with the use of tomographs." In *ICIP*, pp. 3991–3995, 2013.

[SOM10] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. "Earthquake shakes Twitter users: real-time event detection by social sensors." In *Proceedings of the 19th international conference on World wide web*, pp. 851–860. ACM, 2010.

[SV09] Yi Su and Mihaela Van Der Schaar. "Conjectural equilibrium in multiuser power control games." *Signal Processing, IEEE Transactions on*, **57**(9):3638–3650, 2009.

[SXS15] C. Shen, J. Xu, and M. van der Schaar. "Silence is Gold: Strategic Interference Mitigation Using Tokens in Heterogeneous Small Cell Networks." *Selected Areas in Communications, IEEE Journal on*, **PP**(99):1–1, 2015.

[SYK11] David A Shamma, Jude Yew, Lyndon Kennedy, and Elizabeth F Churchill. "Viral Actions: Predicting Video View Counts Using Synchronous Sharing Behaviors." In *ICWSM*, 2011.

[SZ09] Chengqi Song and Qian Zhang. "Achieving cooperative spectrum sensing in wireless cognitive radio networks." *ACM SIGMOBILE Mobile Computing and Communications Review*, **13**(2):14–25, 2009.

[TJ06] Guang Tan and Stephen A Jarvis. "A payment-based incentive and service differentiation mechanism for peer-to-peer streaming broadcast." In *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*, pp. 41–50. IEEE, 2006.

[TL12] Cem Tekin and Mingyan Liu. "Online learning of rested and restless bandits." *Information Theory, IEEE Transactions on*, **58**(8):5588–5611, 2012.

[Tro12] Daniel Trottier. "Social media as surveillance." *Farnham: Ashgate*, 2012.

[TZS14] Cem Tekin, Simpson Zhang, and Mihaela van der Schaar. "Distributed online learning in social recommender systems." *Selected Topics in Signal Processing, IEEE Journal of*, **8**(4):638–652, 2014.

[VCS03] Vivek Vishnumurthy, Sangeeth Chandrakumar, and Emin Gun Sirer. "Karma: A secure economic framework for peer-to-peer resource sharing." In *Workshop on Economics of Peer-to-Peer Systems*, volume 35, 2003.

[Wal90] Michael Waldman. "Up-or-out contracts: A signaling perspective." *Journal of Labor Economics*, pp. 230–250, 1990.

[Wal05] Rosemary L Walker. "Empirical analysis of up-or-out rules for promotion policies." *Journal of Economics and Finance*, **29**(2):172–186, 2005.

[Whi80] Peter Whittle. "Multi-armed bandits and the Gittins index." *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 143–149, 1980.

[WSW12] Zhi Wang, Lifeng Sun, Chuan Wu, and Shiqiang Yang. "Guiding internet-scale video service deployment using microblog-based prediction." In *INFOCOM, 2012 Proceedings IEEE*, pp. 2901–2905. IEEE, 2012.

[WTV10] Tingyao Wu, Michael Timmers, Danny De Vleeschauwer, and Werner Van Leekwijck. "On the use of reservoir computing in popularity prediction." In *Evolving Internet (INTERNET), 2010 Second International Conference on*, pp. 19–24. IEEE, 2010.

[XS12] Jie Xu and Mihaela van der Schaar. "Social norm design for information exchange systems with limited observations." *Selected Areas in Communications, IEEE Journal on*, **30**(11):2126–2135, 2012.

[XS13] Jie Xu and Mihaela van der Schaar. "Token system design for autonomic wireless relay networks." *Communications, IEEE Transactions on*, **61**(7):2924–2935, 2013.

[XS14] Jie Xu and Mihaela van der Schaar. "Incentive design for heterogeneous user-generated content networks." *ACM SIGMETRICS Performance Evaluation Review*, **41**(4):34–37, 2014.

[XSS14] Jie Xu, Yangbo Song, and Mihaela van der Schaar. "Incentivizing information sharing in networks." In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 5467–5471. IEEE, 2014.

[XTZ15] Jie Xu, Cem Tekin, Simpson Zhang, and Mihaela van der Schaar. "Distributed Multi-Agent Online Learning Based on Global Feedback." *Signal Processing, IEEE Transactions on*, **63**(9):2225–2238, 2015.

[XZS13] Jie Xu, Yu Zhang, and Mihaela van der Schaar. "Rating systems for enhanced cyber-security investments." In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 2915–2919. IEEE, 2013.

[XZV12] Jie Xu, William Zame, and Mihaela Van Der Schaar. "Token economy for online exchange systems." In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pp. 1283–1284. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[YCK11] Bei Yu, Miao Chen, and Linchi Kwok. "Toward predicting popularity of social marketing messages." In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pp. 317–324. Springer, 2011.

[YSS13] Chung-Kai Yu, Mihaela van der Schaar, and Ali H Sayed. "Distributed spectrum sensing in the presence of selfish users." *Proc. IEEE CAMSAP, Saint Martin*, pp. 392–395, 2013.

[YTL11] Rui Yan, Jie Tang, Xiaobing Liu, Dongdong Shan, and Xiaoming Li. "Citation count prediction: learning to estimate future citations for literature." In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1247–1252. ACM, 2011.

[ZB92] Terry L Zivney and William J Bertin. "Publish or perish: What the competition is really doing." *The Journal of Finance*, **47**(1):295–329, 1992.

[Zho99] Ruilin Zhou. "Individual and Aggregate Real Balances in a Random-Matching Model." *International Economic Review*, **40**(4):1009–1038, 1999.

[ZM09] Tao Zhu and Eliot Maenner. "Noncash payment methods in a cashless economy." Technical report, Working paper, 2009.

[ZMM00] Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. "Collaborative reputation mechanisms for electronic marketplaces." *Decision Support Systems*, **29**(4):371–388, 2000.

[ZPS14] Yu Zhang, Jaeok Park, and Mihaela van der Schaar. "Rating Protocols in Online Communities." *ACM Transactions on Economics and Computation*, **2**(1):4, 2014.

[ZWL14] Lei Zhang, Feng Wang, and Jiangchuan Liu. "Understand instant video clip sharing on mobile platforms: Twitter's vine as a case study." In *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, p. 85. ACM, 2014.