Multi-Agent Control in Sociotechnical Systems
Coordinating Intelligent Individuals with Soft Regulation

Yu Luo

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2017

ABSTRACT

Multi-Agent Control in Sociotechnical Systems

Yu Luo

Process control is essential in chemical engineering and has diverse applications in automation, manufacturing, scheduling, etc. In this cross-disciplinary work, we shift the domain focus from the control of machines to the control of multiple intelligent agents. Our goal is to improve the optimization problem-solving process, such as optimal regulation of emerging technologies, in a multi-agent system. Achieving that improvement would have potential value both within and outside the chemical engineering community. This work also illustrates the possibility of applying process systems engineering techniques, especially process control, beyond chemical plants.

It is very common to observe crowds of individuals solving similar problems with similar information in a largely independent manner. We argue here that the crowds can become more efficient and robust problem-solvers, by partially following the average opinion. This observation runs counter to the widely accepted claim that the wisdom of crowds deteriorates with social influence. The key difference is that individuals are self-interested and hence will reject feedbacks that do not improve their performance. We propose a multi-agent control-theoretic methodology — soft regulation — to model the collective dynamics and compute the degree of social influence, i.e., the level to which one accepts the population feedback, that optimizes the problem-solving performance.

Soft regulation is a modeling language for multi-agent sociotechnical systems. The state-space formulation captures the individual learning process (i.e., open loop dynamics) as well as the influence of the population feedback in a straightforward manner. It can model a diverse set of existing multi-agent dynamics. Through numerical analysis and linear algebra, we attempt to understand the role of feedback in multi-agent collective dynamics, thus achieving multi-agent control in sociotechnical systems.

Our analysis through mathematical proofs, simulations, and a human subject experiment suggests that intelligent individuals, solving the same problem (or similar problems), could do much better by adaptively adjusting their decisions towards the population average. We even discover that the crowd of human subjects could self-organize into a near-optimal setting. This discovery suggests a new coordination mechanism for enhancing individual decision-making. Potential applications include mobile health, urban planning, and policy-making.

Table of Contents

List of Theorems and Definitions **iv**

List of Figures **vi**

List of Tables **viii**

Notation **ix**

1 Multi-Agent Collective Dynamics and Control **1**

1.1 Making Crowds "Smarter" 2

1.1.1 Chemical Engineering Beyond Plants 2

1.1.2 Can We Make Crowds "Smarter"? 6

1.1.3 Existing Solutions 7

1.2 Collective Intelligence 9

1.2.1 Wisdom of Experts v. Wisdom of Learners 9

1.2.2 Applications of Collective Intelligence 12

1.2.3 Modeling Collective Dynamics 15

1.3 Multi-Agent Control in Sociotechnical Systems 18

2 Soft Regulation: A Multi-Agent Systems Toolset **23**

2.1 Etymology: Hard Regulation v. Soft Regulation 24

2.1.1 Regulating Emerging Industries 24

2.1.2 A Stylized Model 29

2.2 Taxonomy: A Multi-Agent Collective Dynamics and Control Framework . . 31

2.2.1 A General State-Space Model 31
2.2.2 Soft Regulation as a Toolset for Multi-Agent Systems 33

3 Mathematical Properties of Soft Regulation **37**
3.1 Convergence 38
3.1.1 Mathematical Preliminaries 38
3.1.2 Convergence of Noiseless Soft Regulation 40
3.2 Robustness 42
3.3 Efficiency 42
3.4 Optimal Degree of Social Influence 45
3.4.1 Maximum Contraction 45
3.4.2 Maximum Noise Reduction 46
3.5 Different Solutions 49

4 Implementing Soft Regulation **51**
4.1 Regulating Emerging Industries 53
4.1.1 Background 53
4.1.2 Agent-Based Simulation 53
4.1.3 Discussion 61
4.2 Mobile Health 65
4.2.1 Background 65
4.2.2 Experimental Design 65
4.2.3 Results and Discussion 67
4.3 Local Regulation and Policy: U.S. State Tax and Expenditure Case Study . 76
4.3.1 Background 76
4.3.2 Results and Discussion 76
4.4 Multi-Product Revenue Management 86
4.4.1 Background 86
4.4.2 Model 86
4.4.3 Multi-Product Revenue Optimization Algorithm 88
4.4.4 Results and Discussion 90

5 Conclusion **94**

Bibliography **96**

Appendix A Agent-Based Simulation with NetLogo **104**

A.1 NetLogo Simulation Environment 105

A.2 Script “SR.nlogo” 106

Appendix B Amazon Mechanical Turk Experiment **112**

B.1 Institutional Review Board 113

B.2 Amazon Mechanical Turk 116

Appendix C Online Experiment with Google Apps Script **117**

C.1 Google Apps Script 118

C.2 Script “Code.gs” 119

C.3 Script “Engine.gs” 133

Appendix D Data Analysis with MATLAB **138**

D.1 Script “readme.m” 139

D.2 Script “meanValue.m” 142

D.3 Script “sysid.m” 149

List of Theorems and Definitions

1.1 Definition (problem statement) 6

1.2 Definition (action $z_i(t)$) 7

1.3 Definition (solution θ^*) 7

1.4 Assumption (convergence of learning) 7

1.5 Definition (degree of social influence β_i) 8

2.1 Definition (generalized soft regulation) 31

2.2 Definition (state $x_i(t)$ of the i-th agent) 31

2.3 Definition (payoff function $f_i(\cdot)$ and noise $\nu_i(t)$) 31

2.4 Definition (learning function $g_i(\cdot)$ and noise $\omega_i(t)$) 31

2.5 Assumption (regularity of the learning function) 32

2.6 Definition (learning gain $\tilde{g}'_i$) 32

2.7 Definition (learning spectrum $\mathbf{g}'$ and its width σ_g) 32

2.8 Definition (social network S) 32

2.9 Assumption (stochasticity of social network) 33

2.10 Definition (soft feedback $u_i(t)$) 33

2.11 Definition (condensed soft regulation given identical solution) 33

3.1 Definition (linear iterative system) 38

3.2 Theorem (eigenvalue iteration) 38

3.3 Definition (convergent matrix) 38

3.4 Definition (spectral radius ρ) 38

3.5 Theorem (convergence determined by the spectral radius) 39

3.6 Definition (induced matrix norm) 39
3.7 Theorem (spectral radius and induced matrix norm) 39
3.8 Definition (Jacobian matrix evaluated at $\mathbf{x}$) 40
3.9 Definition (coefficient matrix J based on the Jacobian matrix) 40
3.10 Theorem (upper bond of spectral radius) 41
3.11 Theorem (convergence of noiseless soft regulation) 41
3.12 Theorem (robustness against bounded noises) 42
3.13 Assumption (representative agent) 42
3.14 Definition (largest singular value σ_1) 42
3.15 Definition (upper bound of open loop spectral radius m_g) 42
3.16 Theorem (contraction of the Euclidean norm) 43
3.17 Definition (stage cost v) 43
3.18 Definition (cumulative cost V) 44
3.19 Definition (optimal control) 44
3.20 Definition (steady-state mean squared error v_∞) 44
3.21 Conjecture (maximum contraction) 46
3.22 Definition (solution spectrum $\boldsymbol{\theta}^*$ and its width σ_θ) 49
3.23 Definition (root mean squared error (RMSE) v(t)) 50

4.1 Definition (optimization efficiency η) 54
4.2 Definition (linear demand $\tilde{d}_i$ and its revenue $\tilde{r}_i$) 86
4.3 Definition (regret v_i) 86

List of Figures

1.1 "Flowsheet" of a financial network: funding map of a bank/dealer system . 3
1.2 Signed digraph for a bank/dealer system 3
1.3 TeCSMART framework 5

2.1 Taxonomy of soft regulation 36

3.1 Optimal social influence from robust control by minimizing the RHS of (3.34) 48

4.1 Efficiency of soft regulation with best recommendation 57
4.2 Efficiency of soft regulation with crowd recommendation 57
4.3 Efficiency of soft regulation with crowd recommendation (large confidence levels) 62
4.4 Efficiency of soft regulation with crowd recommendation over time 62
4.5 Efficiency of soft regulation with crowd recommendation over time (distributed agents) 64
4.6 Instructions for the "Fitness Game" 68
4.7 Game interface 69
4.8 Learning process of each individual player and the wisdom of crowds 70
4.9 MSE time series of human subject experiment results 72
4.10 Monte Carlo simulation of the expected MSE time series 74
4.11 Optimal taxation: total general sales taxes (T09) 78
4.12 Optimal taxation: total license taxes (C118) 79
4.13 Optimal taxation: alcoholic beverage license taxes (T20) 80
4.14 Optimal taxation: individual income taxes (T40) 81

4.15 Optimal spending: expense item "(E037) Educ-NEC-Direct Expend" 82
4.16 Optimal spending: expense item "(E040) Emp Sec Adm-Direct Exp" 83
4.17 Optimal spending: expense item "(E065) Total Highways-Dir Exp" 84
4.18 Optimal spending: expense item "(E107) Liquor Stores-Tot Exp" 85
4.19 Multi-product total revenue time series 92
4.20 Multi-product MSE time series 93

A.1 NetLogo interface 105

B.1 IRB approval letter 114
B.2 IRB information sheet 115
B.3 Amazon Mechanical Turk 116

List of Tables

2.1 Control and learning in sociotechnical systems 25
2.2 Soft regulation applications 28
2.3 Examples: soft regulation as a toolset for multi-agent systems 35

3.1 Illustration of Conjecture 3.21 47

4.1 Agent-based simulation parameters 55
4.2 Optimal degree of social influence 75
4.3 The "Fitness Game" experiment results 75
4.4 U.S. optimal state tax and expenditure results 77
4.5 Multi-product revenue management simulation result 91

Notation

$\mathbf{1}$	column vector of 1s, i.e., $[1, \ldots, 1]^\top$
a	Kiefer-Wolfowitz algorithm parameter, linear demand parameter
A	compact notation of a complicated expression
b	β profile parameter
B	compact notation of a complicated expression
c	Kiefer-Wolfowitz algorithm parameter, linear demand parameter
d	opinion distance, demand
$\mathbb{E}$	expectation
f	payoff function, utility function
g	learning function, learning gain
G'	diagonal learning gain matrix
$\mathbf{g}'$	learning spectrum
h	soft regulation map
$\mathbf{h}$	vector-valued soft regulation map
i	agent index
j	agent index
J	soft regulation coefficient matrix, Jacobian matrix
k	quadratic parameter of payoff function
m	upper bound of spectral radius
m_g	upper bound of open loop spectral radius
MSE	mean squared error
n	total number of agents

p	price
$\mathbb{R}$	set of real numbers
t	time index
T	length of time horizon
u	soft feedback, recommendation
U	compact notation of a complicated expression
v	stage cost, mean squared error, root mean squared error
V	Lyapunov function, total cost, cumulative mean squared error
$\mathbf{v}$	eigenvector
x	state, decision error
X	compact notation of a complicated expression
$\mathbf{x}$	state vector for all n agents
y	payoff, utility, output
Y	cumulative payoff
z	action, decision
$\mathbf{z}$	action or input vector for all n agents
$\mathbb{Z}$	set of available actions or inputs
β	degree of social influence
δ	fraction value between 0 and 1, pricing perturbation
Δ	parameter of the revenue management model
$\Delta\,\mathrm{MSE}$	decrease of the cumulative expected MSE from the open loop
ϵ	uniformly distributed random variable
η	optimization efficiency
θ	solution, optimal setpoint
$\boldsymbol{\theta}^*$	solution spectrum
κ	scaling parameter for payoff function in the "Fitness Game"
λ	eigenvalue
ν	payoff disturbance, demand noise
ρ	spectral radius, largest absolute eigenvalue, pricing step size
σ_1	largest singular value

σ^2	variance
σ_g	width of the learning spectrum
σ_θ	width of the solution spectrum
σ_ω	noise
τ	time (when t is used in the same equation)
ω	random noise variable
$\boldsymbol{\omega}$	noise vector
$\sum_i$	equivalent to $\sum_{i=1}^n$ when there is no ambiguity
$\hat{x}$	variable derived from x
$x_i(t)$	value of variable x of the i-th agent at time t
x_∞	limit of variable $x(t)$ when $t \to \infty$
$\tilde{x}$	stochastic or time-varying variable x
x^*	optimal x that maximizes certain objective function, solution
x^+	short for $x(t+1)$
$x(0)$	initial value of variable x
$\bar{x}$	open loop decision based on current state x
$\mathbf{x}^\top$	transpose of $\mathbf{x}$
$\lVert\mathbf{x}\rVert$	norm of $\mathbf{x}$

Acknowledgments

Over the past five years, I have received support, encouragement, and inspiration from a great number of people. I would like to first express my deepest appreciation and gratitude to my advisor Prof. Venkat Venkatasubramanian. I was very fortunate to join Columbia University approximately the same time as him. Prof. Venkatasubramanian is as much warm and pleasant as one can hope for a mentor. Since then, it has been my greatest learning experience: I have had the opportunity to work on a remarkably diverse set of projects, including my advisor's three-decade quest for modeling collective dynamics through the lens of statistical mechanics. That project in many ways has influenced the work presented here. Prof. Venkatasubramanian's intuition and perspective are also unparalleled. His recommendation of the book "Nudge," for example, opened up a whole new world of the "wisdom of crowds" to me and eventually led to my central discussion in this study.

My co-advisor Prof. Garud Iyengar, on the other hand, has saved me from research bottlenecks countless times. I owe this thesis to his critical thinking, candid feedback, and breadth of knowledge. It is a delight to discuss with Prof. Iyengar because of his efficiency and enthusiasm: He always comes up with as many solutions as possible within the one-hour meeting window. While diligently leading a very busy department, Prof. Iyengar has spent much more time on helping me than I could presume. There were even times when we had to meet on Saturdays. I am truly grateful for his support.

I would also like to thank my committee members Prof. Jay Sethuraman, Prof. Sanat Kumar, and Prof. Daniel Esposito for their kind guidance and insights. Prof. Sethuraman and I used to have weekly meetings from 2014 to 2015 when we worked on the collective dynamics project with Prof. Venkatasubramanian. We continued to be in touch after completing the project. Over the years, he has given me valuable advice both on research and career in general. Prof. Kumar was one of the examiners of my Ph.D. qualifying exam, and he has served on my committee ever since. I have received much support and constructive advice from Prof. Kumar during his tenure as the department chair. My interactions with Prof. Esposito were not as frequent as other faculty members acknowledged here. Our brief conversations, however, were equally thought-provoking. I very much look forward to future discussions about energy research with him.

Needless to say, I feel blessed to have the amazing company in CRIS Lab namely Zhizun (Albert) Zhang, Miguel Remolona, Matthew Conway, and Akansha Singh, who have patiently sat through so many presentation rehearsals with me. While being a top-notch colleague, Albert is also a very dear friend and one of the most big-hearted persons I have ever known. Our daily walks not only gave us many new ideas in research, but they also made the windowless cubicle life more tolerable.

Last but not least, I would not be able to come this far without the love and support from my family and friends. My mother and father are both university professors. They introduced me to scientific research and inspired me to pursue an academic career. More importantly, they encourage me to chase my dreams and accept me for who I am. So do my soon-to-be spouse and my dearest friends. Thank you!

To mom and dad, who introduced me to research

Chapter 1

Multi-Agent Collective Dynamics and Control

1.1 Making Crowds "Smarter" (p. 2) This cross-disciplinary work addresses a simple yet important question: How could a group collectively become better problem-solvers? How could crowds become "smarter"? The goal is to show how the answer to this question can be obtained as derived results by modeling rigorously, through control theory, the collective dynamics of multi-agent systems.

1.2 Collective Intelligence (p. 9) We discuss here the collective dynamics of multi-agent systems. We first distinguish our work from the popular notion of the "wisdom of crowds," which has been studied for over a century, and introduce the concept of the "wisdom of learners." We then briefly review existing approaches to modeling collective dynamics.

1.3 Multi-Agent Control in Sociotechnical Systems (p. 18) This is an overview of multi-agent control, its related work, and our contribution of incorporating it to sociotechnical systems. This section is a prelude to our main methodology: soft regulation.

1.1 Making Crowds "Smarter"

1.1.1 Chemical Engineering Beyond Plants

Process control is indispensable to many complex engineering systems. This research discipline has evolved from the early feedback control to the modern optimal control. In chemical engineering, for example, every syllabus of process control begins with feedback control, i.e., adaptively changing the input of a process based on feedback from the output. Such setup aims to steer the system towards the setpoint. Feedback control is simple, elegant, and usually, implementing it does not require a model. On the other hand, given a reliable model of the process dynamics, one can then develop sophisticated control schemes such as optimal control to achieve higher precision. In such case, the controller solves an optimization problem by selecting the most viable path that minimizes the "cost" of operation. Regardless of its execution (i.e., feedback control, optimal control, or other types), every process control deals with open loop and closed loop dynamics. Open loop dynamics is the subject of process control; closed loop dynamics is the result.

The control problems or open loop dynamics we encounter in chemical engineering are often related to chemical processes, manufacturing, scheduling, etc. In other words, they are the *control of machines*. Meanwhile, complex *sociotechnical systems*[1], such as modern financial systems and the Internet, are characterized by similar interdependencies and a large number of units one often observes in a chemical plant. In fact, we showed in our previous work that one can model the "flowsheet" of a financial system (Fig. 1.1, p. 3) and capture the complex dynamics in a signed digraph (Fig. 1.2, p. 3) as often applied to fault diagnosis in a chemical plant [Bookstaber *et al.*, 2015]. It is tempting to wonder, what would be the process control in sociotechnical systems beyond chemical plants?

Control in sociotechnical systems has several implications for chemical engineering. One example we will introduce in Chapter 2 (p. 23) describes the problem of optimal regulation and policy. Quite often, the regulator or policymaker is as uncertain about an emerging

[1] *Sociotechnical systems*: an approach to complex organizational work design that recognizes the interaction between people and technology in workplaces. The term also refers to the interaction between society's complex infrastructures and human behavior

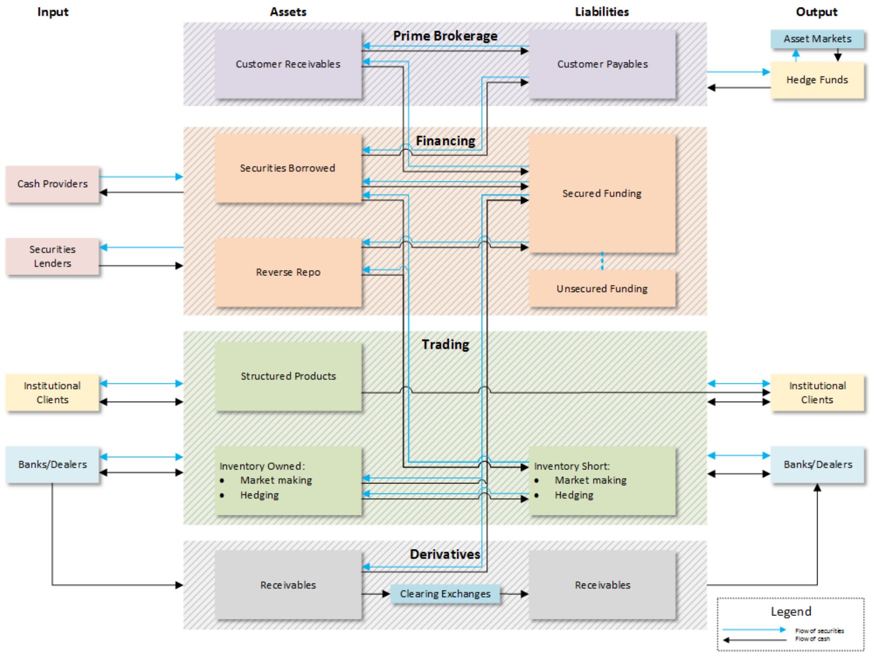


Figure 1.1: **"Flowsheet" of a financial network: funding map of a bank/dealer system**.

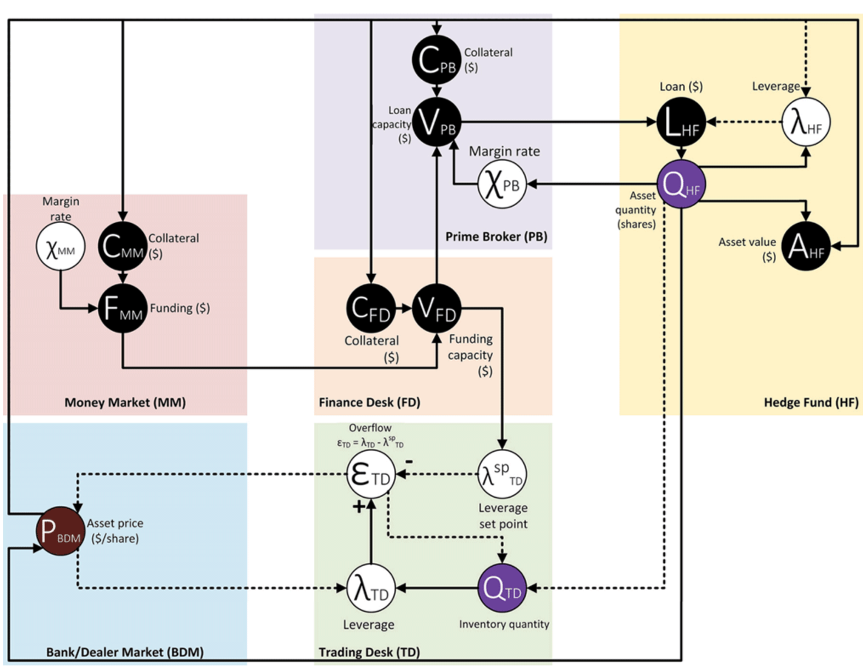


Figure 1.2: **Signed digraph for a bank/dealer system**.

technology or business as other individuals in the system. Controversial topics such as fracking and carbon pricing have generated heated debates in recent years both within and outside the chemical engineering community [Gold, 2014; Bloomberg and Krupp, 2014]. How can we strike the right balance between over- and under-regulation? Is there a way to accelerate the process of optimizing the regulation of emerging technologies and businesses? These are among the many questions we attempt to answer in this work. They represent the *control of intelligent agents*. Consequently, there are new challenges such as self-interest, rationality, and incentives beyond process control in engineering systems. We will address those challenges both through theory (Chapter 3, p. 37) as well as simulations and experiments (Chapter 4, p. 51).

Feedback control has long been recognized as an essential feature of complex adaptive systems where causes and effects are intertwined. There have been several attempts over the years to understand the dynamics of social systems in terms of feedback control (see, e.g., [Powers, 1973; Carver and Scheier, 1982; Trochim *et al.*, 2006; Leveson, 2011]). In addition, Venkatasubramanian and Zhang developed a seven-layer hierarchical framework TeCSMART (Fig. 1.3, p. 5) to describe complex sociotechnical systems from a unifying *process systems engineering (PSE)*[2] perspective [Venkatasubramanian and Zhang, 2016].

TeCSMART provides a structural framework by describing complex systems in terms of feedback loops and their ensembles. Techniques in PSE are readily transferrable to the modeling, design, and control of systems beyond chemical plants. The hierarchy consists of seven layers: process, plant, company, market, local government/regulatory agency, (federal) government, and society. The seven layers can be further divided into small subsystems, e.g., the subsystem of processes and plant, companies and market, as well as regulators and government. Our focus in this work is primarily on those subsystems that consist of multiple decision makers.

[2] *Process systems engineering (PSE)*: study of the design, control, scheduling, optimization, and risk management of large-scale chemical systems

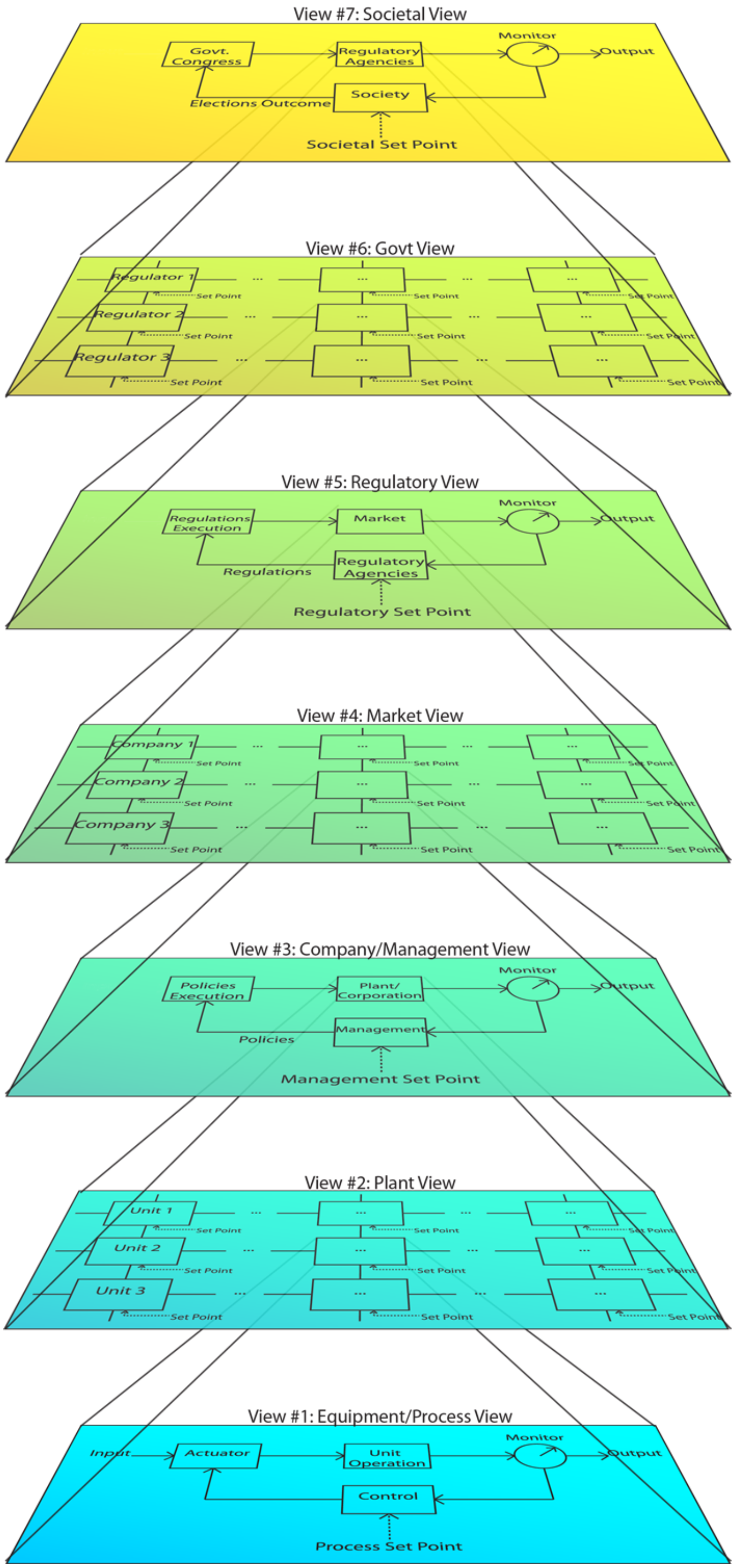


Figure 1.3: **TeCSMART framework**.

1.1.2 Can We Make Crowds "Smarter"?

Often, a large *crowd*[3] of decision makers are attempting to solve the same problem with similar information in a largely independent manner. For the common man, these problems could be as simple as choosing the most appropriate product or improving personal fitness. For a crowd of local governments or nations, the problem could be optimal taxation to promote economic growth. The process of identifying the appropriate decision involves an expensive trial and error process to explore the entire space. Minimizing this search cost by coordinating and improving this collective *learning*[4] process, by making crowds "smarter," has immense societal value.

Optimization typically involves balancing trade-offs. Consider the problem of optimal taxation. Under-taxation results in insufficient funds towards public services and government functioning, whereas over-taxation drives businesses to places where taxes are lower, leading once again to a deficit for the state. Local governments face similar dilemma when setting expenditure to balance between under- and over-spending. The main question we address in this work is whether one can accelerate convergence by making the crowd of fifty states "smarter." Even a small improvement in the convergence rate, magnified by the scale of the problem, could potentially save the nation billions of dollars while improving the overall welfare.

Definition 1.1 (problem statement)**.** A crowd consists of n intelligent individuals (or agents). Each agent attempts to solve a unique optimization problem of which the solution is identically θ^*. The time series $z_i(t)$ denotes the learning process of the i-th agent. The time series of the column vector $\mathbf{z}(t) = [z_1(t), \ldots, z_n(t)]^\top$ denotes the crowd's collective learning process, of which the performance is measured by the mean squared error $\mathrm{MSE}(t) = \frac{1}{n} \|\mathbf{z}(t) - \theta^* \mathbf{1}\|_2^2$. The objective — making the crowd "smarter" — is to accelerate convergence of the collective learning process, i.e., the decay of $\mathrm{MSE}(t)$ or the speed of convergence of $\mathbf{z}(t)$ towards $\theta^* \mathbf{1}$.

[3] *Crowd*: n intelligent individuals (people, organizations, governments, etc.) who attempt to solve the same optimization problem (e.g., finding the ideal diet that maximizes one's health and fitness)

[4] *Learning*: an optimization process that begins with an initial guess of the solution and gradually converges to the solution in probability

Definition 1.2 (action $z_i(t)$)**.** $z_i(t)$ denotes the i-th agent's action at time t. Actions can be decisions, strategies, policies, etc., that would generate certain payoff, reward, or (evolutionary) fitness, which the i-th agent tries to maximize.

Definition 1.3 (solution θ^*)**.** θ^* denotes the optimal action (or solution) that would generate the maximum expected payoff, reward, or (evolutionary) fitness.

Assumption 1.4 (convergence of learning)**.** Each learning process converges to the solution in probability (regardless of the initial condition):

$$\lim_{t \to \infty} \mathbb{E}\big[z_i(t) | z_i(0)\big] = \theta^*. \tag{1.1}$$

1.1.3 Existing Solutions

Using a coordinated crowd or swarm to solve complex problems is well studied in the literature. Particle swarm optimization (PSO) [Kennedy, 2010] is a widely adopted global optimization technique that uses a crowd of simple solvers to explore the fitness landscape of a problem. This swarm of PSO solvers mimics the swarming behavior observed in nature, e.g., among bees, ants, and birds. Each PSO solver revises its search direction based on its past performance and the position of the solver that observes the highest fitness. The PSO technique is very effective in solving deterministic problems that have multiple local extrema. However, PSO or any other parallel computing methodology cannot help us in improving the rate for learning in the optimal regulation and policy setting. The critical difference is that in the PSO setting each solver observes the same function; however, the reward or fitness of an individual in a crowd is typically subjective, private, very noisy, and often, not even numerically expressible. On the other hand, the *inputs*, i.e., actions, are often numerically well defined. We exploit this feature to develop a learning algorithm.

Wisdom of crowds describes the phenomenon — first introduced as *vox populi* in 1907 by Francis Galton [Galton, 1907b], then rediscovered and popularized by James Surowiecki a century later [Surowiecki, 2005] — that the average opinion of a crowd is remarkably close to the otherwise unknown truth although the opinions of individuals in the crowd are very erroneous. This phenomenon partially justifies the efficiency of polling and prediction markets, where a surveyor can gather an accurate estimate of an unknown variable by averaging over multiple independent and informed guesses. Explanations [De Condorcet, 2014; Bergman and Donner, 1964; Simons, 2004] for the success of the wisdom of crowds assume that individuals' estimates are unbiased and *independently* distributed [Surowiecki, 2005; Kittur and Kraut, 2008; Goldstone and Gureckis, 2009; Alvarez, 2011; Lorenz *et al.*, 2011; Quinn and Bederson, 2011]. Social influence renders the wisdom of crowds ineffective [Goldstone and Gureckis, 2009; Sumpter and Pratt, 2009; Lorenz *et al.*, 2011], and in order to guarantee accuracy, interactions among the respondents should be discouraged. Since individuals make decisions solely based on their prior knowledge and expertise, some even suggest *vox expertorum*, instead of *vox populi*, to be a more suitable name [Galton, 1907a; Conradt and Roper, 2005; Goldstone and Gureckis, 2009].

Definition 1.5 (degree of social influence β_i)**.** The degree of social influence (β_i) is the percent adjustment of the i-th agent's action $z_i(t)$ towards another agent's action $z_j(t)$. The adjusted action becomes

$$z_i(t+1) = (1-\beta_i)z_i(t) + \beta_i z_j(t). \tag{1.2}$$

In the absence of social influence (i.e., $\beta_i = 0$ or open loop), each agent sticks to its own action. If $\beta_i = 100\%$, the i-th agent simply copies the j-th agent's action and overrides its own.

Increasingly, today individuals are getting all their information from highly inter-connected online social networks; thus, truly independent opinions are becoming rare. The existing literature suggests that *vox populi* should not be effective. And yet, online networks with very high degree of social interaction appear to be able to harness information effectively to benefit the individuals. We are relying on polling evermore, for selecting movies, restaurants, books, shows, etc. The polls appear to be working in identifying good options, even though the votes are highly correlated. The crowd benefits from these interactions by converging to optimum faster. Social influence here improves, rather than undermines, the collective learning process. How does one reconcile with the previous results on the degradation of the impact of *vox populi* in the presence of social influence? *Is there an optimal degree of social influence for a learning crowd?* This is the question we address in this study.

1.2 Collective Intelligence

1.2.1 Wisdom of Experts v. Wisdom of Learners

Multi-agent collective dynamics is a prevalent phenomenon in nature. Examples include self-assembly, coupled oscillators, cell migration, quorum sensing, birds flocking, ant colony, etc., which all share something in common: Individuals interact with the crowds that they belong to and benefit from such interactions. The wisdom of crowds, a century-old discovery in statistics, offers an intuitive explanation: The average opinion of a crowd can be remarkably close to the otherwise unknown truth even though the opinions of individuals in the crowd are very erroneous. We briefly introduced the wisdom of crowds in Section 1.1.3 (p. 7).

This concept, also known as *vox populi*, was introduced in 1907 by the British polymath Francis Galton [Galton, 1907b]. He observed an ox weighing contest where 800 contestants tried to best guess the weight of an ox. After analyzing 787 valid ballots, Galton discovered that the average guess was merely one pound off of the ox's true weight. Thus, the average opinion was strikingly close to the otherwise unknown truth. A century later, James Surowiecki popularized the same concept in his book *The Wisdom of Crowds* [Surowiecki, 2005] and re-started a discussion among experts in decision theory, business, and ecology [Sumpter, 2006; Kittur and Kraut, 2008; Goldstone and Gureckis, 2009; Krause *et al.*, 2010; Huizingh, 2011; Lorenz *et al.*, 2011; Quinn and Bederson, 2011].

In Surowiecki's book, the examples are even more remarkable, among which, the Challenger explosion story is, to say the least, quite intriguing. Within minutes after the news broke out on January 28, 1986, the stock market started to respond to this tragic event: Investors began selling the stocks of the four major contractors who had participated in the Challenger mission. Their stock prices plummeted. Among the four, Morton Thiokol — the actual manufacturer of the O-ring seals — was hit the hardest: By the end of the day, its stock price was down nearly 12 percent while the other three rebounded to only 3 percent down. In hindsight, this distinction was not surprising. For it was revealed six months later that the O-ring seals indeed became less resilient and created gaps that allowed the gases to leak out [Challenger and Rogers, 1986]. In other words, Morton Thiokol, among the four contractors, bore the direct blame for the disaster. But how could the investors knew that collectively?

Here is Surowiecki's explanation:

> [It's] plausible that once you aggregated all the bits of information about the explosion that all the traders in the market had in their heads that day, it added up to something close to the truth ... [Even] if none of the traders was sure that Thiokol was responsible, collectively they were certain it was.

There are several problems with this explanation. As a journalist, James Surowiecki focused primarily on describing the stories and offering qualitative insights. While the insights are as persuasive as folk wisdom, they lack the mathematical vigor. Similar views are often quoted to support the notion that the stock market is forward looking. However, we all know that the stock market can and often do overreact to news instead of truly reflecting the values and sustainability of market as Paul Samuelson famously mocked such notion [Samuelson, 1966]:

> The stock market has forecast nine of the last five recessions.

The central limit theorem, Condorcet's jury theorem [De Condorcet, 2014], and many wrongs principle [Simons, 2004], all provide partial explanations for the success of the wisdom of crowds under the assumption that the individuals' estimates are unbiased and independently distributed. By averaging out the error (or white noise), one can estimate the mean with greater confidence. The contestants in Galton's game and the investors in the Challenger aftermath all possessed certain knowledge despite the great uncertainty: The contestants were butchers and farmers who weighed livestock for a living [Galton, 1907a]. The investors devoted constant attention to Thiokol's performance. In other words, they formed knowledgeable crowds with each individual decision drawn from a distribution. As long as their knowledge was not systematically biased, the average would be a distribution more narrowly peaked around the true value. In some sense, since individuals make decisions solely based on their prior knowledge and expertise, the term *vox expertorum* (or wisdom of experts) instead of *vox populi* might be more suitable to describe the two examples [Galton, 1907a].

In nature, however, such prior knowledge-based wisdom of crowds is less common. Flocking birds, for example, are not a *knowledgeable* crowd. The environment (weather, predators, food, etc.) changes constantly. The uncertainty makes any prior knowledge of migration route useless. In other words, there isn't a static true value like the ox's weight or the O-ring malfunction, but a dynamic solution that adapts to the changing environment. So instead, the birds gather local information, follow their closest neighbors, and revise directions constantly. They are the *learning* crowds. In his famous paper "Flocks, herds and schools: a distributed behavioral model," Craig Reynolds enumerated three principles for the flocking phenomenon [Reynolds, 1987]:

1. Collision avoidance: avoid collisions with nearby flockmates
2. Velocity matching: attempt to match velocity with nearby flockmates
3. Flock centering: attempt to stay close to nearby flockmates

Cell migration, an orchestrated movement of cells in particular directions to specific locations, is another example of nature's wisdom of crowds. Similar to birds flocking, cells also only have access to local information and rely on interactions to function properly as a group. *So why does the stock market overreact to news while birds and cells benefit from interactions? What is the fundamental difference between these two types of crowds?*

To make the comparison less confusing, let's call the phenomenon described by Galton and Surowiecki the "wisdom of experts," and call what we observe in nature the "wisdom of learners." Note that experts in this context refer to individuals who voice informed opinions based on prior knowledge, regardless of the quality of such knowledge (this is a generalization of the conventional definition of an expert, i.e., a person who has a comprehensive and authoritative knowledge of or skill in a particular area). Learners, on the other hand, refer to individuals who revise and attempt to improve their decisions or opinions based on information they receive via rational trial and error.

Even though both the wisdom of experts and the wisdom of learners fall under the umbrella of the wisdom of crowds, they have a few key differences. For instance, interaction is missing from the wisdom of experts, often done so purposely to avoid correlated opinions. Researchers tend to agree that social influence poses a big threat to the wisdom [Lorenz *et* *al.*, 2011], and in order to guarantee accuracy, interactions among the respondents should be discouraged. The reasoning is simple: Interactions violate the independency and thus undermine such wisdom. The consequence is a much undesired situation called *groupthink*[5] [Sunstein and Hastie, 2014]. So why is the wisdom of learners seemingly immune to this deterioration from interactions?

There are two possible explanations: First, interactions prove advantageous because self-interested learners are likely to filter feedback they deem unhelpful and only make use of the one that improves their performance. This self-interest based filtering appears to be key to the success of the wisdom of learners in a correlated environment. Another key difference between learners and experts is that learners are both respondents who generate new information, and surveyors who poll their social networks to improve their decisions. For each decision, the learner receives either a gain or a loss associated with it. On the other hand, for the wisdom of experts, information flows in only one direction, i.e., from the respondents to the surveyor. Decisions are only of importance to the surveyor.

Second, the influence of others' opinions is not as prominent as a decision factor for learners as it is for experts. Literature [Lorenz *et al.*, 2011; Krishnamurthy and Poor, 2014] suggests that, given the chance of revising one's decision after observing others', the individual tends to flock towards a consensus that might be off from the true value. For learning crowds, however, the learners receive information not only from the crowd's feedback but also by learning on their own.

1.2.2 Applications of Collective Intelligence

The wisdom of crowds sees its applications in many modern settings. Crowdsourcing, for example, loosely adapts such concept. It combines the words crowd and outsourcing, meaning to obtain (information or input into a particular task or project) by enlisting the services of a number of people. In today's world where individuals are members of highly connected social networks, one would expect the collective wisdom to be ineffective under the traditional wisdom of crowds setting (or as defined earlier, wisdom of experts). And

[5] *Groupthink*: a psychological phenomenon that occurs within a group of people in which the desire for harmony or conformity in the group results in an irrational or dysfunctional decision-making outcome

yet, we are relying on polling evermore, for selecting movies, restaurants, books, shows, etc. Such polls appear to be working in identifying good options even though votes are correlated.

We are witnessing the coexistence of three types of crowdsourcing. The first type, primarily supported by the wisdom of experts, is polling *before* the Internet (and social networks) era. The surveyor (pollster) outsources his/her problem of finding certain true value by collecting the average opinion among independent individuals. It is always a one-shot action. Neither interaction nor revision of opinion is allowed so that independency is preserved. Galton's *vox populi*, literally voice of the people, laid down the scientific foundations for democratic voting and other polling measures. As Galton himself commented after observing the ox weighing contest:

> The result seems more creditable to the trustworthiness of a democratic judgment than might have been expected.

Independency critically determines the polling accuracy. For a very long time, pollsters collect opinions by calling landline numbers randomly from the yellow page. This helps eliminate the demographic bias and preserve the independency. As landline is becoming obsolete, such polling method also faces dire challenges. The response rate has dropped from 40% in the 1990s [Craighill and Clement, 2014] to 9% in 2012 [Edwards-Levy and Jackson, 2016]. Even though communication has shifted from telephones to the Internet, such transition is not happening soon for polling because it is very difficult to find an unbiased demography. Each website has its own dominant demographics. There are, thankfully, ongoing efforts to correct such bias. For instance, in "Forecasting elections with non-representative polls," researchers managed to process a highly biased opinion poll (game console XBOX users, predominantly young males) and predict the 2012 election statistics [Wang *et al.*, 2015].

The second type of crowdsourcing lies between the wisdom of experts and the wisdom of learners. Whoever uses the Internet should be familiar with this type of crowdsourcing. They are the online review systems. Similar to the first type, an online rating also takes the average opinion and the users usually only participate once. Feedback is the key difference here: The average is publicly available to the new participants. Let's take the online restaurant review Yelp as an example. The truth in this case would be the quality of restaurants being reviewed by Yelp's users. Except for the very first few customers, people who review a restaurant are not making independent decisions because they are aware of others' opinions. In this case, the customers are both the respondents who provide information and the surveyors who poll opinions to better make decisions.

There are debates over whether the online review system is able to identify the best options. One criticism towards the reviews is that the final rating is sensitive to the initial conditions [Salganik *et al.*, 2006; Krishnamurthy and Poor, 2014]. That is to say, if an otherwise fine product receives a few negative reviews at the very beginning, it might prevent customers from giving it a second chance and discovering its true value. There are also a growing number of studies on the effect of social influence. The general consensus is that social influence undermines the wisdom of crowds [Goldstone and Gureckis, 2009; Sumpter and Pratt, 2009; Lorenz *et al.*, 2011].

We are now transitioning towards the third type of crowdsourcing that reflects the massive collective intelligence of learners. The wisdom of learners, as introduced in the last section, is not an entirely new concept because it is essentially nature's wisdom of crowds: Individuals interact with one another to survive and proliferate. Neither is this concept new for the human society: Team collaboration, research conferences, think tank, etc., are all "offline" examples of such collective intelligence. As the society shifts towards the highly connected social networks, where people can easily and quickly get updated information on a variety of topics in almost real-time, the wisdom of learners is experiencing a paradigm shift from centralization to distributed networks, from small or moderate scales to massive scales, and from delayed information exchange to real-time updates.

For the third type of crowdsourcing, there are two driving forces of the crowd dynamics. At the individual level, each tries to maximize his/her own utility by exploring and learning from the past. Let's call it the learning force. At the social network level, one's decision also influences others and consensus might eventually be reached. Let's call it the consensus force. The two forces are the key of success for the third type of crowdsourcing.

On one hand, the learning force moves the crowd towards the right direction. Such force is fueled by the individual self-interest. Every trial and error is associated with either a gain or a loss. It is of the individual's interest to move along the gradient and climbs the fitness landscape. The force is therefore the gradient signal one receives by interfacing with the problem itself. However, because of one's limited learning capacity, this gradient signal might be noisy or inaccurate. Just like the individual opinions from the wisdom of experts, the individual learning forces might not be enough to discover the truth.

On the other hand, the consensus force induces convergence. Consensus forming is a well-studied subject in the multi-agent research community. Recent efforts have been focused on how network structures affect the process [Tanner, 1984], distributed consensus protocols [Nedic and Ozdaglar, 2009], and continuum modeling [Perthame and Tadmor, 1991]. Despite receiving much more attention than the learning force does, consensus alone cannot drive the system to the truth: A purely consensus-based dynamics does converge, but not necessarily to the optimum. That is why the wisdom of experts does not work well under social influence. Each individual opinion is based on prior knowledge. The revision is determined by the social influence instead of learning. Consensus forms at the cost of inaccuracy.

1.2.3 Modeling Collective Dynamics

Multi-agent collective dynamics describes the emergent system-wide behavior from the interactions of multiple self-organizing agents. We discussed in Section 1.2.1 (p. 9) two different forms of collective dynamics in terms of the wisdom of learners in nature and the wisdom of experts in statistics. Such study of collective dynamics envelopes a wide spectrum of research fields. We offer here a glimpse of this topic and do not attempt to exhaust the list. There are three broad schools of research: collective dynamics of physical phenomena, nature inspired science and engineering, and multi-agent control.

The physical angle of collective dynamics has existed long before the terms "collective dynamics" or "multi-agent systems" were coined. In physics, the study of molecular phenomena has been an ongoing effort to bridge molecules to macroscopic behaviors such as thermodynamics, heat and mass transfer, and hydrodynamics. Statistical mechanics, for example, connects thermodynamics to the statistical nature of molecules. The discovery of entropy being a logarithmic function of microscopic rearrangement of states, i.e., $S = k \log W$ [Boltzmann, 1877], was one of the many efforts to convince the world that tiny particles really exist.

Such connection has fascinated not only physicists but also people from outside the physics discipline, or even outside the scientific research world. Think about it: *Dumb* molecules bump into one another in a seemingly random way. A thermodynamics system, however, exhibits orderly behaviors from such chaotic molecular phenomena. What if the molecules are replaced with goal-driven and *intelligent* individuals? What would be the statistical mechanics for a social system?

Economics is one of the first fields that well embrace such connection. Economists are constantly simulating macroeconomic phenomena with physics in mind. At its extreme form, such mimicking results in a brand new subject called *econophysics*[6]. One of the topics in econophysics that has attracted much attention recently is income distribution and income inequality, as summarized by our earlier work [Venkatasubramanian *et al.*, 2015]:

> The increasing inequality in income and wealth in recent years, and the associated excessive pay packages of CEOs in the U.S. and elsewhere, is of growing concern among policy makers as well as the common person. However, there seems to be no satisfactory answer, in conventional economic theories and models, to the fundamental questions of what kind of income distribution we ought to see, at least under ideal conditions, in a free market environment, and whether this distribution is fair.

Many attempted to answer such question [Champernowne, 1953; Champernowne and Cowell, 1998; Piketty *et al.*, 2014; Saez and Zucman, 2014]. We pursued a bottom-up approach in the paper "How much inequality in income is fair? A microeconomic game theoretic perspective" [Venkatasubramanian *et al.*, 2015]. Instead of forcing the narrative into any particular physics model, we started with a game-theoretic formulation where each individual has his/her own utility function based on his/her income as well as interactions with other individuals. Game theory is a native economics branch that is both mathematical

[6] *Econophysics*: an interdisciplinary research field, applying theories and methods originally developed by physicists in order to solve problems in economics, usually those including uncertainty or stochastic processes and nonlinear dynamics

and takes behaviors into consideration. Population games and evolutionary dynamics [Sandholm, 2010], a recent addition to game theory, is an ideal framework for the income distribution problem: It is suitable for a large number of agents and instead of the one-shot setting in classical game theory, this new framework integrates utility-maximizing agents into a dynamical system. The connection between population games and thermodynamics is difficult to ignore: Entropy is a measure of fairness in a social system with goal-driven and self-interested agents. We also proposed a new measure of income distribution as an alternative to the widely used yet controversial Gini index. By examining income data of twelve countries from 1920 to 2012 under the new measure, our analysis suggests that the Scandinavian countries have managed to get close to the ideal distribution for the bottom 99% of the population, while the U.S. and U.K. remain less fair at the other extreme.

Fluid dynamics, like thermodynamics, also inspires new endeavors in social science. The opinion (or consensus) forming is an important subject of collective dynamics. A general approach is the so-called environmental averaging [Motsch and Tadmor, 2014]. The idea is simple: Each individual updates his/her own opinion by selectively averaging others' opinions. As the number of individuals approaches infinity, this multi-agent system becomes a continuum thereby the name social hydrodynamics [Tadmor, 2015]. Many findings of opinion forming are about how network structure and the weights of averaging affects the consensus forming. For instance, the subdominant eigenvalue of the adjacency matrix of the social network determines how fast a consensus forms [Tanner, 1984].

Biology and ecology also inspire numerous pursuits in multi-agent collective dynamics. For instance, particle swarm optimization [Kennedy, 2010] (mentioned in Section 1.1.3, p. 7) borrows the concept of swarm intelligence (i.e., flocking) from nature, and applies it to solving complex optimization problems. A typical swarm optimization problem has multiple local extrema in the "fitness landscape," (a fitness landscape visualizes the problem by mapping the state-space to the corresponding fitness, i.e., utility and payoff). A large number of elementary solvers form a swarm. Each solver, while exploring the fitness landscape, also occasionally follows the "leader," (the solver that has the best solution so far). This wisdom of learners type of optimization appears to be capable of finding a global optimum even when the fitness landscape is rugged with many local extrema.

Lastly, multi-agent control is the control engineering community's approach to collective dynamics. Controlling a large number of agents faces many challenges. The optimization problem becomes exponentially harder, and eventually intractable, as the number of agents increases. A centralized control is simply too costly to implement. There has been growing efforts in distributed control that breaks down the global optimization into localized problems that individual controllers can solve by themselves. A practical challenge is to implement distributed mechanisms that can reach consensus steadily and timely [Olfati-Saber and Murray, 2002; Olfati-Saber *et al.*, 2007]. That consensus can then be incorporated into local optimizations.

What about the control of multiple self-interested agents in sociotechnical systems? We now have the tools to model the behavior of intelligent agents (econophysics, population games, opinion dynamics, etc.); we also have the tools to control multiple agents (control theory, mechanism design, etc.). It is only natural to step forward and begin the quest from understanding complex collective dynamics to coordinating multiple self-interested agents. In the next section, we will discuss how to approach this problem and our modeling philosophy.

1.3 Multi-Agent Control in Sociotechnical Systems

Multi-agent control in sociotechnical systems (or "MACISTS") is the study of dynamical systems with multiple intelligent individuals and how the collective behavior is "controlled" by feedback. MACISTS here refers to a broader definition of control than that of process control, where the *input* can be manipulated directly (e.g., opening of the fuel valve of an automobile). In sociotechnical systems, such manipulation has to be subtler and more indirect. Often, there might involve "nudges" that carefully steer crowds towards certain desired behaviors [Thaler and Sunstein, 2008]. We hope to approach MACISTS in a general and mathematical manner such that we can generate unambiguous and quantitative conclusions. Our motivation can be perfectly summarized by Geoffrey West from the Santa Fe Institute, when he commented on the mathematical regularities underlying both biological and social systems [West, 2012]:

> Are there any principles at work? Can we put this into a mathematical framework? Can we make it a quantitative and predictive science of these phenomena? Part of that is purely for the understanding and for the satisfaction of understanding. And the last part is, this is the great hope, in some ways can it actually be of practical importance and have a significant impact on the wellbeing of my fellow human beings.

The first step in MACISTS, like other concepts in control theory, is the study of open loop dynamics. We can always model the behavior of a specific agent i ($i = 1, \ldots, n$) as if it follows the evolution below:

$$z_i(t+1) = \bar{z}_i(t) \equiv g_i\big(z_i(t)\big) + \omega_i(t). \tag{1.3}$$

The mapping g_i (later we will formally define it as the "learning function") encodes the i-th agent's rational respond to its current action $z_i(t)$, while the zero-mean random variable $\omega_i(t)$ indicates a white noise that affects the action $\bar{z}_i(t)$ this agent takes. Should g_i satisfy certain regularity criteria (such as the existence of a unique and attracting fixed point θ^*, Assumption 1.4, p. 7), agent i would *in probability* converge to an equilibrium state and thus reach its goal (without external feedback).

The goal for any rational individual is always to maximize certain utility or payoff:

$$y_i(t) = f_i\big(z_i(t)\big) + \nu_i(t). \tag{1.4}$$

y_i is the *output* variable and subject to another white noise $\nu_i(t)$. An acceptable g_i, therefore, is supposed to have its fixed point θ^* that maximizes f_i.

Then we introduce interactions among agents via *selective averaging* mediated by the degree of social influence β_i, i.e.,

$$\begin{aligned} z_i(t+1) &= \sum_j s_{ij}\big[(1-\beta_i)\bar{z}_i(t) + \beta_i z_j(t)\big], \\ &= (1-\beta_i)\bar{z}_i(t) + \beta_i \sum_j s_{ij} z_j(t). \end{aligned} \tag{1.5}$$

We will formally introduce the social network parameter s_{ij} in the next chapter. For the i-th agent, s_{ij} always sums up to 1. In the simplest situation, $s_{ij} \equiv 1/n$. The above equation indicates that the i-th agent adjusts itself partially towards the wisdom of crowds by a factor of β_i. Based on the actual social network s_{ij}, this agent then finalizes its decision by aggregating the effects from all the signals it receives. Note that unless otherwise stated, we use $\sum_i$ instead of $\sum_{i=1}^n$ to indicate the summation over n agents for simplicity.

The state-space representation used here is the standard language of modern control theory. Open loop dynamics encodes how the state of a system evolves without the interference of control as depicted in (1.3), while closed loop dynamics considers feedback, e.g., (1.5). Nonetheless, open and closed loop systems are relative and scale-dependent. For instance, in the control of an intelligent agent, the open loop dynamics reflects the agent's own decision process that navigates from current state to the next in the absence of social influence. Such decision process, if examined under microscope, actually consists of closed loop neurological and physiological feedback processes.

The input is a variable that one can change *hoping* to steer the system towards a desired state. In industrial process control such as temperature control and velocity control, the input is usually designed by heuristics (see "PID control" [Seborg *et al.*, 2010]) and the open loop dynamics is often modeled as a black box. Tuning the controller parameters (proportional gain, integral gain, derivative gain, etc.) brings certain optimality to the control system. It requires trials and errors to obtain a suitable tuning. Biological control (e.g., blood sugar control) falls under this category. The controller gains are primarily determined by one's genes and it takes generations of evolution to "optimize." Here the degree of social influence β_i is the input in MACISTS. Note that we avoid using the alternative terminology "manipulated variable" for the input β_i because in a sociotechnical setting, we cannot *directly* manipulate the social influence. But we still hope that, by understanding how social influence affects the collective dynamics, we can implement MACISTS in significant and practical ways.

The output determines how good or bad the state is. In modern optimal control, the input is determined in such a way that the trajectory of the system dynamics is optimized: The total payoff is maximized (or the total cost is minimized) and the final state is inside an acceptable region (see "model predictive control" [Rawlings and Mayne, 2009]). To find an optimal control, one solves an optimization problem. On rare occasions, the system is simple enough that a closed-form analytical solution is possible by transforming the optimization problem into a Bellman equation or its variations (such as Hamilton-Jacobi-Bellman equation). Surprisingly, such technique was first discovered in the 18th century as the "principle of least action" and later widely applied to the fields of theory of relativity and quantum mechanics. The term "action" is in fact the cost of a particular trajectory of a system, and every physical law follows the trajectory that minimizes such action. It is beyond the scope of this work but always worth thinking: What is the fundamental connection between physical laws in nature and the very artificial optimal control? Are cities and other man-made systems following the same principles as biological and ecological systems?

We choose the state-space representation to model multi-agent control because its formulation matches the multi-agent collective dynamics. As mentioned earlier, the open loop dynamics is how agents decide their next-moves internally. The input reflects the interaction among agents and the social influence from the crowd. The output is the utility an individual receives by exploring the state-space. Also, by formulating the problem this way, we can apply existing theories in optimal control to collective dynamics.

Selective averaging via β_i and s_{ij} is what makes MACISTS unique by bringing self-interest and intelligence into the picture. Since agents can selectively choose between their own internal decision and the crowd's opinion, such control is *softer* than conventional process control and robotics, as the titular name soft regulation suggests. The social influence parameter β_i can be any real number, but we are particularly interested in the case where $\beta_i \in [0, 100\%]$ because such range also defines a spectrum of behaviors from being totally independent ($\beta_i = 0$) to blindly copying others ($\beta_i = 100\%$).

Selective averaging is a common practice. In particle swarm optimization (mentioned in Section 1.1.3, p. 7 and Section 1.2.3, p. 15), velocity of each particle is a convex combination between its own and the leader. In statistics, the James-Stein estimator [Efron and Morris, 1977] dominates the more intuitive least squares approach by shrinking the least squares estimator towards a global mean. What is surprising about this estimator is that even if the variables are unrelated to each other (e.g., baseball player's batting statistics and demographics of a country), such shrinking can still provide marginally better estimation considering the accuracy of *all* variables combined. This is also known as the Stein's paradox. In opinion dynamics, researchers also discover that on average people tend to compromise between their own opinions and others' by a factor of 30% (i.e., $\beta_i = 30\%$) [Lim and O'Connor, 1995; Harvey and Fischer, 1997; Yaniv and Kleinberger, 2000; Yaniv, 2004; Soll and Larrick, 2009].

In the context of the wisdom of learners, selective averaging provides a self-interest based filtering. Such filtering balances the two driving forces, i.e., the learning force and the consensus force (p. 14). If selective averaging is leaning towards learning (with a small β_i), the dynamics is close to open loop. On the other hand, if the selective averaging is leaning towards consensus (with a large β_i), there is more sharing and less exploration. Neither extreme might be ideal. How β_i affects the overall system-level dynamics and what the optimal β_i is are among the central questions of this work.

Chapter 2

Soft Regulation: A Multi-Agent Systems Toolset

2.1 Etymology: Hard Regulation v. Soft Regulation (p. 24) Soft regulation is our approach to multi-agent control in sociotechnical systems (or MACISTS). We introduce here the origin of the name "soft regulation" to contrast the often punitive mandates known as hard regulation. Soft regulation was first discussed in [Luo *et al.*, 2016] as an alternative to regulating emerging industries.

2.2 Taxonomy: A Multi-Agent Collective Dynamics and Control Framework (p. 31) In this section, we formally introduce soft regulation as a *toolset* for multi-agent systems research: It can model multi-agent collective dynamics for a variety of situations; it is a control-theoretic framework that *quantitatively* describes the collective intelligence; it is also an implementable algorithm that could make crowds "smarter." We generalize the soft regulation model from [Luo *et al.*, 2016] into a state-space control formula. We also elaborate on the possible topics that this model is suitable for.

2.1 Etymology: Hard Regulation v. Soft Regulation

2.1.1 Regulating Emerging Industries

Regulating emerging technologies is challenging, and often controversial; requiring a careful trade-off between stability, security, performance, and cost in an *uncertain* environment. Recent examples of emerging technologies include hydraulic fracturing, carbon sequestration, deep sea mining, geoengineering, and personalized medicine. Hydraulic fracturing, for example, has grown to be a transforming force in the petrochemical industries in recent years with its proponents and opponents debating passionately about its benefits and costs to the society with the attendant regulatory challenges [Bloomberg and Krupp, 2014; Gold, 2014].

The regulator's (or central planner's) dilemma with regard to emerging technologies is to strike the appropriate *balance* in regulation. Under-regulation can result in damage to plant personnel, surrounding communities, and the environment. Over-regulation, on the other hand, can hamper economic growth and security. When a technology is new, the inherent risks and benefits are not immediately obvious and only become clear over time, making it harder for the regulatory agency to strike the correct balance in the early stages. This uncertainty necessitates a framework that allows for a very close collaboration between the regulatory agency and the regulated entities that have direct access to field performance, and hence have direct knowledge of what worked and what did not.

In a typical regulatory environment involving conventional technologies, regulators issue mandates that have to be followed by the regulated agents. The agents face fines and other punitive consequences for non-compliance. We call this approach *hard regulation*. We argue that hard regulation is not very effective for regulating emerging technologies. Hard regulation also hinders innovation [Krupp, 2008]. The regulation of the Internet illustrates these issues very well. Laws like Digital Millennium Copyright Act (DMCA) and Stop Online Piracy Act (SOPA) have been criticized [Congress, 1998; Rai, 1999; Smith, 2011; Tribe, 2011; Phillips, 2012], as they arguably "reduce freedom of expression and undermine the dynamic, innovative global Internet." In addition, while attempting to protect intellectual property, these laws hurt computer security by inhibiting research on

Table 2.1: Control and learning in sociotechnical systems

	Control	**Learning**
Hard	feedback control, model predictive control, hard regulation, robot formation, laws, etc.	machine learning, stochastic approximation, Kalman filter, evolutionary dynamics, etc.
Soft	persuasion, soft paternalism, peer pressure, social engineering, mechanism design, etc.	social sensing, social learning, pervasive mobile computing, etc.

security related issues [Higgins, 2014]. During the period when a new technology is still maturing, the regulator is just as unsure as the regulated agents about the risk-benefit trade-off, and therefore, hard regulation, through its unintended consequences, could potentially do more harm than good. Instead of issuing potentially misdirected mandates, the regulator and the agents should jointly participate in learning about the emerging technology and its payoff structure. The focus of this work is on how to achieve this elusive goal through an intellectual framework that facilitates both *control* and *learning* in sociotechnical systems.

Control and learning are essential elements in managing risk and regulating behavior in sociotechnical systems (Table 2.1, p. 25). In a purely technical setting, i.e., when all the elements of the systems are machines, the common practice to maintain an efficient and stable system is to use *hard* control where the entities follow strictly specified policies. Process control, robotics, etc., are all examples of hard regulation or hard control. Especially in feedback control, the feedback is hard and has to be followed. When there is no reliable model or a desirable setpoint available, one needs to simultaneously learn and control the system dynamics. We call techniques, such as machine learning, stochastic approximation, etc., *hard* learning techniques since they also require the entities to follow strict instructions.

However, in a sociotechnical system with active human participants, hard control, or strict mandates, may not always be appropriate. Mandates can potentially do more harm than good as we argued earlier. A more appropriate course of action would be to offer options to agents that are adopted only when they are incentive compatible. We call such approaches *soft* control [Han *et al.*, 2006; Zhang and Parkes, 2008]. Examples of this approach include the soft paternalism approach for modifying social behavior [Thaler and Sunstein, 2003] wherein carefully designed options "nudge" people to make better decisions [Thaler and Sunstein, 2008]; or policy teaching [Zhang and Parkes, 2008] wherein the regulator allocates rewards in such a way that the induced action of agents maximizes the regulator's value. Other examples include efforts by utility companies to induce consumers to minimize power wastage by reporting average consumption [Brotman, 2014]; and health tracking devices, e.g., Fitbit and Apple Watch, that all incorporate social nudging to motivate physical activities. The soft control policy using peer pressure is shown to promote cooperation in these and other settings, both in theory and in practice [Kandel and Lazear, 1992; Aharony *et al.*, 2011; Mani *et al.*, 2013; Shmueli *et al.*, 2014].

As in the case with hard control, soft control can be used only when there is a reliable model and a well-defined setpoint. Soft paternalism and similar social mechanisms are effective because we understand saving energy and staying physically active are the right things to do. What if we do not know what is best for the agents? Soft learning is a class of learning mechanisms that appropriately incentivize agents in a social network to aggregate important information. Examples of soft learning include social sensing and social learning [Rendell *et al.*, 2010; Krishnamurthy and Poor, 2014; Shmueli *et al.*, 2014] in the context of real-time traffic information and online reviews (such as Yelp).

We propose *soft regulation* as a new regulatory paradigm that combines features of soft control and soft learning. The regulator aggregates key system-level statistics in a privacy-preserving [Abbe *et al.*, 2012] manner (individuals do not need to explicitly disclose their actions) and shares these statistics with all agents. The agents have the flexibility to accept, reject, or partially accept the recommendations from the regulator based on their own self interests. The recommendations are simply "nudges" [Thaler and Sunstein, 2008]. The mechanism does not interrupt the regulated entities who have direct access to field performance. It creates a collective learning environment for both the regulator and the agents. Soft regulation seeks a balance between over- and under-regulation: Agents have the freedom to rely on both individual exploration and social learning.

We expect soft regulation to be effective when the system has the following features:

1. *Imperfect information*: The action-utility payoff structure is poorly understood, i.e., the data are *noisy* and the models are absent or *incomplete*. Each individual may only possess partial information about the unknown process. Agents rely on inaccurate measurements, approximations, or subjective evaluations to optimize.

2. *Weak interaction*: The agents can optimize their own actions without taking into consideration the response of other agents, i.e., each's utility or payoff is only a function of the agent's own state, and the optimal setpoint is identical among agents. A good example of such a setting is the initial stages of a new technology; the resources being exploited are abundant and the profits of the agents are not limited by competition but by their ability to exploit the resource effectively. Although the reward an agent receives while operating at a setpoint may vary, the setpoint itself, however, is likely to be identical or at least restricted to a narrow range. The discovered setpoints (by soft regulation or traditional methods) will later become the industry standards when the technology matures. Another example of setting with weak or no interaction is when humans improve their own health conditions by changing habits, medications, or even environments. The interaction among agents is usually minimal. Although each has his/her own unique physiological configurations, grouped by characteristics such as age, gender, profession, etc., they are likely to exhibit common optimal setpoints within groups.

3. *Bounded rationality*: Agents are autonomous and self-interested, and they *always* move in a direction that locally improves utility, subject to available information.

Despite the name, soft regulation has applications beyond industrial regulation (Section 4.1, p. 53). The soft regulator module can be integrated in different control systems and problem-solving scenarios (Table 2.2, p. 28). We only analyze a specific and stylized

Table 2.2: Soft regulation applications

Action	$\rightarrow$	**Utility**
health behavior: e.g., sleep habit, exercise frequency, diet, etc.	$\rightarrow$	**health condition** e.g., sleep quality, BMI, etc.
operating condition: e.g., T, P, feed ratio, flow rate, catalyst, etc.	$\rightarrow$	**yield** e.g., production rate, etc.
workplace environment: e.g., indoor temperature, lighting, etc.	$\rightarrow$	**productivity** e.g., profitability, etc.
infrastructure planning: e.g., traffic light control, hospital resource, budget allocation, etc.	$\rightarrow$	**efficiency** e.g., congestion time, etc.

model here to illustrate the efficacy of the mechanism. In practice, soft regulation should be implemented and modified in a case by case manner.

The medical domain is another applicable area of soft regulation (Section 4.2, p. 65). Powered by mobile phones and wearables, researchers can now collect timely mass medical data (via Apple's ResearchKit [Apple, 2015] for example). Soft regulation is suitable in this scenario because medical research satisfies all three features, i.e., imperfect information (unknown relationships between patient behaviors and health conditions), weak interaction (one patient's condition is not affected by another's), and bounded rationality (patients always wish to improve their own health, however, have limited information). In addition, thanks to the convenience of mobile devices, we expect good participation rate. A large population size further ensures the accuracy of recommendation. Patients can optimize their own health while contributing to medical research. Even if patients do not want to optimize themselves, medical researchers may implement the soft regulation module to do that based on data collected locally.

2.1.2 A Stylized Model

We analyze a model of soft regulation that preserves the essential features discussed in the previous section. This multi-agent system consists of one regulator and n agents. Agent i wants to select an action z_i that maximizes the value of the real-valued and strongly concave utility function $\tilde{f}_i(z_i)$ over a convex compact set $\mathbb{Z} \subseteq \mathbb{R}$. We assume that although the individual utility functions $\tilde{f}_i$ might be different for each agent, the solution $\theta^* = \arg\max_{z \in \mathbb{Z}} \mathbb{E}\big[\tilde{f}_i(z)\big]$ is identical.

We also assume that the utility function $\tilde{f}_i(z_i)$ is not explicitly known, nor is it deterministic; agents cannot solve the optimization problem explicitly. In theory, by averaging out the noise, one can obtain a more accurate mapping of the utility function. However, in our setting of regulating emerging industries, each sample corresponds to actual utility each agent receives; therefore, they might not have the incentive to oversample at the location where the utility is low. The agents update individual actions using the following learning dynamics:

$$\bar{z}_i(t) \equiv g_i\big(z_i(t)\big) + \omega_i(t), \tag{2.1}$$

where g_i denotes the optimization algorithm (or the "learning function" to be introduced later in Definition 2.4, p. 31) used by the i-th agent and $\omega_i(t)$ denotes the noise associated with such optimization process. In practice, g_i can be any function that maps an old action z_i to a new action $\bar{z}_i$. In order to converge to the optimal θ^*, the function must satisfy regularity conditions. More specifically, g_i should converge to a unique fixed point regardless of the initial value of z_i (Assumption 1.4, p. 7). For instance, the Kiefer-Wolfowitz stochastic gradient method is a commonly used algorithm [Kiefer *et al.*, 1952] where

$$\tilde{g}_i\big(z_i(t), t\big) \equiv g_i\big(z_i(t), t\big) + \omega_i(t) = z_i(t) + \frac{a(t)}{c(t)} \cdot \Big(\tilde{f}_i\big(z_i(t) + c(t)\big) - \tilde{f}_i\big(z_i(t) - c(t)\big)\Big). \tag{2.2}$$

At time t, the i-th agent samples the payoff twice at the vicinity of its current action $z_i(t)$, which is only known to the agent. The parameters $a(t)$ and $c(t)$ are known and predefined. The agent then computes the next step according to (2.2). This algorithm is guaranteed to converge in probability when

$$\mathbb{E}\Big[\tilde{f}_i - \mathbb{E}[\tilde{f}_i]\Big]^2 < \infty, \quad \lim_{t\to\infty} c(t) = 0, \qquad \sum_{t=1}^{\infty} a(t) = \infty, \qquad \sum_{t=1}^{\infty} \frac{a(t)^2}{c(t)^2} < \infty. \tag{2.3}$$

We call a setting where an agent updates its action based on its own measurement the open loop scenario (or asocial learning as in [Rendell *et al.*, 2010]).

In the soft regulation setting the regulator computes a feedback recommendation $u(t)$. The agents then combine $u(t)$ with $\bar{z}_i(t)$ to compute a new action $z_i(t+1)$ in the following manner:

$$z_i(t+1) = h_i\big(z_i(t)\big) \equiv (1-\beta_i)\tilde{g}_i\big(z_i(t)\big) + \beta_i u(t) = (1-\beta_i)\bar{z}_i(t) + \beta_i u(t), \tag{2.4}$$

where $\beta_i \in [0,1]$ or $[0, 100\%]$ is a measure of the influence that the recommendation has on the i-th agent (Definition 1.5, p. 8). Action changes are relatively independent of recommendation for agents with small β (the *explorers*), and action remains in the vicinity of u for agents with large β (the *followers*).

Note that soft regulation is not an example of direct social learning as described in [Rendell *et al.*, 2010]: There is no "best agent" or "leader" to follow because the payoffs are private information and noisy. That said, explorers do resemble the asocial innovators and the followers resemble the copying agents in the social learning setting [Rendell *et al.*, 2010]. The value of β_i may be indirectly related to peer pressure [Kandel and Lazear, 1992; Aharony *et al.*, 2011; Mani *et al.*, 2013; Shmueli *et al.*, 2014]: The followers experience a higher peer pressure than the explorers, and therefore, set a higher value of β_i. Also note that $h_i\big(z_i\big)$ can be re-written as follows:

$$h_i(z_i) = z_i + (1-\beta_i)(\bar{z}_i - z_i) + \beta_i(u - z_i). \tag{2.5}$$

The soft regulation feedback function resembles the feedback seen in bird flocks and swarm intelligence [Kennedy, 2010].

2.2 Taxonomy: A Multi-Agent Collective Dynamics and Control Framework

2.2.1 A General State-Space Model

The previous section offers a specific example of soft regulation that first appeared in [Luo *et al.*, 2016] where the term was created. We generalize the model and discuss the essential components of soft regulation in this section.

Definition 2.1 (generalized soft regulation)**.** Here we generalize soft regulation as a modeling framework for multi-agent control in sociotechnical systems (MACISTS, Section 1.3, p. 18)

$$z_i(t+1) = \big(1-\beta_i(t)\big)\Big(g_i\big(z_i(t)\big)+\omega_i(t)\Big)+\beta_i u_i(t), \tag{2.6}$$

$$x_i(t) = z_i(t)-\theta_i^*, \tag{2.7}$$

$$y_i(t) = f_i\big(x_i(t)\big)+\nu_i(t). \tag{2.8}$$

Definition 2.2 (state $x_i(t)$ of the i-th agent)**.** The state variable $x_i(t) = z_i(t) - \theta_i^*$ is the *decision error*, i.e., difference between the individual decision/action $z_i(t)$ and the solution θ_i^* (Definition 1.3, p. 7). As a general case here, the solutions can be different. $x_i^* = 0$ thus indicates the state where the i-th agent reaches the solution θ_i^*.

Definition 2.3 (payoff function $f_i(\cdot)$ and noise $\nu_i(t)$)**.** $f_i(\cdot)$ is the payoff (or utility/fitness) function of the i-th agent. The solution $x_i^* = 0$ maximizes $f_i(x)$. The realized payoff $y_i(t)$ is disturbed by a zero-mean random variable $\nu_i(t)$. Note that we used $f_i\big(z_i(t)\big)$ in (1.4) (p. 19). One can easily replace $z_i(t)$ with $x_i(t)$ using (2.7) and construct a payoff function that takes $x_i(t)$ as argument. With an abuse of notation, we retain the symbol f_i to describe payoff as a function of the state. Similar reasoning is used in Definition 2.4 (p. 31) next.

Definition 2.4 (learning function $g_i(\cdot)$ and noise $\omega_i(t)$)**.** The learning function g_i of the i-th player encodes the process where the agent makes a decision, observes the corresponding payoff, and then updates the state.

With an abuse of notation, we also define the learning function in terms of the state variable $x_i(t)$ as $g_i(\cdot)$. The open loop time series $x_i(t)$ is given as

$$x_i(t+1) = g_i\big(x_i(t)\big) + \omega_i(t). \tag{2.9}$$

$\omega_i(t)$ is a zero-mean random variable with variance σ_ω^2 sampled at t. It represents the impact of the error in function evaluation on the decision. Such error can be a result of noise in measurement or external disturbance. If the learning function g_i is based on evaluation of the payoff f_i (e.g., the Kiefer-Wolfowitz algorithm introduced in the last section), $\omega_i(t)$ will also be correlated with $\nu_i(t)$.

Assumption 2.5 (regularity of the learning function)**.** For all $i \in \{1, \ldots, n\}$, the function g_i is differentiable, $x^* = 0$ is the unique attracting fixed point of g_i, and furthermore, g_i is a contraction [Browder, 1965], i.e., $|g_i'(x)| < 1$ for all $1 \leq i \leq n$ and $x \in \mathbb{Z}$ (domain $\mathbb{Z}$ was defined on p. 29). The closer $|g_i'(x)|$ is to 1, the slower $g_i(x)$ converges. This assumption is motivated by the fact that all agents converge to the solution in the open loop setting independent of the starting guess.

Definition 2.6 (learning gain $\tilde{g}_i'$)**.** From mean value theorem, we can also establish that $g_i(x)/x = g_i'(\delta x)$, where $0 \leq \delta \leq 1$ and $x \neq 0$, is strictly less than 1. We define learning gain, denoted by $\tilde{g}_i' \equiv g_i(x)/x$, as the amplification of decision error. We also define $\tilde{g}_i' = 0$ when $x = 0$.

Definition 2.7 (learning spectrum $\mathbf{g}'$ and its width σ_g)**.** We define $\mathbf{g}' = [\tilde{g}_1', \ldots, \tilde{g}_n']^\top$ as the learning spectrum of the system. We let its standard deviation $\sigma_g = \sqrt{\frac{1}{n}\sum_i (\tilde{g}_i' - \tilde{g}')^2}$ denote the width of the learning spectrum, where $\tilde{g}' \equiv \frac{1}{n}\sum_j \tilde{g}_j'$ is the average (or representative) learning gain.

The learning function is one of the two most important features of soft regulation (social influence, in Definition 1.5, p. 8, of course, is the other one). By defining the learning function in this way, we bypass the complexity and specificity of how individual agents optimize their payoffs. But rather, we only observe the stochastic convergence of the states x_i (or actions z_i).

Definition 2.8 (social network S)**.** S is an n-by-n matrix with each element represents the weight of connection. For a completely connected social network with equal weights, $S = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$.

Assumption 2.9 (stochasticity of social network)**.** S satisfies $\sum_j s_{ij} = 1$, i.e., row sum is 1.

Definition 2.10 (soft feedback $u_i(t)$)**.** We denote the soft feedback as the weighted population average:

$$u_i(t) \equiv \sum_j s_{ij} z_j(t), \tag{2.10}$$

where s_{ij} is the i-th row and j-th column of the social network S. Again, unlike feedback in control theory, soft feedback does not have to be followed.

Definition 2.11 (condensed soft regulation given identical solution)**.** If $\theta_i^* \equiv \theta^*$, by subtracting θ^* from both LHS and RHS of(2.6), we have the x-version of (2.6)

$$x_i(t+1) = \big(1 - \beta_i(t)\big)\Big(g_i\big(x_i(t)\big) + \omega_i(t)\Big) + \beta_i u_i(t), \tag{2.11}$$

where $u_i(t)$ is, with an abuse of notation, $\sum_j s_{ij} x_i(t)$. (2.11) will be the governing equation used for the majority of analysis in this study.

There are a few special cases of the soft feedback. For instance, if $S = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, the soft feedback is the wisdom of crowds feedback discussed in the last section, where $u_i(t)$ is simply the arithmetic average of $x_i(t)$.

Lastly, social influence was defined earlier (Definition 1.5, p. 8): β_i denotes the weight the i-th player places on the soft feedback while learning. $\beta_i = 0$ reduces the soft regulation setting in (2.11) to the open loop setting in (2.9).

2.2.2 Soft Regulation as a Toolset for Multi-Agent Systems

As the title of this chapter suggests, soft regulation provides a toolset for multi-agent systems. In Table 2.3 (p. 35), we present a partial coverage of scenarios where soft regulation could model. The list includes but is not limited to particle swarm optimization [Kennedy, 2011], James-Stein estimator [Efron and Morris, 1977], social learning [Rendell *et al.*, 2010], the wisdom of crowds [Galton, 1907b; Surowiecki, 2005], and collective dynamics [Perthame and Tadmor, 1991; Motsch and Tadmor, 2014; Tadmor, 2015]. A taxonomy is also included in Fig. 2.1 (p. 36).

Let's now analyze the different elements of soft regulation. First, the learning function can be identical across all agents or different. For the simplest (i.e., linear) learning function $g_i(x) = g_i x$, if it is identical, $g_i = g$. In the next chapter, we will focus primarily on the mathematical property of soft regulation under this situation (also known as the representative agent assumption in economics). Particularly, for identical linear learning function, increasing the degree of social influence slows down the speed of convergence in a noiseless setting.

Next, degree of social influence is the input for MACISTS (p. 18). It critically determines the performance of collective learning. There always exists an optimal degree of social influence such that the performance of collective learning is maximized. In the next chapter, we will have an in-depth discussion about this topic.

There are many ways to generate the soft feedback. In this study, we primarily focus on the grand average scenario where the feedback is simply the population arithmetic average from all individuals (i.e., *vox populi*). In the context of collective dynamics, how such feedback is generated can significantly affect how fast consensus is formed.

We assumed identical solution in our problem statement (Definition 1.1, p. 6). We will also briefly analyze the situation where the solutions are unique and different, and argue that if the solutions are sufficiently close, soft regulation would still improve the collective learning process. This has applications such as multi-product revenue management. It is related to the Stein's Paradox [Efron and Morris, 1977].

Lastly, the payoff function can be convex/concave or non-convex. A non-convex payoff function would require a global optimization technique. As we discussed in the last section, the learning function (Definition 2.4, p. 31) contains all information about optimizing payoff. Therefore, our analysis of soft regulation does not explicitly depend on the type of payoff function.

Table 2.3: Examples: soft regulation as a toolset for multi-agent systems

Scenario	Learning function	Social influence	Soft feedback	Solution	Payoff
Particle swarm optimization	Local best	$[0, 100\%]$	Leader	Identical	Non-convex
James-Stein estimator	LMS estimator	$[0, 100\%]$	Grand average	Different	Squared error
Social learning	Asocial learning	$\{0, 100\%\}$	Leader	Identical	Non-convex
Wisdom of crowds	-	-	Grand average	Identical	-
Collective dynamics	-	100%	Weighted average	-	-

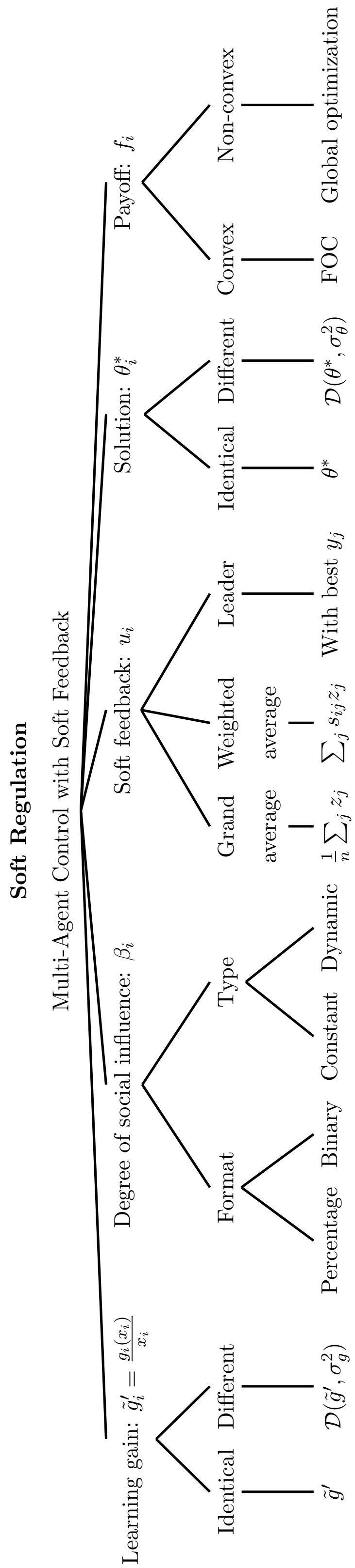


Figure 2.1: **Taxonomy of soft regulation.**

Chapter 3

Mathematical Properties of Soft Regulation

3.1 Convergence (p. 38) We investigate here the convergence properties of soft regulation. More specifically, we show that, in a noiseless setting, partially following the average converges to the solution if $0 \le \beta_i < 100\%$.

3.2 Robustness (p. 42) We show here that soft regulation is also robust against bounded noises. That is to say, the equilibrium (where every agent reaches the solution) can be restored when the system is disturbed by bounded noises.

3.3 Efficiency (p. 42) How much better can soft regulation improve the performance of collective problem-solving? We simplify the system such that $\beta_i = \beta$ is identical across agents. Doing so enables us to look closely into how a representative degree of social influence affects the efficiency of soft regulation.

3.4 Optimal Degree of Social Influence (p. 45) What is the theoretically maximum efficiency at which a crowd can solve a problem via soft regulation? We use control theory and linear algebra to answer this question.

3.5 Different Solutions (p. 49) We show here that even if the solution to each agent's problem is different, soft regulation can still improve the collective problem solving process.

3.1 Convergence

3.1.1 Mathematical Preliminaries

We first present here *nearly verbatim* a few important theorems from Peter J. Olver's *Numerical Analysis Lecture Notes* [Olver, 2008]. The notes are well written and useful for deriving our convergence result in Section 3.1.2 (p. 40).

Definition 3.1 (linear iterative system)**.** A linear iterative system takes the form

$$\mathbf{x}(t+1) = J\mathbf{x}(t). \tag{3.1}$$

The coefficient matrix J has size $n \times n$. For $t = 1, 2, 3, \ldots$, the solution $\mathbf{x}(t)$ is uniquely determined by the initial conditions $\mathbf{x}(0)$.

Theorem 3.2 (eigenvalue iteration)**.** *If the coefficient matrix J is complete, then the general solution to the linear iterative system* (3.1) *is given by*

$$\mathbf{x}(t) = \sum_i c_i \lambda_i^t \mathbf{v}_i, \tag{3.2}$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are the linearly independent eigenvectors and $\lambda_1, \ldots, \lambda_n$ are the corresponding eigenvalues of J. The coefficients $c_1, \ldots, c_n$ are arbitrary scalars and are uniquely prescribed by the initial conditions $\mathbf{x}(0)$.

Proof. Since we already know that (3.2) is a solution to the system for arbitrary $c_1, \ldots, c_n$, it suffices to show that we can match any prescribed initial conditions. To this end, we need to solve the linear system

$$\mathbf{x}(0) = \sum_i c_i \mathbf{v}_i. \tag{3.3}$$

Completeness of J implies that its eigenvectors form a basis of $\mathbb{R}^n$, and hence (3.3) always admits a solution. □

Definition 3.3 (convergent matrix)**.** A matrix J is called convergent if its powers converge to the zero matrix, $J^t \to O$, meaning that the individual entries of J^t all go to 0 as $t \to \infty$.

Definition 3.4 (spectral radius ρ)**.** The spectral radius $\rho(J)$ of a matrix J is its largest absolute eigenvalue, i.e., $\rho(J) = \max_{1 \le i \le n} |\lambda_i|$.

Theorem 3.5 (convergence determined by the spectral radius)**.** *The matrix J is convergent if and only if its spectral radius is strictly less than one: $\rho(J) \equiv \max_{1\leq i\leq n} |\lambda_i| < 1$.*

Proof. If J is complete, then we can apply the triangle inequality to (3.2) to estimate

$$\begin{aligned} \|\mathbf{x}(t)\| &= \left\| \sum_i c_i \lambda_i^t \mathbf{v}_i \right\|, \\ &\leq \sum_i |\lambda_i|^t \|c_i \mathbf{v}_i\|, \\ &\leq \rho(J)^t \sum_i |c_i| \|\mathbf{v}_i\|, \end{aligned} \tag{3.4}$$

where $\|\cdot\|$ denotes any (vector) norm on $\mathbb{R}^n$ (e.g., 1-norm, Euclidean norm, infinity norm). If $\rho(J) < 1$, then

$$\|\mathbf{x}(t)\| \to 0 \quad \text{as} \quad t \to \infty. \tag{3.5}$$

□

Definition 3.6 (induced matrix norm)**.** If $\|\cdot\|$ is any norm on $\mathbb{R}^n$, then the quantity

$$\|J\| \equiv \max_{\|\mathbf{v}\|=1} \|J\mathbf{v}\|, \tag{3.6}$$

defines the induced matrix norm of J.

Theorem 3.7 (spectral radius and induced matrix norm)**.** *The spectral radius of a matrix is bounded by its induced matrix norm:*

$$\rho(J) \leq \|J\|. \tag{3.7}$$

Proof. If λ is a real eigenvalue, and $\mathbf{v}$ a corresponding unit eigenvector, so that $J\mathbf{v} = \lambda\mathbf{v}$ with $\|\mathbf{v}\| = 1$, then

$$\|J\mathbf{v}\| = \|\lambda\mathbf{v}\| = |\lambda| \|\mathbf{v}\| = |\lambda|. \tag{3.8}$$

Since $\|J\|$ is the maximum of $\|J\mathbf{v}\|$ over all possible unit vectors, according to Definition 3.6, this implies that

$$|\lambda| \leq \|J\|. \tag{3.9}$$

If all the eigenvalues of J are real, then the spectral radius is the maximum of their absolute values, and so it too is bounded by $\|J\|$, proving (3.7). We omit the proof here for complex eigenvalues. See [Olver, 2008] for the detailed proof. □

3.1.2 Convergence of Noiseless Soft Regulation

Recall the noiseless closed loop soft regulation dynamics:

$$x_i(t+1) = (1-\beta_i)g_i\big(x_i(t)\big) + \beta_i \frac{1}{n}\sum_j x_j(t). \tag{3.10}$$

The individual learning functions $\{g_i(\cdot) : 1 \leq i \leq n\}$ are assumed to satisfy the regularity condition in Assumption 2.5 (p. 32). Let $\mathbf{x} \equiv [x_1, \ldots, x_n]^\top$ denote the state vector for the n agents. The noiseless soft regulation map for the vector $\mathbf{x}$ is given by $\mathbf{x}(t+1) = \mathbf{h}\big(\mathbf{x}(t)\big)$ where the map

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} (1-\beta_1)g_1(x_1) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & (1-\beta_n)g_n(x_n) \end{bmatrix} + \frac{1}{n}\begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} \mathbf{1}^\top \mathbf{x}. \tag{3.11}$$

Definition 3.8 (Jacobian matrix evaluated at $\mathbf{x}$)**.** The Jacobian matrix of the nonlinear iterative system (3.11) is the matrix of all first-order partial derivatives of $\mathbf{h}(\mathbf{x})$ evaluated at $\mathbf{x}$:

$$\begin{aligned} J(\mathbf{x}) &\equiv \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \ldots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \ldots & \frac{\partial h_n}{\partial x_n} \end{bmatrix}, \\ &= \begin{bmatrix} (1-\beta_1)g_1'(x_1) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & (1-\beta_n)g_n'(x_n) \end{bmatrix} + \frac{1}{n}\begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} \mathbf{1}^\top. \end{aligned} \tag{3.12}$$

Since $\mathbf{h}(\mathbf{0}) = \mathbf{0}$, the mean value theorem implies that

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} J_1(\delta_1 \mathbf{x}) \\ \vdots \\ J_n(\delta_n \mathbf{x}) \end{bmatrix} \mathbf{x}, \tag{3.13}$$

for some $\delta_i \in [0,1]$, $i = 1, \ldots, n$, and $J_i(\delta_i \mathbf{x})$ denotes the i-th row of the Jacobian of $\mathbf{h}(\delta_i \mathbf{x})$.

Definition 3.9 (coefficient matrix J based on the Jacobian matrix)**.** The nonlinear iterative system $\mathbf{x}(t+1) = \mathbf{h}\big(\mathbf{x}(t)\big)$ is converted into a linear iterative system $\mathbf{x}(t+1) = J\mathbf{x}(t)$ with a time-varying coefficient matrix J where

$$J = \begin{bmatrix} (1-\beta_1)g_1'(\delta_1 x_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & (1-\beta_n)g_n'(\delta_n x_n) \end{bmatrix} + \frac{1}{n}\begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} \mathbf{1}^\top. \tag{3.14}$$

$g_i'(\delta_i x_i)$ is the learning gain (Definition 2.6, p. 32).

We first show that the state vector $\mathbf{x}(t)$ converges to the solution $\mathbf{x}^* = \mathbf{0}$ if the learning functions $\{g_i(\cdot) : 1 \le i \le n\}$ satisfy Assumption 2.5 (p. 32) and $0 \le \max_i \beta_i < 1$.

Theorem 3.10 (upper bond of spectral radius)**.** *The spectral radius $\rho(J)$ of the coefficient matrix J (Definition 3.9, p. 40) satisfies $\rho(J) \le m \equiv \max_{1\le i\le n, x\in\mathbb{Z}} \big[(1-\beta_i)|g_i'(x)| + \beta_i\big] < 1$.*

Proof. The induced ∞-norm $\|J(\mathbf{x})\|_\infty$ of J satisfies

$$\begin{aligned} \|J\|_\infty &\equiv \max_{\|\mathbf{v}\|_\infty = 1} \|J\mathbf{v}\|_\infty, \\ &= \max_{\|\mathbf{v}\|_\infty = 1} \max_{1\le i\le n} |J_i(\delta_i \mathbf{x})\mathbf{v}|, \\ &= \max_{\|\mathbf{v}\|_\infty = 1} \max_{1\le i\le n} \Big[(1-\beta_i)|g_i'(\delta_i x_i)| v_i + \frac{1}{n}\beta_i(\mathbf{1}^\top \mathbf{v})\Big], \\ &\le m. \end{aligned} \tag{3.15}$$

The result follows from Theorem 3.7 (p. 39) that $\rho(J) \le \|J\|_\infty = m$. It is easy to see that $m < 1$ whenever $\max_{1\le i\le n} \beta_i < 1$. □

Theorem 3.11 (convergence of noiseless soft regulation)**.** *The soft regulation iteration converges to $\mathbf{x}^* \equiv \mathbf{0}$ if $0 \le \beta_i < 1$.*

Proof. Since $m < 1$ for all $0 \le \beta_i < 1$, according to Theorem 3.5 (p. 39), the coefficient matrix J is convergent because $\rho(J) \le m < 1$. □

This proof is also applicable for time-varying g_i. As long as $g_i(\cdot)$ satisfies Assumption 2.5 (p. 32) for $t > T$ and $T < \infty$, the convergence result will hold. Note that $|g'(\cdot)| < 1$ is sufficient but not necessary for $\mathbf{h}$ to converge. A weaker regularity condition $|g(x)/x| < 1$, for instance, also ensures convergence, however, only for $x \neq 0$. We stick to $|g'(\cdot)| < 1$ for the convenience that it applies to the entire domain $x \in \mathbb{Z}$.

3.2 Robustness

In reality, the learning function g_i is subjected to noise because of the noisy payoff $\tilde{f}_i$ or the uncertainty associated with evaluating $\tilde{f}_i$.

Theorem 3.12 (robustness against bounded noises)**.** *The equilibrium fixed point* $\mathbf{x}^* = \mathbf{0}$ *of the map* $\mathbf{h}$ *is robust when subjected to bounded disturbances.*

Proof. Let $V(\mathbf{x}) \equiv \|\mathbf{x}\|_\infty$. Thus,

$$V\big(\mathbf{h}(\mathbf{x})\big) = \|\mathbf{h}(\mathbf{x})\|_\infty = \|J\mathbf{x}\|_\infty \leq \|J\|_\infty \|\mathbf{x}\|_\infty \leq m \|\mathbf{x}\|_\infty . \tag{3.16}$$

The first inequality follows from the property of induced matrix norm. Since the continuous function $V(\mathbf{x})$ is a Lyapunov function for $\mathbf{h}$, the result follows from standard results in stability theory [Teel, 2004]. □

3.3 Efficiency

Assumption 3.13 (representative agent)**.** We will henceforth assume that β_i values are identically equal to the representative degree of social influence β:

$$\beta_i \equiv \beta. \tag{3.17}$$

The representative agent assumption regards a system with heterogeneous agents equivalent to the one with identical agents (i.e., typical or representative agents) if the latter demonstrates the same aggregate behavior as the former. Despite receiving criticisms from economists, the representative agent is among the very few techniques that can model multi-agent systems analytically (i.e., no simulation is performed).

Definition 3.14 (largest singular value σ_1)**.** The largest singular value of a matrix J is its induced 2-norm:

$$\sigma_1(J) \equiv \|J\|_2 \equiv \max_{\|\mathbf{v}\|_2=1} \|J\mathbf{v}\|_2 . \tag{3.18}$$

Definition 3.15 (upper bound of open loop spectral radius m_g)**.** We define m_g as the maximum learning gain:

$$m_g \equiv \max_{1\leq i\leq n, x\in\mathbb{Z}} |g_i'(x)|. \tag{3.19}$$

It is the upper bound of spectral radius m in the open loop setting.

Theorem 3.16 (contraction of the Euclidean norm)**.** *Suppose β_i are all identically equal to β (Assumption 3.13, p. 42). Then $\|\mathbf{h}(\mathbf{x})\|_2 \le m\,\|\mathbf{x}\|_2$, where $m \equiv (1-\beta)\max_{1\le i\le n, x\in\mathbb{Z}}|g_i'(x)| + \beta = (1-\beta)m_g + \beta$.*

Proof. Let $G' \equiv \operatorname{diag}\big(g_1'(\delta_1 x_1), \ldots, g_n'(\delta_n x_n)\big)$. Then $J = (1-\beta)G' + \frac{\beta}{n}\mathbf{1}\mathbf{1}^\top$.

$$\left\|G'\right\|_2^2 \equiv \max_{\|\mathbf{v}\|_2=1}\left\|G'\mathbf{v}\right\|_2^2 = \max_{\|\mathbf{v}\|_2=1}\sum_i |g_i'(\delta_i x)|^2 v_i^2 \le \max_{1\le i\le n}|g_i'(\delta_i x_i)|^2 \le \max_{1\le i\le n, x\in\mathbb{Z}}|g_i'(x)|^2. \tag{3.20}$$

Thus, $\|G'\|_2 \le \max_{1\le i\le n, x\in\mathbb{Z}}|g_i'(x)|$. Therefore,

$$\begin{aligned}
\|J\|_2^2 &\equiv \max_{\|\mathbf{v}\|_2=1}\|J\mathbf{v}\|_2^2, \\
&= \max_{\|\mathbf{v}\|_2=1}\left\{(1-\beta)^2\left\|G'\mathbf{v}\right\|_2^2 + \frac{\beta^2}{n^2}(\mathbf{1}^\top\mathbf{v})^2\|\mathbf{1}\|_2^2 + \frac{2\beta(1-\beta)}{n}(\mathbf{1}^\top\mathbf{v})(\mathbf{1}^\top G'\mathbf{v})\right\}, \\
&\le (1-\beta)^2\left\|G'\right\|_2^2 + \beta^2 + \frac{2\beta(1-\beta)}{n}\big(\max_{\|\mathbf{v}\|_2=1}|\mathbf{1}^\top v|\big)\big(\max_{\|\mathbf{v}\|_2=1}|\mathbf{1}^\top G'\mathbf{v}|\big), \qquad (3.21)\\
&\le (1-\beta)^2\left\|G'\right\|_2^2 + \beta^2 + \frac{2\beta(1-\beta)}{\sqrt{n}}\|\mathbf{1}\|_2\big(\max_{\|\mathbf{v}\|_2=1}\left\|G'\mathbf{v}\right\|_2\big), \\
&= (1-\beta)^2\left\|G'\right\|_2^2 + \beta^2 + 2\beta(1-\beta)\left\|G'\right\|_2 = m^2.
\end{aligned}$$

Since $\mathbf{h}(\mathbf{x}) = J\mathbf{x}$, it follows that $\|\mathbf{h}(\mathbf{x})\|_2 = \|J\mathbf{x}\|_2 \le \|J\|_2\|\mathbf{x}\|_2 \le m\,\|x\|_2$. □

Next, we introduce noise. Let $\{\boldsymbol{\omega}(t) \in \mathbb{R}^n : t \ge 0\}$ denote an IID sequence of random vectors where $\boldsymbol{\omega}(t) = [\omega_1(t), \ldots, \omega_n(t)]^\top$, and each $\omega_i(t)$ is an IID sample of a zero mean random variable with variance σ_ω^2. The noisy collective dynamics is given by

$$x_i(t+1) = (1-\beta_i)\Big(g_i\big(x_i(t)\big) + \omega_i(t)\Big) + \beta_i\frac{1}{n}\sum_j x_j(t). \tag{3.22}$$

We first define the metrics of performance as follows.

Definition 3.17 (stage cost v)**.** We define the mean squared error $v(t) \equiv \mathrm{MSE}(t)$ as the stage cost at time t:

$$v(t) \equiv \frac{1}{n}\sum_i x_i(t)^2 = \frac{1}{n}\|\mathbf{x}(t)\|_2^2. \tag{3.23}$$

Note that MSE is only one of many indicators of performance. Total realized payoff, $\sum_i y_i(t)$ for example, also measures how well a system performs. The state of a system with a better realized payoff, however, is not unambiguously better than another state due to disturbances and measurement noises.

Definition 3.18 (cumulative cost V)**.** The cumulative cost for a finite time horizon T is defined as

$$V(T) \equiv \sum_{t=0}^{T-1} v(t). \tag{3.24}$$

Definition 3.19 (optimal control)**.** We define the following optimal control problem for computing the optimal degree of social influence β that minimizes the expected cost function V:

$$\begin{aligned} \min_{\beta} \quad & \mathbb{E}\Big[V\big(\beta; \mathbf{x}(0), \boldsymbol{\omega}(t), T\big)\Big], & (3.25)\\ \text{s. t.} \quad & \mathbf{x}(t+1) = \mathbf{h}\big(\mathbf{x}(t)\big), & (3.26) \end{aligned}$$

Based on our previous results, we have the follow recursion of the stage cost

$$\begin{aligned} \mathbb{E}\big[v(t+1) \mid \mathbf{x}(t)\big] &= \frac{1}{n}\mathbb{E}\Big[\, \|\mathbf{x}(t+1)\|_2^2 \mid \mathbf{x}(t)\Big], & (3.27)\\ &= \frac{1}{n}\mathbb{E}\Big[\, \|\mathbf{h}(\mathbf{x}(t)) + (1-\beta)\boldsymbol{\omega}(t)\|_2^2 \mid \mathbf{x}(t)\Big], & (3.28)\\ &= \frac{1}{n}\, \|\mathbf{h}(\mathbf{x}(t))\|_2^2 + \frac{(1-\beta)^2}{n}\mathbb{E}\big[\|\boldsymbol{\omega}(t)\|_2^2\big], & (3.29)\\ &\le \frac{1}{n}\, \|J\|_2^2\, \|\mathbf{x}(t)\|_2^2 + (1-\beta)^2\sigma_\omega^2, & (3.30)\\ &\le m^2 v(t) + (1-\beta)^2\sigma_\omega^2, & (3.31) \end{aligned}$$

where (3.29) follows from the fact that $\boldsymbol{\omega}(t)$ is independent of $\mathbf{x}(t)$, and (3.30) follows from the bound in Theorem 3.16 (p. 43). Iterating the bound (3.31) we get

$$\mathbb{E}[v(t)] \le m^{2t} v(0) + \frac{(1-\beta)^2(1-m^{2t})}{(1-m^2)}\sigma_\omega^2. \tag{3.32}$$

Definition 3.20 (steady-state mean squared error v_∞)**.** The mean squared error for a large t can be approximated by equating the LHS and the RHS of (3.32):

$$\begin{aligned} v_\infty &\equiv \lim_{t\to\infty} \mathbb{E}\big[v(t)\big],\\ &\approx \frac{(1-\beta)^2\sigma_\omega^2}{1-m^2},\\ &= \frac{(1-\beta)\sigma_\omega^2}{\big(1-m_g\big)\big[\beta(1-m_g)+m_g+1)\big]}. \end{aligned} \tag{3.33}$$

Note that v_∞ decreases monotonically as β increases.

We can then obtain the upper bound on the expected total cost easily:

$$
\begin{aligned}
\mathbb{E}\big[V(T)\big] &\leq \frac{1-m^{2T}}{1-m^2}v(0) + \left(T - \frac{1-m^{2T}}{1-m^2}\right)\frac{(1-\beta)^2\sigma_\omega^2}{1-m^2}, \\
&= \frac{1-m^{2T}}{1-m^2}\big(v(0) - v_\infty\big) + Tv_\infty.
\end{aligned}
\tag{3.34}
$$

Solving the β^* that minimizing RHS of (3.34) gives a conservative measure of the optimal degree of social influence.

3.4 Optimal Degree of Social Influence

We briefly discussed how β affects the efficiency of soft regulation in Section 3.3 (p. 42). By minimizing the RHS of (3.34), one can obtain the optimal β that minimizes the worst case $\mathbb{E}\big[V(T)\big]$, i.e., expected cumulative MSE. In the following segments, we will look into two aspects of how β affects the performance of soft regulation.

3.4.1 Maximum Contraction

We established in (3.30) that the reduction of MSE is controlled by the contraction property of J (as well as the noise reduction term $(1-\beta)^2\sigma_\omega^2$, which we will discuss in the next segment). More specifically, the contraction determines how MSE decreases in the absence of noise. We used the induced 2-norm (Definition 3.14, p. 42) of J to quantify such contraction in the context where MSE is the metric of performance.

It is apparent that the smaller the spectral radius, the faster $\mathbf{x}$ goes to $\mathbf{0}$. The contraction property can then be approximated by the spectral radius of J. One can interpret the contraction property as this:

$$
\frac{\|J\mathbf{x}\|}{\|\mathbf{x}\|} \leq \rho(J) \quad (\mathbf{x} \neq \mathbf{0}). \tag{3.35}
$$

In (3.35), the contraction is not state ($\mathbf{x}$) dependent. That simplifies the analysis because for any state $\mathbf{x} \neq \mathbf{0}$ at any time t, (3.35) would hold and the spectral radius only depends on J (instead of depending on both J and $\mathbf{x}$). In contrast, the actual contraction, i.e., the LHS of (3.35) can only be measured after the dynamics (3.1) takes place. Similarly, we have

$$
\frac{\|J\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \sigma_1(J) \quad (\mathbf{x} \neq \mathbf{0}), \tag{3.36}
$$

where σ_1 is the largest singular value (Definition 3.14, p. 42).

The optimal β that maximizes contraction would therefore be the one that minimizes either the spectral radius or the largest singular value of J. Even though Theorem 3.10 (p. 41) and Theorem 3.16 (p. 43) suggest that increasing β monotonically increases the *upper bond* of $\rho(J)$ and $\sigma_1(J)$, the actual relationship is more complex.

Conjecture 3.21 (maximum contraction)**.** For a coefficient matrix $J(\beta) = (1-\beta)G' + \beta S$ (Definition 3.9, p. 40) as a function of β, given $\sigma_g \neq 0$ (Definition 2.7, p. 32), there exists a $\beta^* > 0$ such that ρ (or σ_1) is minimized. In other words, there exists an optimal degree of social influence that maximally improves the *noiseless* contraction of soft regulation.

We do not attempt a proof for this Conjecture. Instead, we demonstrate here a simple system with $G' = \text{diag}\big(-0.68, -0.029, 0.60, 0.91, 0.94\big)$. When $\beta = 0$, $J = G'$ thus $\rho(J) = 0.94$. As β increases, $\rho(J)$ first decreases then increases until $\rho(J) = 1$ when $\beta = 1$. A few $\rho(J)$ values against different β values are listed in Table 3.1 (p. 47). Interested readers can use the following MATLAB code to recreate the results:

```
>> g = [-0.68; -0.029; 0.60; 0.91; 0.94];
>> rho = @(beta) max(abs(eig((1-beta) * diag(g) + beta * ones(5) / 5)));
>> sv = @(beta) norm((1-beta) * diag(g) + beta * ones(5) / 5, 2);
>> beta_best_rho = fminbnd(rho, 0, 1)
beta_best_rho =
    0.44
>> beta_best_sv = fminbnd(sv, 0, 1)
beta_best_sv =
    0.44
```

3.4.2 Maximum Noise Reduction

The noise reduction effect from β is much more straightforward than the contraction effect. Using expected MSE as the performance metric leads to the following noise term in (3.30):

$$(1-\beta)^2\sigma_\omega^2. \tag{3.37}$$

Table 3.1: Illustration of Conjecture 3.21

β	$\rho(J)$	$\sigma_1(J)$
0	0.94	0.94
0.1	0.88	0.88
0.2	0.84	0.84
0.3	0.81	0.81
0.4	0.80	0.80
0.44	**0.79**	**0.79**
0.5	0.80	0.80
0.6	0.82	0.82
0.7	0.85	0.85
0.8	0.89	0.89
0.9	0.94	0.94
1	1.0	1.0

As a result, increasing β makes the noise reduction stronger and leads to a smaller steady-state error (Definition 3.20, p. 44). β, however, cannot increase indefinitely because a large β would hurt the contraction as discussed earlier. For a system where $\sigma_\omega \gg \sigma_g$, we can safely use the worst case formulation (3.34) to identify the optimal β^* (Fig. 3.1, p. 48).

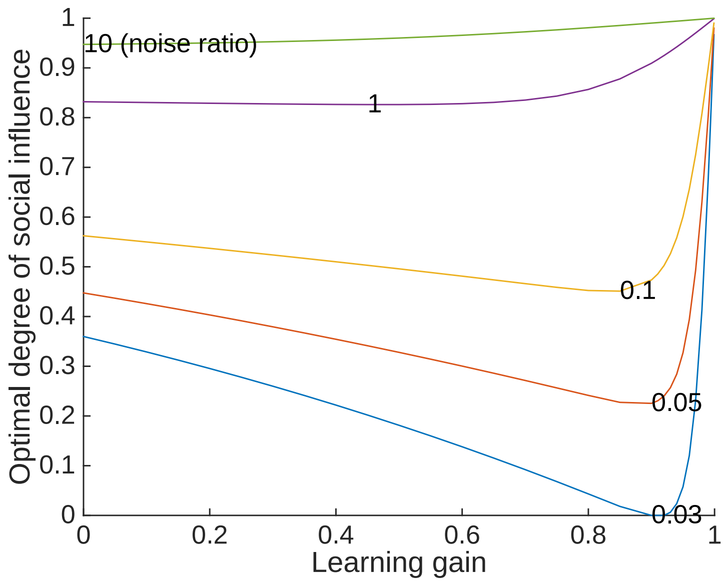


Figure 3.1: **Optimal social influence from robust control by minimizing the RHS of** (3.34). The general trend is that a moderately strong social influence is desirable if the system is uncertain (high noise-to-initial-MSE ratio) or the learning gain is low (fast open loop convergence). An interesting observation is that as the learning gain crosses a certain threshold (e.g., 0.9), the optimal social influence rapidly increases as the learning gain increases. For a high learning gain, the contraction becomes insensitive to the change in β while the noise reduction still does.

3.5 Different Solutions

There are situations where the solutions are similar but not identical to all agents. We discuss here how soft regulation performs under such circumstance.

Definition 3.22 (solution spectrum $\boldsymbol{\theta}^*$ and its width σ_θ)**.** Similar to the naming convention for the learning spectrum (Definition 2.7, p. 32), we let $\boldsymbol{\theta}^* \equiv [\theta_1^*, \ldots, \theta_n^*]^\top$ denote the solution spectrum and $\sigma_\theta \equiv \sqrt{\frac{1}{n}\left(\theta_i^* - \theta^*\right)^2}$ denote the width of the solution spectrum, where $\theta^* \equiv \frac{1}{n}\sum_j \theta_j^*$ is the average (or representative) solution.

Consider the open loop learning dynamics:

$$\bar{x}_i(t) = g_i\big(x_i(t)\big) + \omega_i(t). \tag{3.38}$$

Replacing the LHS $\bar{x}_i$ with $\bar{z}_i - \theta_i^*$, we have

$$\bar{z}_i(t) = g_i\big(x_i(t)\big) + \omega_i(t) + \theta_i^*, \tag{3.39}$$

and subsequently we also have

$$\begin{aligned} z_i(t+1) &= (1-\beta)\bar{z}_i(t) + \beta\frac{1}{n}\sum_j z_j(t), \\ &= (1-\beta)\Big(g_i\big(x_i(t)\big) + \omega_i(t) + \theta_i^*\Big) + \beta\frac{1}{n}\sum_j \big(x_j(t) + \theta_j^*\big). \end{aligned} \tag{3.40}$$

Subtracting θ_i^* from both LHS and RHS, we have

$$x_i(t+1) = (1-\beta)\Big(g_i\big(x_i(t)\big) + \omega_i(t)\Big) + \beta\frac{1}{n}\sum_j x_j(t) + \beta\frac{1}{n}\sum_j(\theta_j^* - \theta_i^*). \tag{3.41}$$

We can convert (3.41) into a linear iterative system by replace $g_i(x)$ with $\tilde{g}_i' x$ where the learning gain (Definition 2.6, p. 32):

$$\mathbf{x}(t+1) = J\mathbf{x}(t) + (1-\beta)\boldsymbol{\omega} + \beta(S-I)\boldsymbol{\theta}^*, \tag{3.42}$$

where $J = (1-\beta)G' + \beta S$ is identical to what we defined earlier (Definition 3.9, p. 40) with $G' = \operatorname{diag}\left(\mathbf{g}'\right)$ and $S = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. It is not difficult to extend (3.30) and obtain the following inequality:

$$\mathbb{E}\big[\,\|\mathbf{x}(t+1)\|_2\,|\mathbf{x}(t)\big] \le \sigma_1(J)\,\|\mathbf{x}(t)\|_2 + (1-\beta)\sqrt{n}\sigma_\omega + \sqrt{n}\sigma_\theta. \tag{3.43}$$

Definition 3.23 (root mean squared error (RMSE) v(t))**.** We define the root mean squared error of the system as the stage cost $v(t) = \frac{1}{\sqrt{n}} \|\mathbf{x}(t)\|$.

The RMSE version of (3.43) is therefore

$$\mathbb{E}\big[v(t+1)|v(t)\big] \leq \sigma_1(J)v(t) + (1-\beta)\sigma_\omega + \beta\sigma_\theta. \tag{3.44}$$

The steady-state expected RMSE (similar to Definition 3.20, p. 44) is therefore bounded by the following equation:

$$v_\infty \equiv \lim_{t\to\infty} \mathbb{E}\big[v(t)\big] \leq \frac{(1-\beta)\sigma_\omega + \beta\sigma_\theta}{1-\sigma_1(J)}. \tag{3.45}$$

Should the deviation of solution (i.e., the width of the solution spectrum) be sufficiently small (in other words, $\sigma_\theta \leq \sigma_\omega$), there would exist a $\beta^* \in (0,1)$ such that the *worst case* steady-state RMSE is minimized.

Chapter 4

Implementing Soft Regulation

4.1 Regulating Emerging Industries (p. 53) The term soft regulation was first created in the context of regulating emerging industries (Section 2.1, p. 24), where one needs to properly balance between being too conservative (i.e., over-regulation) and being too liberal (i.e., under-regulation). In this section, we use an agent-based simulation to model the collective optimization process of an emerging technology. We discuss how soft regulation could improve the overall welfare of the system by accelerating the convergence.

4.2 Mobile Health (p. 65) We focus here on healthcare. Mobile fitness tracking services have been integrated into many's daily lives: We often change our health behaviors such as sleep, dietary, and exercise patterns in order to improve fitness. Can we use soft regulation to help multiple consumers identify the optimal health behaviors? We present the results from an Amazon Mechanical Turk experiment with human subjects. We analyze experiment data and estimate, through control theory, the optimal degree of social influence that would maximally improve the open loop problem-solving process.

4.3 Local Regulation and Policy: U.S. State Tax and Expenditure Case Study (p. 76) We return to the topic of optimal regulation and policy. In this case study, instead of a federal oversight or a central planner (Section 4.1, p. 53), local regulators and policymakers form a decision-making crowd. Our optimal control analysis indicates that soft regulation could have accelerated the convergence of state tax and expenditure.

4.4 Multi-Product Revenue Management (p. 86) Even if individual solutions are different, soft regulation can still improve the collective problem-solving process. This discovery is applicable to cases such as multi-product revenue management. We propose here an improved algorithm based on the Besbes-Zeevi dynamic pricing model.

4.1 Regulating Emerging Industries

4.1.1 Background

Consider the following situation: There are one central planner (regulator) and n regulated entities (companies, $i = 1, \ldots, n$) in an emerging industry. Since the technology or business has just been discovered or developed, there lacks an industry standard that properly balances production, stability, and environmental impact. The optimal operating standard θ^* is unknown to either the regulator or the companies. As more time and resources are devoted to R&D, individual companies can gradually improve their operations and eventually discover the optimum *independently*. In the meantime, the regulator periodically issues recommendations (instead of mandates). The companies can accept, reject, or partially accept the recommendations based on how confident they feel towards the regulator. The confidence level β is an alternative form of the degree of social influence present in the system. In this exercise, we implement an agent-based simulation to understand how 1) different types of recommendations and 2) different confidence levels affect the performance of soft regulation.

4.1.2 Agent-Based Simulation

There are several ways to generate feedbacks for soft regulation. We focus on two methodologies here: the *best recommendation* and the *crowd recommendation*. As their names suggest, best recommendation corresponds to the case where the regulator has *full information* and computes the feedback by solving a *centralized* optimal control problem; the crowd recommendation on the other hand, is simply the average of the agents' actions (i.e., the wisdom of crowds). We show that, despite its simplicity, crowd recommendation is as good as the best recommendation for a wide range of confidence levels.

When the regulator is fully informed about the functions $\tilde{f}_i$, $\tilde{g}_i$, and β_i (Section 2.1.2, p. 29), the optimal feedback u^* can be computed explicitly by solving the following centralized optimal control problem that maximizes social welfare (cumulative payoff) over the projected trajectory:

$$\begin{aligned} \max \quad & \sum_{t=0}^{T-1} \sum_{i=1}^{n} \mathbb{E}\Big[\tilde{f}_i\big(z_i(t)\big)\big|u(\cdot)\Big], \\ \text{s.t.} \quad & z_i(t+1) = (1-\beta_i)\tilde{g}_i\big(z_i(t)\big) + \beta_i u(t). \end{aligned} \tag{4.1}$$

We call the solution u^* to this problem the best recommendation.

Since the function $\tilde{f}_i$, $\tilde{g}_i$, and the parameter β_i are only privately known to the agents, in practice, it is unlikely that the regulator knows the functions and the parameters. Following [Brotman, 2014; Surowiecki, 2005; Sunstein and Hastie, 2014], we assume that the regulator, instead, reports the average, i.e., $u \equiv \frac{1}{n}\sum_i z_i$. We call this recommendation the crowd recommendation. Note that using privacy-preserving computations [Abbe *et al.*, 2012], the regulator can compute the crowd recommendation without ever learning any individual input z_i. We demonstrate that the crowd recommendation ensures the convergence to the optimal setpoint; moreover, it is as good as the best recommendation for a wide range of confidence levels.

We have established the convergence and robustness properties of soft regulation in the previous chapter. For practical applications, it is important to understand the transient or *finite-time* dynamics of soft regulation, and more specifically, the role of confidence level β_i in setting the transient performance. We are able to illustrate with a few additional assumptions about the system.

In this section, our analysis will be focused on the *simplest* concave function, namely an identical and quadratic utility function $\tilde{f}_i(z_i) = \tilde{f}(z_i) = f(z_i) + \nu = -k(z_i - \theta^*)^2 + \nu$. Representative agents are helpful in identifying the effect of confidence level (Assumption 3.13, p. 42). In order to study the convergence behavior, one can without loss of generality, assume that $\theta^* = 0$ (i.e., decision error $x_i = z_i$). This particular choice for f is motivated by the fact that any strongly concave function can be approximated by a quadratic function near its optimum. The noise is $\nu \sim \mathcal{N}(0, \sigma_\nu)$. Agents only observe the noisy function values — the underlying structure is not known to the agents.

Definition 4.1 (optimization efficiency η)**.** We define the optimization efficiency as the percent reduction in MSE (or stage cost v, Definition 3.17, p. 43):

$$\eta(t) \equiv \frac{v(0) - v(t)}{v(0)} \times 100\%. \tag{4.2}$$

The efficiency is 100% when the system reaches optimum.

We simulated the agent dynamics in NetLogo (Appendix A, p. 104). The agent set was randomized by a fixed random seed in the program to ensure consistency. For each set of parameters, we ran the simulation five times and took the average. The observed standard error is insignificant; therefore we omit error bars in the figures. The model parameter values are listed in Table 4.1 (p. 55). The noise $\nu \sim \mathcal{N}(0, \sigma_\nu)$ has the same variance as another random variable $\epsilon \sim \mathcal{U}(-\sqrt{3}\sigma_\nu, \sqrt{3}\sigma_\nu)$. We chose σ_ν to be $200/\sqrt{3}$ so that it is computationally equivalent to a uniform ± 200 noise. The parameters do not represent practical meanings. The particular values were chosen such that the results are easily identifiable.

Table 4.1: Agent-based simulation parameters

n	σ_ν	θ^*	k	$a(t)$	$c(t)$
1000	$200/\sqrt{3}$	0	100	$1/t$	$1/(t+200)^{1/3}$

We first run the simulation for soft regulation with best recommendation. Given the quadratic utility, Kiefer-Wolfowitz algorithm (p. 29), and system-wide confidence level, the regulator can easily compute best recommendation by solving the optimal control problem (4.1). One can obtain the noiseless system dynamics to be

$$\mathbf{x}(t+1) = (1-\beta)\big(1 - 4ka(t)\big)\mathbf{x}(t) + \beta \mathbf{1} u(t). \tag{4.3}$$

Since the stage cost does not penalize input u, the optimal u^* at stage t can be solved as follows

$$u^*(t) = -\left[\frac{(1-\beta)\big(1 - 4ka(t)\big)}{\beta}\right] \cdot \frac{1}{n} \sum_i x_i(t). \tag{4.4}$$

In Fig. 4.1 (p. 57), we plot the efficiency η after 200 iterations against different confidence levels β. We observe the efficiency increases monotonically as the confidence level increases.

This result is not surprising. As the confidence level increases, the regulator has a stronger influence on the agents, therefore, exerting a more efficient control. Even though for each confidence level, the regulator issues the best recommendation, the recommendation is only effective when the agents choose to listen.

In Fig. 4.2 (p. 57) and Fig. 4.3 (p. 62), we plot the efficiency against confidence level for soft regulation with crowd recommendation. The results from Fig. 4.1 (p. 57) are also included as a reference. It is remarkable that soft regulation with crowd recommendation is as good as the one with best recommendation for a wide range of confidence levels (from 0 to 99%). The real advantage of best recommendation only appears when the confidence level is close to 100%. However, to achieve such best recommendation or even hard regulation, the regulator needs information about utility function, optimization algorithm, and the confidence level. This practice, despite being efficient under the setting of complete information, is costly, impractical, and error prone in practical settings. Especially for hard regulation, additional cost of enforcement needs to be considered.

The results in Fig. 4.2 (p. 57) and Fig. 4.3 (p. 62) indicate that the confidence level should be set to a large value but not too close to 100%. The open loop system only reaches about 70% optimum. The system performance is more than 90% optimal when the confidence level is 50% (i.e., the agent takes an average between its own optimization result and the recommendation). We also see a sharp decline in performance when confidence level is too close to 100%. Beyond this "cliff," the agents explore very little and essentially stay where they are.

In Fig. 4.4 (p. 62), we plot the time series of efficiency for different confidence levels. When confidence level is low ($\beta = 0$ or 10%), the MSE increases (efficiency declines) before converging. This is caused by large initial step sizes. As confidence level increases, the system begins convergence earlier. As the confidence level further increases, the system shifts from the regime dominated by exploration to the one dominated by conformity, and the recommendation does not have enough time to converge to optimum before agents start conforming.

In order to better understand the connections between the confidence level and the performance that we hypothesized in the previous section and observed in the simulation

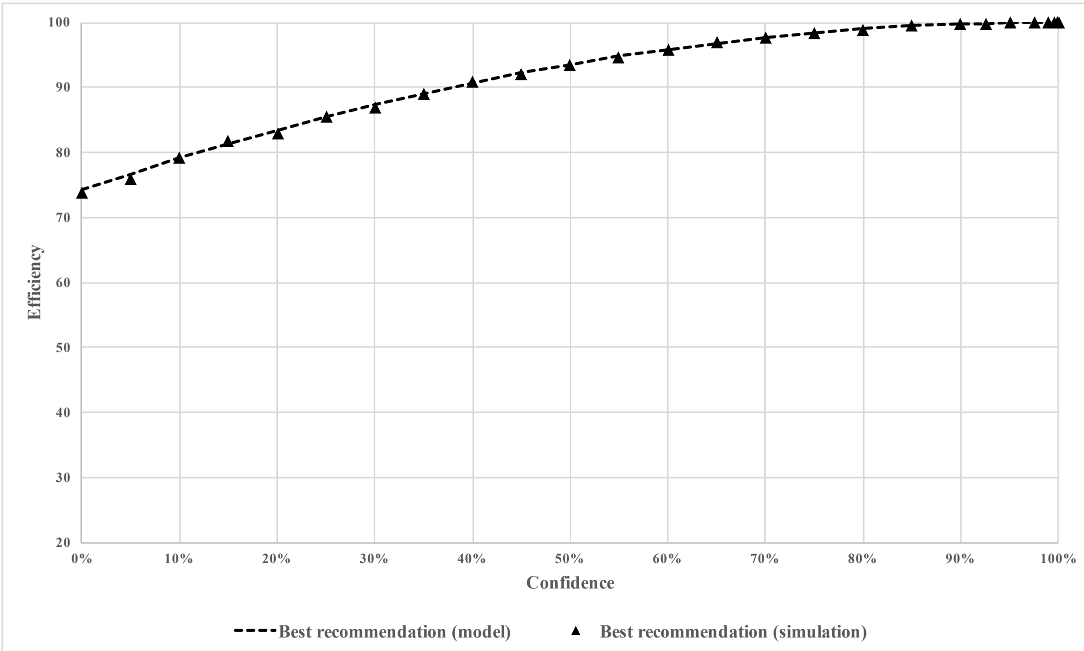


Figure 4.1: **Efficiency of soft regulation with best recommendation**.

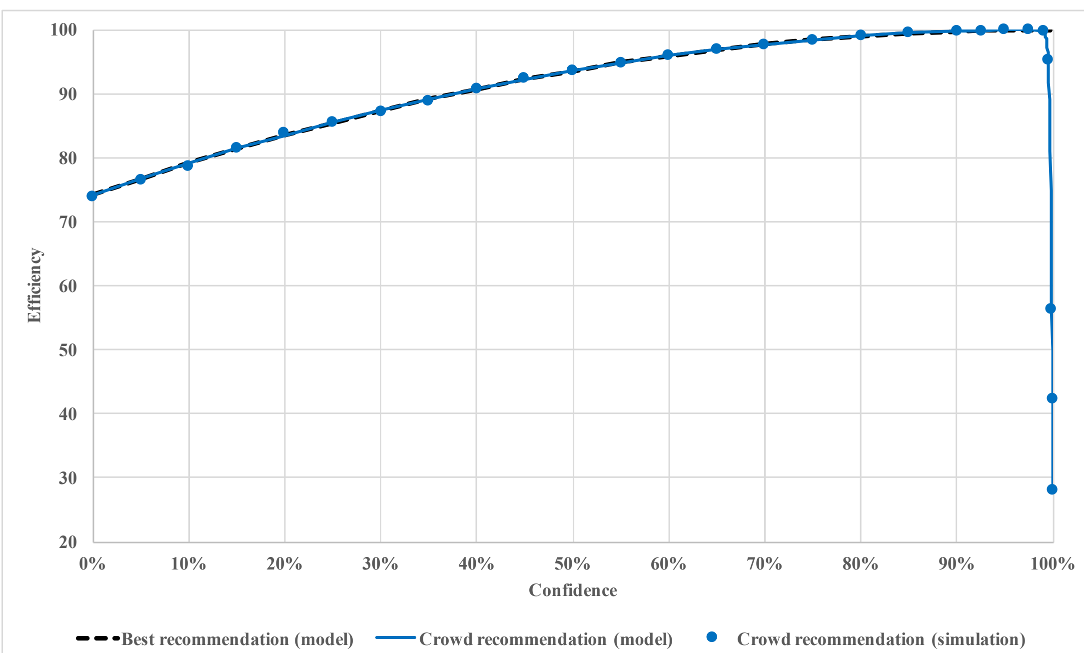


Figure 4.2: **Efficiency of soft regulation with crowd recommendation**.

results, we now attempt to compute a *closed-form* expression for the system state. Recall that x_i denotes the current state of the i-th agent. The updated state x_i^+ implied by the gradient-based update scheme in (2.2) is given by

$$x_i^+ = (1-\beta)(1-4ka)x_i + \beta u + a \cdot \frac{1-\beta}{c} \cdot \hat{\nu}_i \tag{4.5}$$

where $\hat{\nu}_i \sim \mathcal{N}(0, \sqrt{2}\sigma_\nu)$ is the effective noise resulting from computing the discrete approximation to the gradient in (2.2) on p. 29. Recall that the crowd recommendation is $u = \sum_i x_i/n$. Thus, the updated u^+ of the recommendation is given by

$$u^+ = \Big[1 - 4ka(1-\beta)\Big]u + a \cdot \frac{1-\beta}{c} \cdot \frac{1}{n}\sum_i \hat{\nu}_i. \tag{4.6}$$

When $n \gg 1$, the expected $u(t)$ can be treated as a deterministic variable. When β is small, $u(t)$ quickly converges to 0. Let $U(t) \equiv \mathbb{E}[u(t)]$. Since $\mathbb{E}[\hat{\nu}] = 0$, we have that

$$U(t+1) - U(t) = -4ka(t)(1-\beta)U(t). \tag{4.7}$$

Note that the variance of $u(t+1)$ is $2a(t)^2(1-\beta)^2\sigma_\nu^2/\big(c(t)^2 n\big) \ll 1$ for $n \gg 1$. Therefore, compared to $x_i(t)$, the recommendation $u(t)$ can be safely treated as a deterministic variable i.e., $u(t) \approx U(t)$. Especially when $|4ka(t)(1-\beta)| \ll 1$, one can approximate the difference equation by the ODE

$$\frac{\mathrm{d}U(t)}{\mathrm{d}t} = -4ka(t)(1-\beta)U(t). \tag{4.8}$$

$\mathbb{E}[u(t)]$ can be approximated as follows:

$$\mathbb{E}[u(t)] \approx u(0)\exp\left[-4k(1-\beta)\int_0^t a(\tau)\mathrm{d}\tau\right]. \tag{4.9}$$

For the wisdom of crowds, this implies that a group is smart only when the population is large ($n \gg 1$) and agents are not strongly conforming ($\beta \ll 100\%$). Surowiecki's book [Surowiecki, 2005] shares the same insights. Unlike the averaging method in the book, soft regulation is a continuous feedback process. Even though the open loop ($\beta = 0$) system has the fastest converging recommendation, agents cannot make use of it unless they at least partially accept ($\beta > 0$). This paradox suggests some trade-off and balancing between consensus and efficiency.

In addition, as k increases, $u(t)$ approaches the solution faster, i.e., a more sensitive utility function implies a more reliable recommendation. Unless an agent can estimate the curvature ($\sim k$ for a quadratic function) of the payoff accurately, it is safer to rely on the recommendation when curvature is larger.

From (4.5) it follows that

$$\begin{aligned}
v(t+1) \equiv & \frac{1}{n}\sum_i x_i^2(t+1),\\
=&(1-\beta)^2\big(1-4ka(t)\big)^2\frac{1}{n}\sum_i x_i^2(t) + \beta^2 u(t)^2\\
&+2(1-\beta)\big(1-4ka(t)\big)u(t)\frac{1}{n}\sum_i x_i(t) + \frac{1}{n}a(t)^2\frac{(1-\beta)^2}{c(t)^2}\sum_i \hat{\nu}_i^2(t)\\
&+\frac{1}{n}\sum_i\Big[(1-\beta)\big(1-4ka(t)\big)x_i(t)+\beta u(t)\Big]a(t)\frac{(1-\beta)}{c(t)}\hat{\nu}_i(t).
\end{aligned} \tag{4.10}$$

Since $u \equiv \sum_i x_i/n$, $u(t) \approx U(t)$, and $\mathbb{E}[\hat{\nu}_i] = 0$, we have

$$\begin{aligned}
\mathbb{E}\left[v(t+1)\right] =&(1-\beta)^2\big(1-4ka(t)\big)^2\mathbb{E}\left[\frac{1}{n}\sum_i x_i^2(t)\right]\\
&+\beta\Big[2(1-\beta)\big(1-4ka(t)\big)+1\Big]U(t)^2\\
&+2a(t)^2\frac{(1-\beta)^2}{c(t)^2}\sigma_\nu^2.
\end{aligned} \tag{4.11}$$

We therefore define $X(t) \equiv \mathbb{E}[v(t)]$. It follows that the update for $X(t)$ is given by recursion

$$X(t+1) = A(t)X(t) + B(t), \tag{4.12}$$

where

$$A(t) \equiv (1-\beta)^2\big(1-4ka(t)\big)^2, \tag{4.13}$$

and

$$B(t) \equiv \beta\Big[2(1-\beta)\big(1-4ka(t)\big)+1\Big]U(t)^2 + 2a(t)^2\frac{(1-\beta)^2}{c(t)^2}\sigma_\nu^2. \tag{4.14}$$

Since $\lim_{t\to\infty} A(t) = A_\infty$, $\lim_{t\to\infty} B(t) = B_\infty$, and $\lim_{t\to\infty} X(t+1) = B_\infty/(1-A_\infty)$, one would expect that when t is large,

$$X(t+1) \approx \frac{B(t)}{1-A(t)}, \tag{4.15}$$

$$= \frac{2a(t)^2(1-\beta)^2\sigma_\nu^2/c(t)^2 + \beta\Big[2(1-\beta)\big(1-4ka(t)\big)+1\Big]U(t)^2}{1-(1-\beta)^2\big(1-4ka(t)\big)^2}. \tag{4.16}$$

This approximation in fact agrees with the simulation (see lines in all simulation result figures).

From (4.16), it follows that MSE converges to 0 (soft regulation is optimal). Meanwhile, when confidence level is low, i.e., $\beta \approx 0$ (explorers), the dependence on $u(t)$ vanishes very quickly, and (4.16) can be simplified as follows:

$$\mathbb{E}[v(t+1)] \approx \frac{2a(t)^2\sigma_\nu^2/c(t)^2}{1/(1-\beta)^2 - \big(1-4ka(t)\big)^2}, \tag{4.17}$$

and MSE monotonically *decreases* as β increases. On the other hand, when confidence level is high, i.e., $\beta \approx 100\%$ (followers), the $u^2(t)$ term dominates σ_ν^2, one can simplify (4.16) to

$$\mathbb{E}[v(t+1)] \approx \beta \left\{ u(0) \exp\left[-4k(1-\beta) \int_0^t a(\tau)\mathrm{d}\tau \right] \right\}^2, \tag{4.18}$$

and MSE monotonically *increases* as β increases. This estimation agrees well with our previous hypotheses and simulation results (Fig. 4.2, p. 57 to Fig. 4.4, p. 62).

An interesting fact arises from this approximation, i.e., imperfect information is necessary for soft regulation to add value. If the system has very low noise or noise-free, the σ_ν^2 term will be dominated by $u^2(t)$, and an increase in β hurts performance. That is to say, *for a deterministic process, soft regulation with crowd recommendation may not be a good mechanism for agents to adopt.*

In practice, each individual may have a distinct confidence level and personal traits (as opposed to Assumption 3.13, p. 42). Modeling such rich details as well as formulating related best recommendation is beyond the scope of this exercise. Nevertheless, for purposes of illustration, we propose the following adaptive confidence mechanism:

$$\beta(x_i, u) = e^{-b(x_i - u)^2} \quad (b > 0). \tag{4.19}$$

The rationale for this update scheme is as follows. When an agent's action is far away from the recommendation, the agent is fairly skeptical. Suppose, by incorporating the recommendation, the agent's action moves further away from the recommendation, the agent would rely even less on the regulator. However, when the action comes closer to the recommendation, agent is likely to be more confident about the regulator, and incorporate the recommendation in future updates. One flaw in this adaptive mechanism is that if everyone performs the same action in the beginning, this results in an identical confidence level $\beta = 100\%$ for all agents, and the system will not move at all. This situation might be remedied with an occasional, small perturbation. In Fig. 4.5 (p. 64), we plot two new simulation results, i.e., 1) agents have uniformly distributed confidence levels (*Dist.*), and 2) agents have uniformly distributed initial confidence levels and the confidence is adaptive (*Dist.+adap.*) according to the update scheme above. We also include previous results with fixed and identical confidence level to the graph. We observe a fairly good performance.

4.1.3 Discussion

In the soft regulation setting, the regulator's role is to help agents learn, understand, and optimize an unknown process without interrupting normal operations. The essence of this mechanism is to take guided decisions by updating actions using the map $x_i^+ = (1 - \beta_i)\bar{x}_i + \beta_i u$, where $\bar{x}_i$ is the i-th agent's open loop action and u is the regulator's soft feedback (or crowd recommendation). Self-interested agents have the freedom to choose to partially accept the regulator's recommendation. Soft regulation provides a more balanced coordination: Unlike hard regulation, it does not force the agents; this creates a collective learning environment and avoids possibly erroneous mandates. On the other hand, a soft regulatory system is not under-regulated or uncoupled. The exploration of some agents benefits others. Useful information is shared indirectly instead of being wasted in an asocial learning environment.

We notice the efficiency of soft regulation is impacted by the following factors:

Population Because of noise, recommendation is subject to uncertainty. However, when $n \gg 1$, the variance becomes negligible, and the recommendation becomes deterministic (very close to mean) and accurate. This dependence on population size is intuitive: The information aggregated from a large population should be more useful compared to the one from a small population.

Process We have proved that soft regulation with crowd recommendation preserves optimality. The advantage of the mechanism, however, is especially pronounced when the system is very noisy and the payoff function is very sensitive. A rule of thumb for the

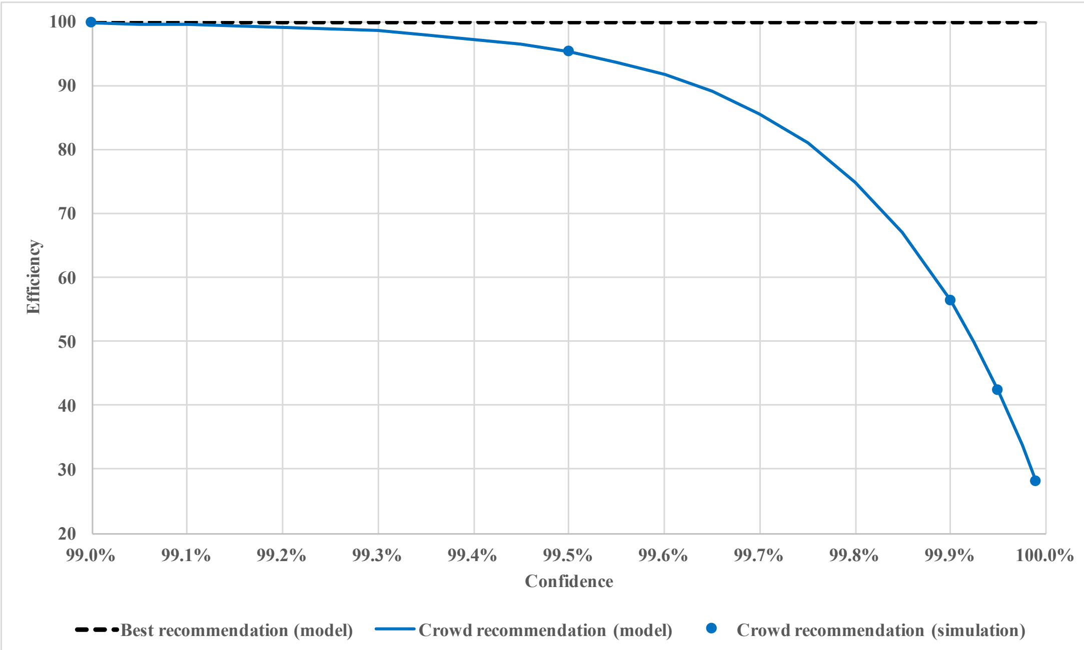


Figure 4.3: **Efficiency of soft regulation with crowd recommendation (large confidence levels).**

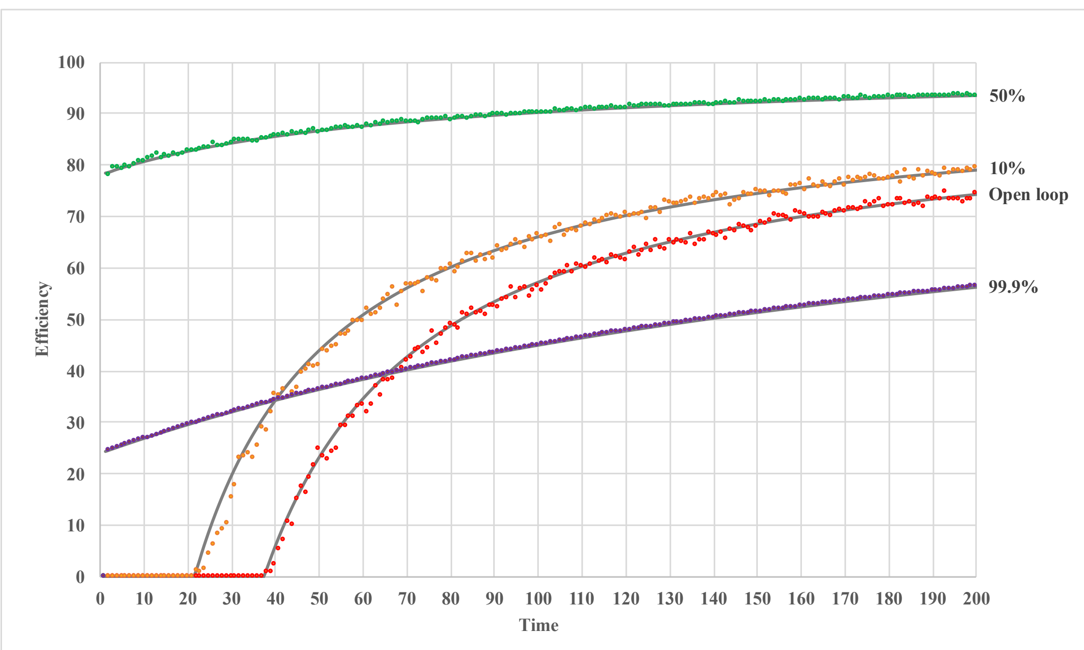


Figure 4.4: **Efficiency of soft regulation with crowd recommendation over time.**

agents would be when a large sensitivity of the process is observed (either because of high noise or large curvature), the agents may be better off relying on the recommendation. Uncertainty drives the system towards cooperation. Soft regulation can potentially stabilize an open loop unstable process. This result also provides some insights on the wisdom of crowds. For example, the average performance can outperform the best individual when the system is very uncertain. In that sense, the "expert" knowledge may not be as useful in an emerging industry as the collective wisdom.

Confidence level β We conclude that the best confidence level should be large but not too close to 100%. This is especially true when the system is very noisy and the process is very sensitive. In such setting, agents should put a substantial amount of trust on the regulator's recommendations. Because of the trade-off between consensus and efficiency, in the early stage of soft regulation, the confidence level should be kept low for recommendation to quickly converge. As time proceeds, agents can be more and more confident regarding the recommendation.

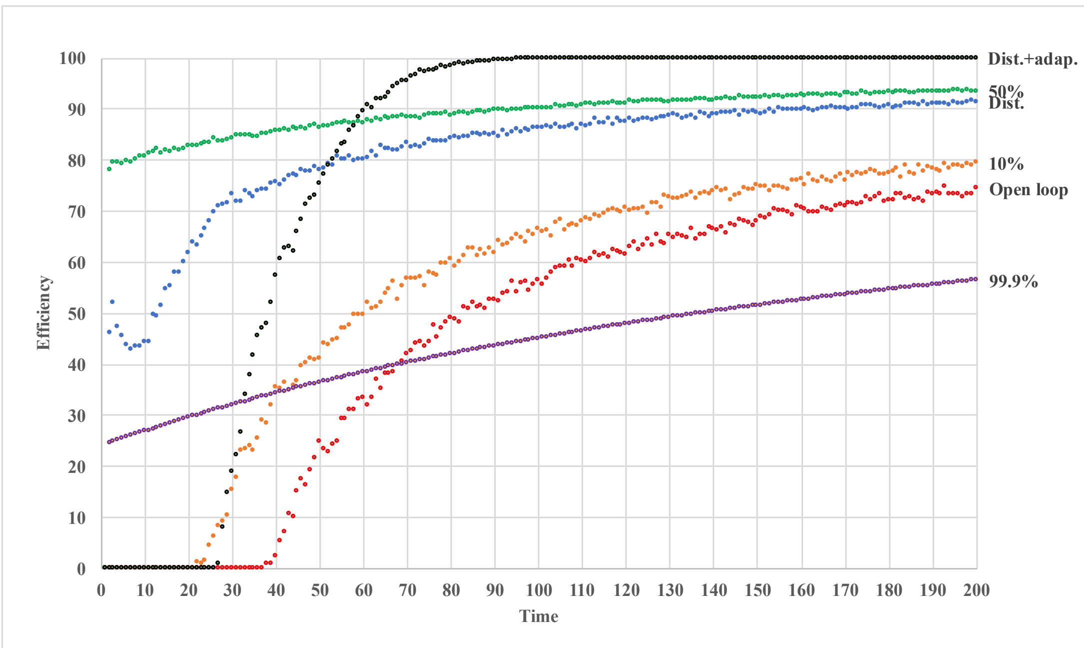


Figure 4.5: **Efficiency of soft regulation with crowd recommendation over time (distributed agents)**.

4.2 Mobile Health

4.2.1 Background

There are n consumers of certain fitness service such as Fitbit or Apple Watch. They form a social network. Each individual consumer attempts to adjusts his/her health behavior z_i in terms of sleep, dietary, or exercise pattern such that his/her fitness y_i is maximized. S/he can also poll the social network to see the others' behaviors (or the population average). Based on the degree of social influence β_i, the i-th consumer adjusts his/her decisions.

4.2.2 Experimental Design

We conducted three sets of online experiments on Amazon Mechanical Turk with human subjects (Appendix B, p. 112). We focus our analysis on experiment set B ($N = 194$) but present the final results for all three sets. Each set consisted of five replications of the experiment with its unique conditions.

The participants (or players) were asked to estimate the "diet level" that maximizes the "fitness" of a virtual character in the "Fitness Game" designed by us. The true relationship between the diet level and fitness was a given deterministic and concave function $f(z_i)$, however, the players received a noisy value $\tilde{f}(z_i) \equiv f(z_i) + \nu$ of the fitness associated with the guessed diet level z_i. This noise ν, in reality, could be from other external factors such as environment and mood. The players were allowed multiple guesses, and were rewarded instantly based on the character's fitness level. The players also received monetary rewards based on their relative performances.

In each replication $p \in \{1, \ldots, 5\}$ of the "Fitness Game," the n_p participants first entered a session (control group) where they played the game in an *open loop* for 240 seconds (four minutes). In this session, each participant entered a series of guesses to best predict the unknown optimal diet level $\theta^* \in [2000, 2500]$ kcal. When a player entered a guess, the interface would refresh and the player would see the virtual character's fitness level (maximum 100%) for the guessed value. The player could then enter a new value until this session ended.

Subsequently, the same cohort entered the closed loop session (treatment group) where they played the same game with a population feedback. The game was reset and a new optimal diet level θ^* was chosen. In this session, in addition to the fitness level corresponding to their own guess z, players also received a soft feedback saying "We recommend $\frac{1}{n_p}\sum_{i=1}^{n_p} z_i$ kcal," where z_i denotes the most recent guess of the i-th player. This feedback only updated when the players took actions. The players had the option of using the feedback in any manner they desired.

In this treatment group, we revealed the population average of the diet level to each player. Thus, the choices of the players were *not* independent. However, we allowed the players the freedom to accept, reject, or partially accept such a population feedback, i.e., set the diet level to be a combination of their individual guesses and the feedback.

In the previous section (p. 61), we had introduced the possibility of partial acceptance of population recommendation in the context of regulating emerging industries. We also argued that soft regulation is appropriate and efficient (and desirable) when the observed outcomes are very noisy, individual decision makers are rational utility-maximizing agents, and the agents are exploiting abundant resources, and therefore, not competing. Medical research and health optimization using large-scale social interactions, for example via Apple's ResearchKit and CareKit [Apple, 2016], are examples of systems that satisfy these three conditions. The "Fitness Game" is meant to mimic these conditions.

Upon completion, participants received monetary rewards based on their relative game scores within the same cohort. We hoped to incentivize the participants in this way so that they would make rational decisions and actively optimize their virtual character's fitness, instead of making random guesses to get the participation rewards.

Detailed Description of the Experiment Environment We developed the "Fitness Game" using Google Apps Script (Appendix C, p. 117). All the data were stored in Google Sheets. Once the players accepted the task, they were first asked to carefully read the game instructions (Fig. 4.6, p. 68). The total task duration was ten minutes. The open loop (game level 1) and soft regulation (game level 2) sessions lasted precisely four minutes each. Players who wished to practice could enter the practice mode (game level 0) any time before open loop session began. After completing both open loop and soft regulation sessions, the players received a message about compensation information.

The interactive app (Fig. 4.7, p. 69) consists of the following components: The upper left panel shows the number of attempted guesses, the most recent guess, the fitness level, and the latest score. The panel changes from red to green whenever the player scores one point. In the soft regulation session, an additional message recommends the current *vox populi* population feedback (see Fig. 4.7, bottom). The upper right panel records latest game scores. The lower left scatter chart plots the ten most recent entries (fitness versus diet). The lower right line chart plots the fitness history of the ten most recent entries.

The virtual character's random fitness level $\tilde{f}(x)$ as a function of player's decision error $x = z - \theta^*$ was given by

$$\tilde{f}(x) \equiv y^* - \left(\frac{x}{\kappa}\right)^2 + \nu,$$

where $y^* = 98\%$ is the maximum achievable fitness, $\kappa = 500$ kcal is the scale of the fitness function, and ν is a sample from a random variable uniformly distributed over $[-2\%, 2\%]$. The player was awarded one score point whenever the guess led to a fitness level of 99% or higher.

4.2.3 Results and Discussion

Wisdom of Crowds Effect Let's begin with the analysis of the wisdom of crowds effect. We plot the time series of each individual player's decision error (x_i) as well as that of the wisdom of crowds (u) in Fig. 4.8 (p. 70). The performance of the wisdom of crowds is clearly superior: u steadily and quickly reaches the solution within the first minute while individual players lag behind.

Fig. 4.8 (p. 70) also confirms the behavior observed in the literature: The wisdom of crowds significantly outperforms the individual estimates, but such effect is weakened by social influence. The average in the soft regulation setting slightly lags behind that in the open loop.

Improvement from Soft Regulation Now, let's analyze how soft regulation improves the crowd's learning performance. By visually inspecting Fig. 4.8 (p. 70), we observe the narrowing of individual error distribution in the soft regulation setting: There are fewer ex-

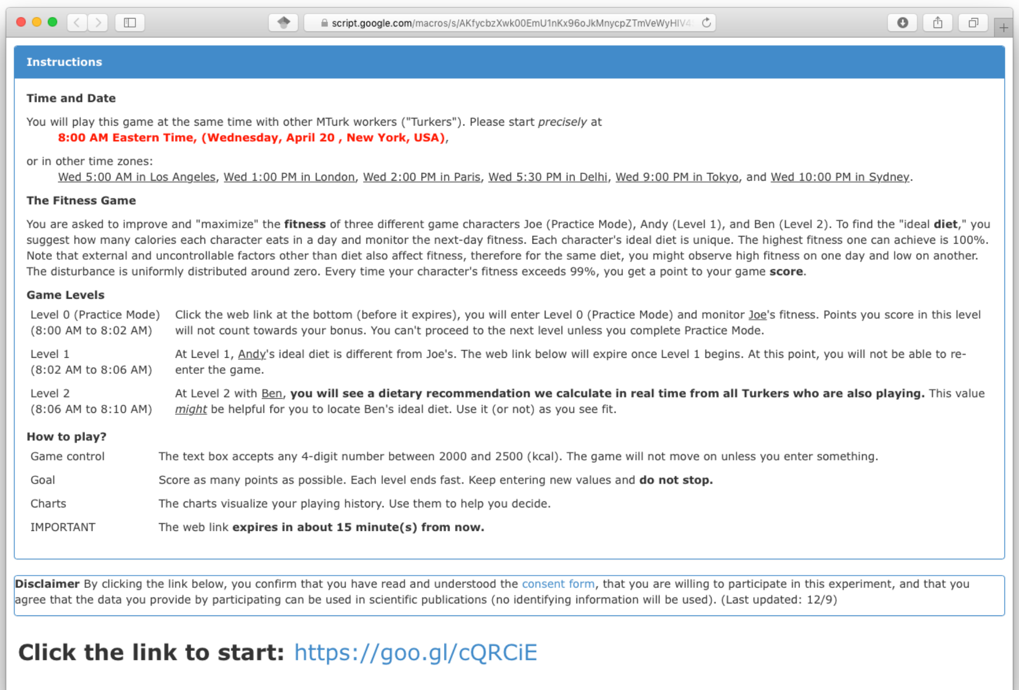


Figure 4.6: **Instructions for the "Fitness Game".**

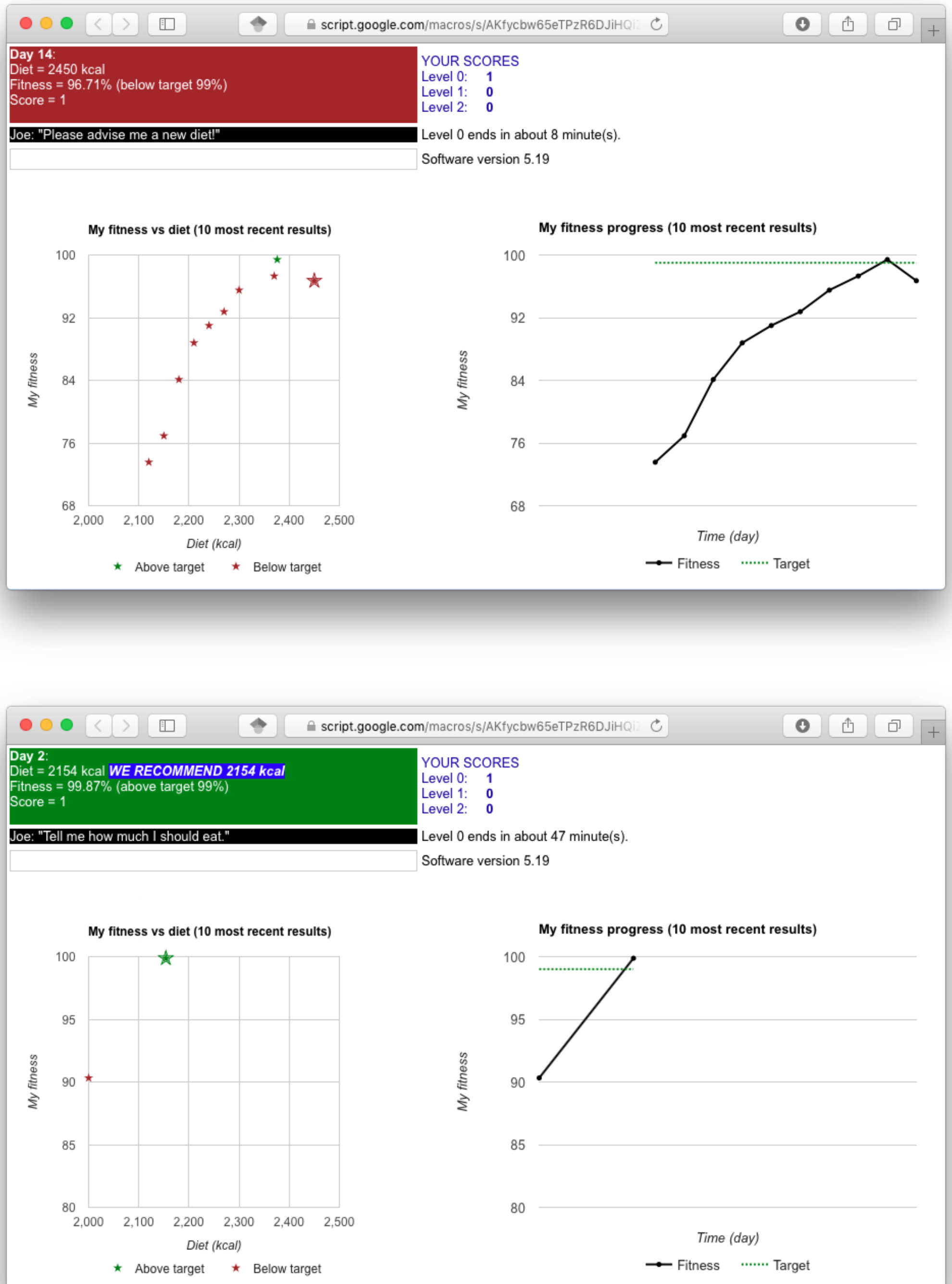


Figure 4.7: **Game interface.**

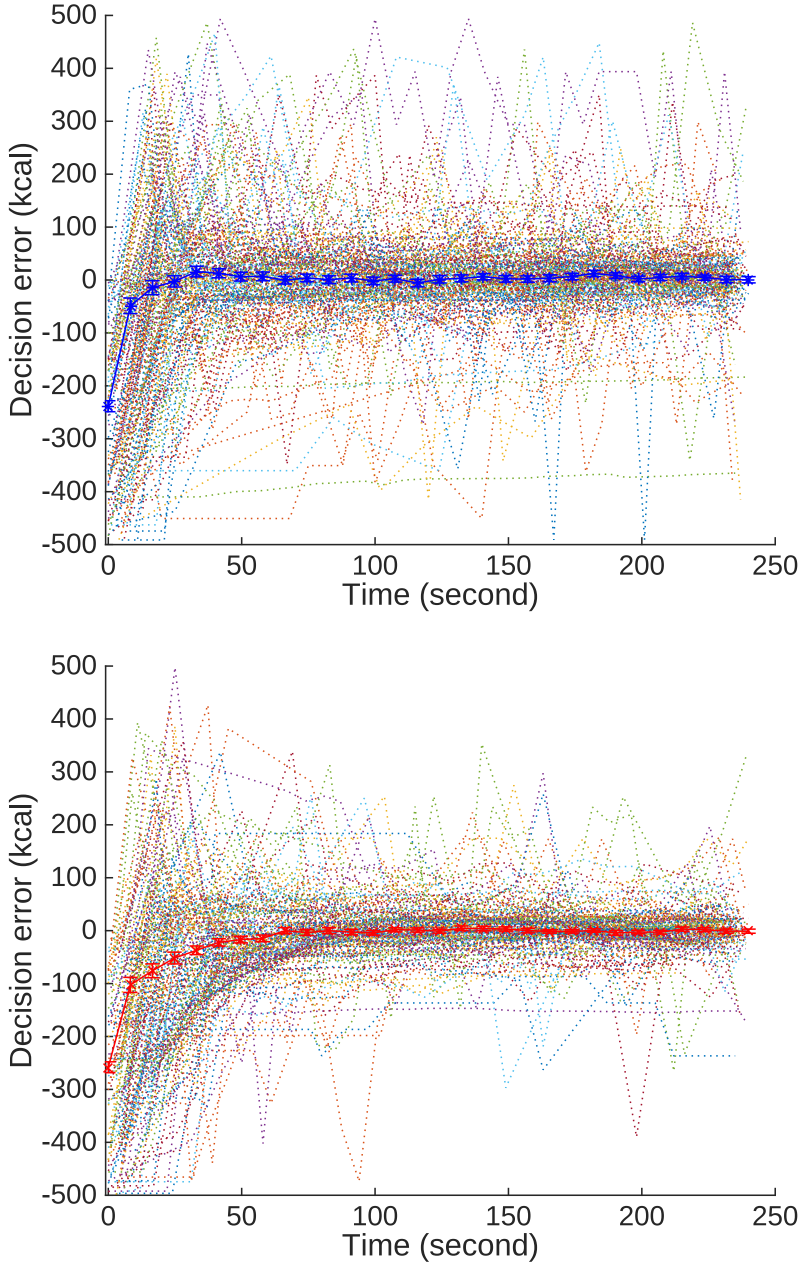


Figure 4.8: **Learning process of each individual player and the wisdom of crowds**. Top: open loop setting. Bottom: soft regulation setting. Each colored dashed line represents an individual participant's time series of decision error. The solid line is the arithmetic average of individual decision errors. Error bars reflect the standard errors of the mean.

treme errors than those in the open loop setting; most guesses are confined within ±100 kcal around optimum. In contrast, there are a significant number of players making completely off guesses (±500 kcal) in the open loop (even towards the end of sessions).

In Fig. 4.9 (p. 72) we plot the MSE time series to quantitatively assess the crowd's performance. The total MSE is approximately 30% lower in the soft regulation setting than in the open loop setting. Unlike the deterioration in the performance of wisdom of crowds, here social influence improves convergence and reduces the effect of noise. *The critical feature of soft regulation is that the players can ignore the feedback.* Since self-interested individuals reject feedbacks that appear unhelpful, the self-filtered social feedback significantly improves performance.

The observed improvement from soft regulation indicates that, without external interference, partially following the average opinion helped the players solve the "Fitness Game" problems. Next, we will characterize the system and estimate how much social influence was present in the experiment, and the optimal degree of social influence that would have optimized the crowd's performance.

System Identification We assume the learning function to be $g_i(x) \equiv g(x) = gx + \omega$ and the degree of social influence to be $\beta_i \equiv \beta$ (Assumption 3.13, p. 42). The estimate $\hat{g}(x) = 0.75x + \omega$ and $\hat{\sigma}_\omega = 60$ ($r^2 = 0.97$) was computed using the open loop results. From (3.31) on p. 44, we first estimated g and σ_ω by regressing MSE time series values $v(t+1)$ against $v(t)$. Using these estimates as an initial guess, we then ran a Monte Carlo (MC) simulation with 5000 samples and computed the expected MSE time series. By minimizing the mean squared difference between that with the open loop MSE time series, we obtained the g and σ_ω estimates (Appendix D, p. 138). The corresponding MSE evolution is plotted in Fig. 4.9 (p. 72).

The estimate $\hat{\beta} = 32\%$ ($r^2 = 0.99$) for the degree of social influence was computed using the treatment group results where the players received the population feedback. The corresponding MSE evolution is plotted in Fig. 4.9 (p. 72). Following the studies [Soll and Larrick, 2009; Moussaïd *et al.*, 2013] that have established that people rely more on themselves when the opinions of others are very dissimilar, we computed an "opinion dis-

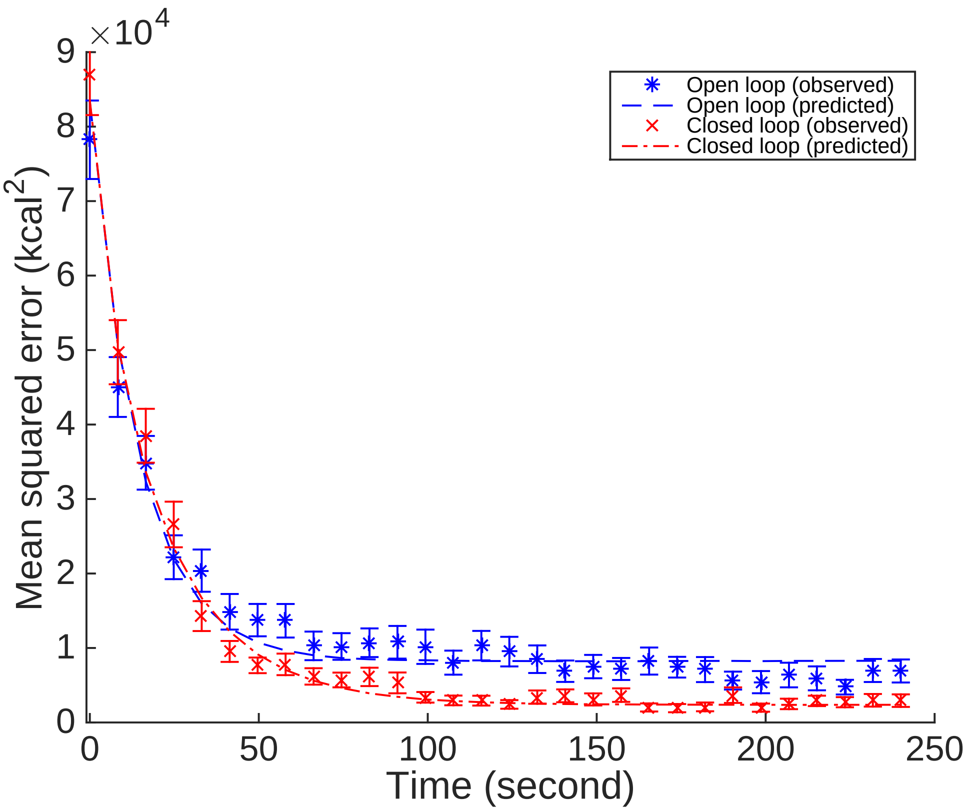


Figure 4.9: **MSE time series of human subject experiment results**. Blue dots (or red crosses) are the MSE values sampled at different points in time ($T = 30$) in the open loop (or soft regulation) setting. The dashed lines are simulation results based on models from system identification.

tance" function $\beta(d)$, where $d \equiv |g(x) - u|$ is the distance of an individual decision from the population feedback. We found it to be $\hat{\beta}(d) = \exp(-0.011d)$ ($r^2 = 0.98$).

Optimal Degree of Social Influence Given the estimates $\hat{g}(x)$ and $\hat{\sigma}_\omega$, one can compute the optimal degree of social influence β^* that, hypothetically, would optimize the closed loop soft regulation performance. The results are summarized in Table 4.2 (p. 75), and the associated MSE time series are displayed in Fig. 4.10 (p. 74). We first consider the case where the degree of social influence β is fixed. The empirical estimate $\hat{\beta}$ of social influence computed from experiment data is listed as a reference. The *robust* social influence β^*_{R} was calculated by minimizing the RHS in (3.34) (p. 45), i.e., optimizing the worst case cumulative expected MSE. The *MC* estimate β^*_{MC} was calculated by minimizing the cumulative MSE $V(T)$ in (3.25) (p. 44) with the expectation approximated by a Monte Carlo estimate. We regard β^*_{MC} as the true optimal degree of social influence. In Table 4.2 (p. 75), the column labeled ΔMSE lists the decrease of the cumulative expected MSE from the open loop to the soft regulation setting. The performances of the empirical estimate $\hat{\beta}$, the robust estimate β^*_{R}, and the optimal value β^*_{MC} are quite close. It is comforting to know that the social influence present in the experiment was close to the optimum.

We expect the degree of social influence, a function of the opinion distance or a function of time, to likely improve convergence. The $\hat{\beta}(d)$ profile estimated from experimental data results in $\Delta\mathrm{MSE} = 30\%$, which is not distinguishable from the performance of a constant β. However, the optimal β profile $\beta^*_{\mathrm{MC}}(d)$ with $\Delta\mathrm{MSE} = 47\%$ is significantly superior. The performance of the optimal dynamic robust social influence $\beta^*_{\mathrm{R}}(t)$ is also listed in Table 4.2 (p. 75). Since we do not have evidence to suggest the subjects used a dynamic value for β, and the performance of $\hat{\beta}(d)$ is close to $\hat{\beta}$, we assume that the subjects used the constant $\hat{\beta}$ for the rest of our results.

Discussion There is a fundamental difference between *vox populi* and the soft regulation mechanism proposed in this work. Even though both come under the umbrella of "collective intelligence," the *vox populi* aggregates the wisdom of *experts* while the latter harnesses the wisdom of *learners* (Section 1.2.1, p. 9). Experts base their opinions on prior knowledge. Such knowledge comes from experience and beliefs, which are unlikely to change.

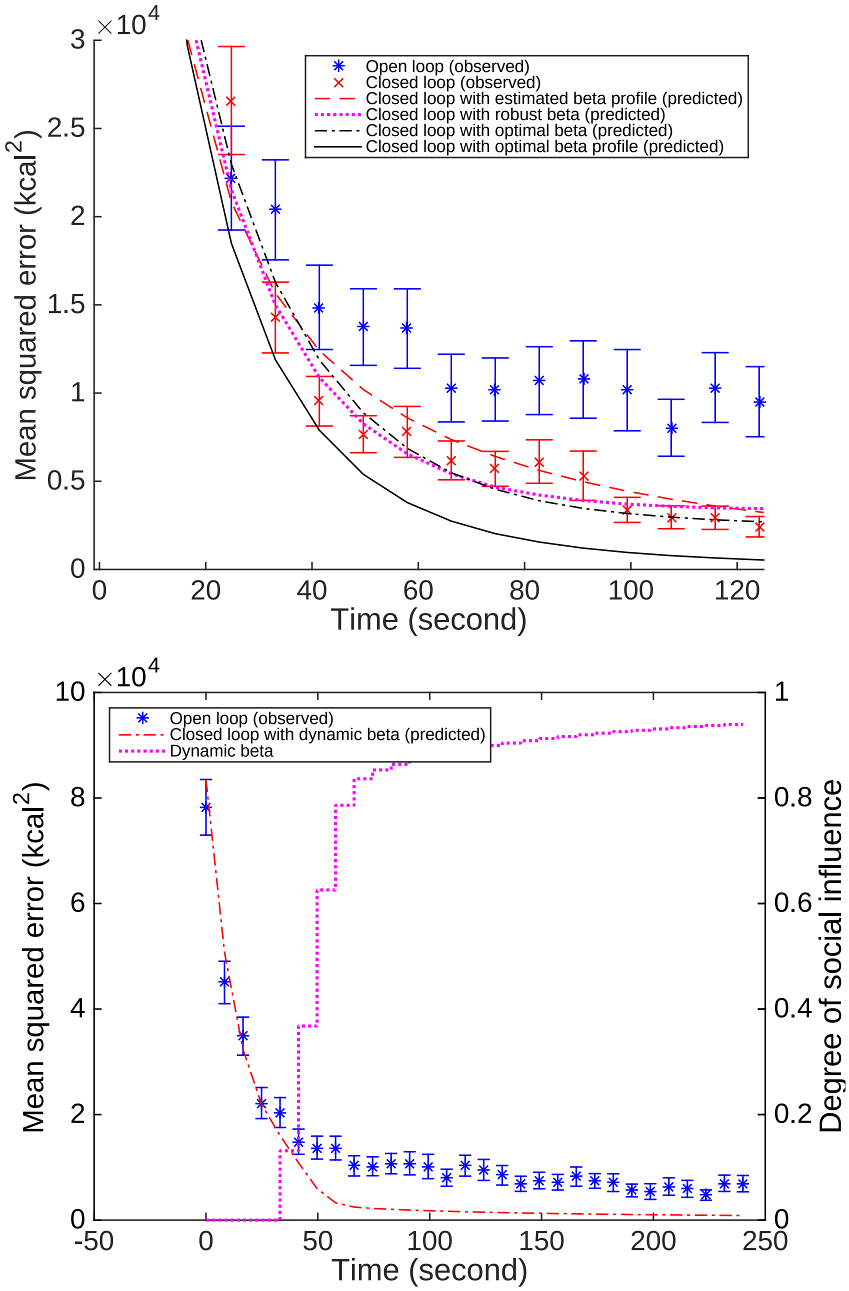


Figure 4.10: **Monte Carlo simulation of the expected MSE time series.**

Table 4.2: Optimal degree of social influence

Type	Value	ΔMSE
β (observed)	$\hat{\beta} = 32\%$	29%
β (robust)	$\beta^*_{\mathrm{R}} = 23\%$	27%
β (MC, true optimum)	$\beta^*_{\mathrm{MC}} = 30\%$	29%
β profile (observed)	$\hat{\beta}(d) = \exp(-0.011d)$	30%
β profile (MC, true optimum)	$\beta^*_{\mathrm{MC}}(d) = \exp(-0.026d)$	47%
Dynamic β (robust)	$\beta^*_{\mathrm{R}}(t)$	39%

Table 4.3: The "Fitness Game" experiment results

Description	Duration	Crowd size (n)	Horizon (T)	Learning gain ($\hat{g}$)	Noise ($\hat{\sigma}_\omega$)	r^2	Noise ratio	Optimal β	ΔMSE
The Fitness Game (Set B)	0-240s	39	30	0.75	60	0.97	5%	30%	29%
The Fitness Game (Set N)	0-240s	41	30	0.7	57	0.98	4%	32%	25%
The Fitness Game (Set S)	0-240s	9	30	0.65	51	0.98	3%	30%	17%

Independency and diversity of opinions prevent the "groupthink" behavior — undesirable convergence of individual estimates [Sunstein and Hastie, 2014]. In this setting, social influence, which violates independency, reduces the accuracy of the wisdom of crowds.

Learners, on the other hand, *revise* their decisions by interacting with the problem as well as other learners. Consider, for example, flocking birds. The birds have to adapt to changing weather; they gather local information, follow their closest neighbors, and revise directions constantly [Reynolds, 1987]. In this collective learning environment, individuals, like the flocking birds, are *both* respondents who generate new information, and surveyors who poll their social networks to improve decisions.

It appears that a social influence degree of 30% is robust across many different scenarios. In Table 4.3 (p. 75), the optimal degree of social influence ranges from 30% to 32% for the three sets of the "Fitness Game" experiment. Prior literature [Lim and O'Connor, 1995; Harvey and Fischer, 1997; Yaniv and Kleinberger, 2000; Yaniv, 2004; Soll and Larrick, 2009] also reports 30% to be the commonly observed degree of social influence on average. Whether this value is a mere coincidence requires further investigation.

The self-interested filtering of the feedback is key to ensuring the accuracy and efficiency of the soft regulation mechanism. Individuals will reject the feedbacks that appear useless. The experimentally observed magnitude of soft regulation is close to the theoretically predicted value for the optimal degree of social influence. This discovery suggests the promise of soft regulation for challenging real-world problems that require collective learning and action.

4.3 Local Regulation and Policy: U.S. State Tax and Expenditure Case Study

4.3.1 Background

There are 50 states in the U.S. ($n = 50$, or $n = 51$ if we consider the District of Columbia). Individual states constantly revise their tax and expenditure policies to maximize the overall wellbeing (economic growth, political stability, etc.). In this case study, the policy or strategy z_i for the i-th state here is the percentage of a particular tax (or expenditure) revenue from the total revenue (or spending). Such percentage reflects the relative importance of the tax/expenditure item. The main question we address here is whether one can accelerate the convergence thereby making the crowd of fifty states "smarter."

4.3.2 Results and Discussion

We apply the same control-theoretic analysis used in Section 4.2 (p. 65) to the state tax and expenditure case study. Here the optimal taxation or expenditure policy θ^* is unknown. To compute the MSE, and subsequently carrying out the system identification, we assume that the arithmetic mean of individual policies (i.e., the wisdom of crowds) has converged to the optimum in the last decade. This is a reasonable assumption that resonates with observations made by Galton and Surowiecki [Galton, 1907b; Surowiecki, 2005] as well as our observations from the experiment (Fig. 4.8, p. 70). Another critical assumption we are making here is that there exists a true optimal taxation/expenditure policy and it is identical to all the states.

The results are displayed in Table 4.4 (p. 77). We plot the time series of individual and

Table 4.4: U.S. optimal state tax and expenditure results

Description	Duration	Crowd size (n)	Horizon (T)	Learning gain ($\hat{g}$)	Noise ($\hat{\sigma}_\omega$)	r^2	Noise ratio	Optimal β	ΔMSE
Total Gen Sales Tax (T09)	1946-2014	50	69	0.96	4	0.89	3%	35%	73%
Total License Taxes (C118)	1946-2014	50	69	0.97	0.82	0.89	0.4%	14%	34%
Alcoholic Beverage Lic (T20)	1946-2014	50	69	0.93	0.04	0.99	0.09%	20%	34%
Individual Income Tax (T40)	1946-2014	50	69	0.98	2.9	0.86	1%	14%	32%
Educ-NEC-Dir Expend (E037)	1977-2013	51	37	0.96	0.097	0.85	1%	28%	54%
Emp Sec Adm-Direct Exp (E040)	1977-2013	51	37	0.93	0.037	0.99	0.6%	11%	14%
Total Highways-Dir Exp (E065)	1977-2013	51	37	0.93	0.76	0.89	3%	31%	53%
Liquor Stores-Tot Exp (E107)	1977-2013	51	37	0.95	0.17	0.95	1%	42%	67%

crowd decision errors as well as MSE in Fig. 4.11 (p. 78) to Fig. 4.18 (p. 85) for four tax cases and four expenditure cases. The learning gains (Definition 2.6, p. 32) of the states are all very close to 1, i.e., in a noiseless setting, the convergence is very slow. A possible explanation is that drastic change of tax and expenditure strategies is either prohibited or discouranged. A larger noise (see e.g., T09 and E065) or a smaller learning gain (see e.g., T20 and E65) calls for a larger optimal degree of social influence, which is consistent with the results presented in Fig. 3.1 (p. 48). The improvement from soft regulation ranges from 14% to 73%. Even a small improvement could make a significant difference in the nation's overall welfare.

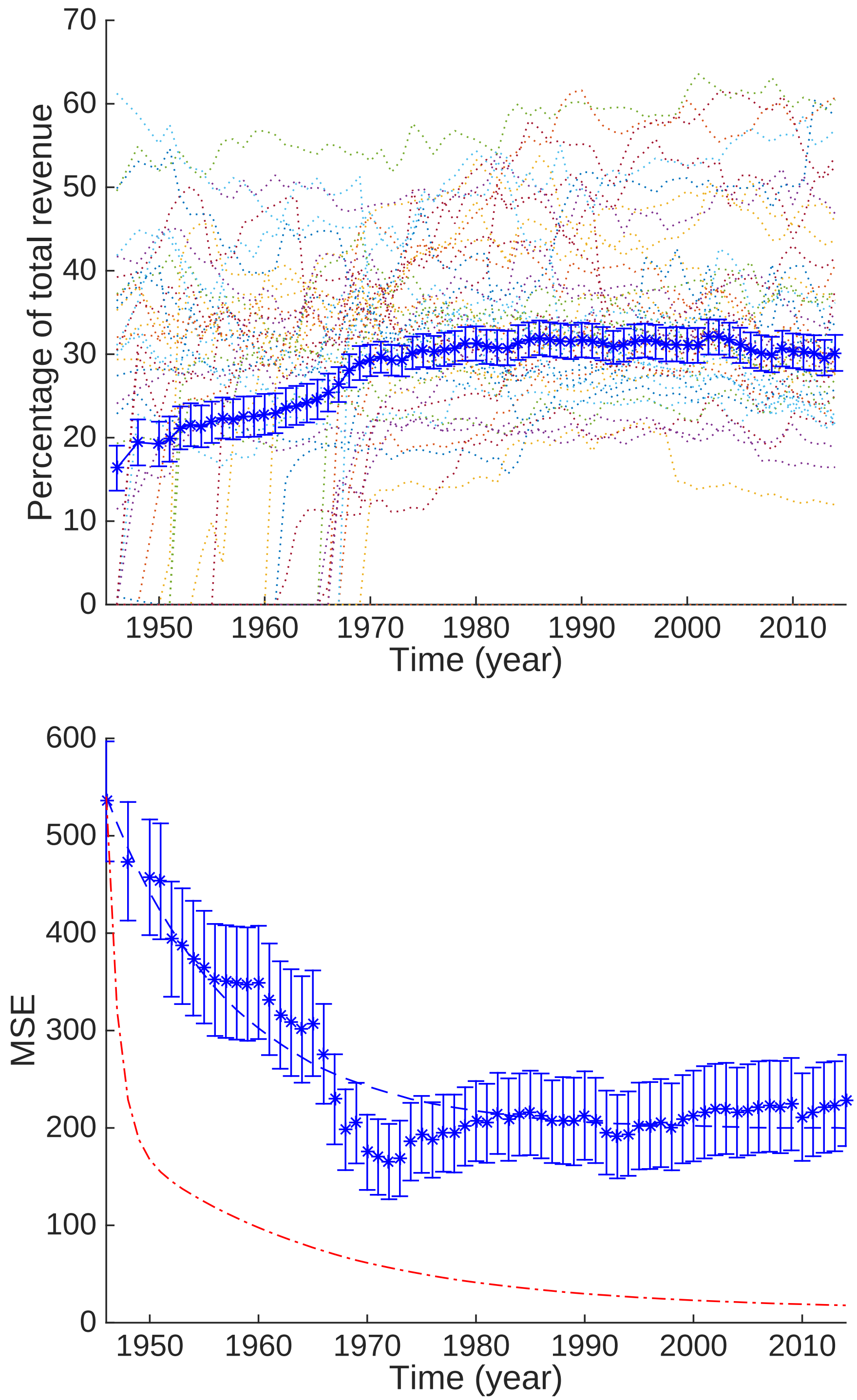


Figure 4.11: **Optimal taxation: total general sales taxes (T09)**. Top: time series of decision errors (individual and crowd). Bottom: time series of MSE (open loop in blue dashed line and closed loop in red dash-dot line). Fig. 4.12 (p. 79) to Fig. 4.18 (p. 85) follow the same legend style.

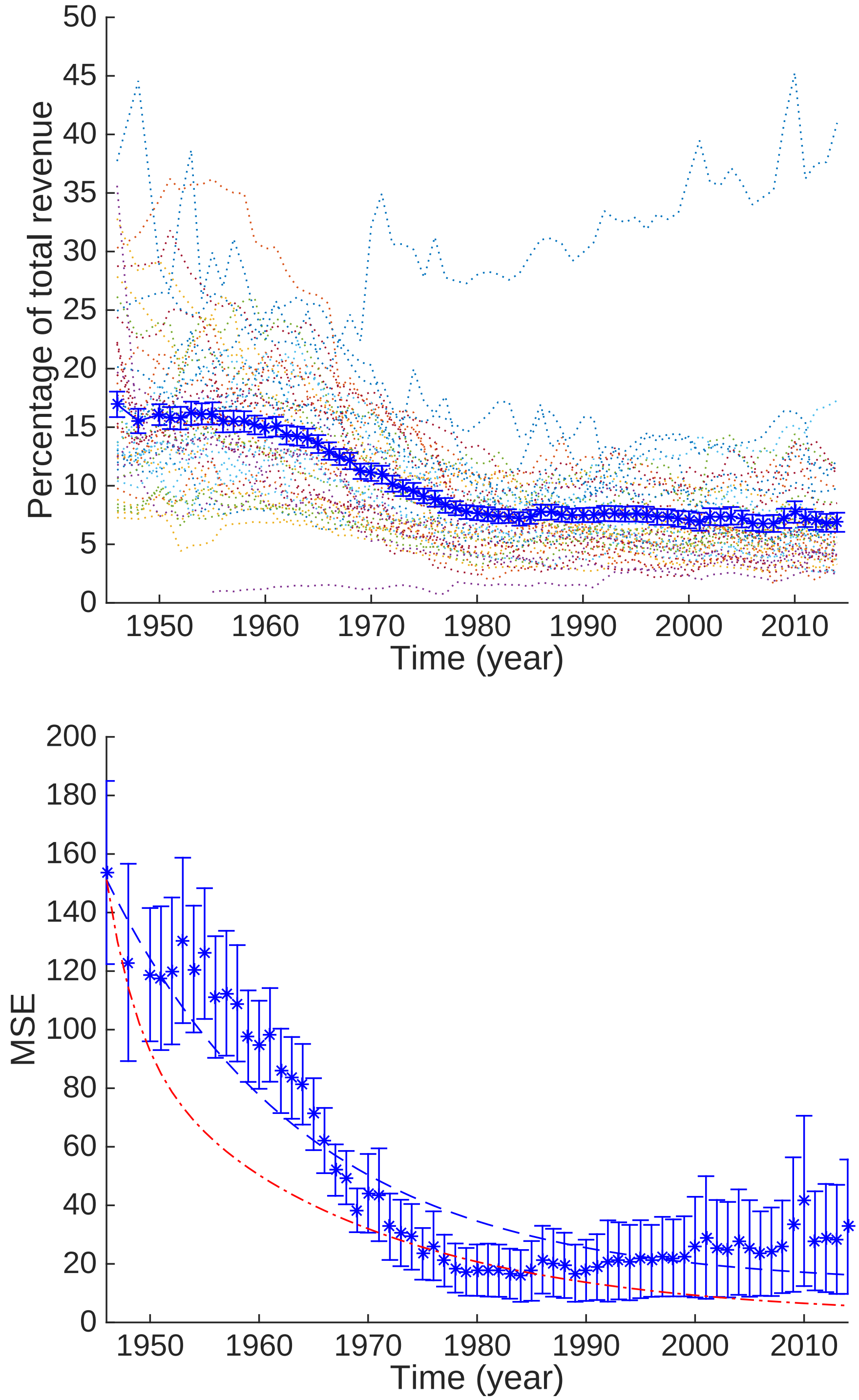


Figure 4.12: **Optimal taxation: total license taxes (C118).**

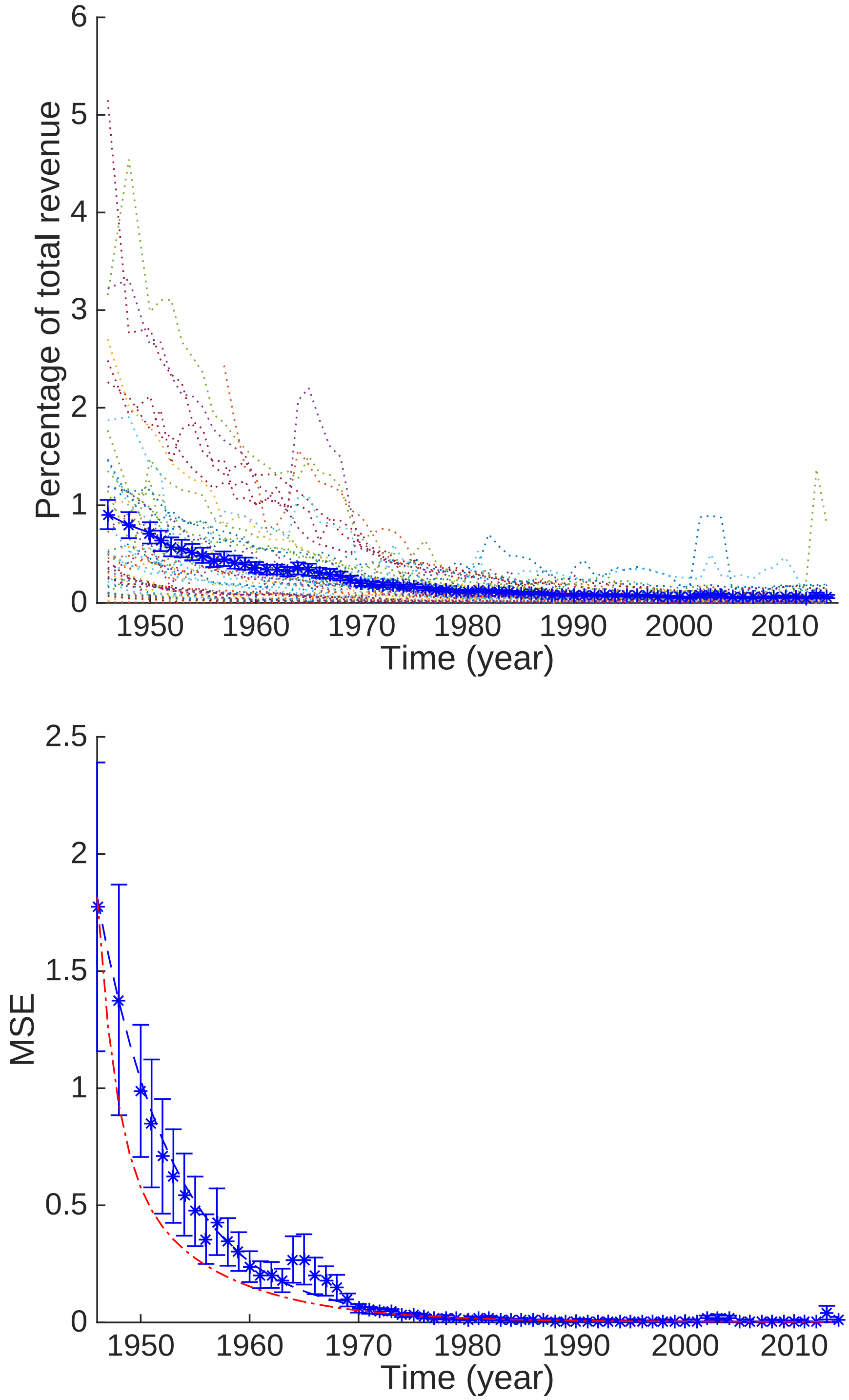


Figure 4.13: **Optimal taxation: alcoholic beverage license taxes (T20)**.

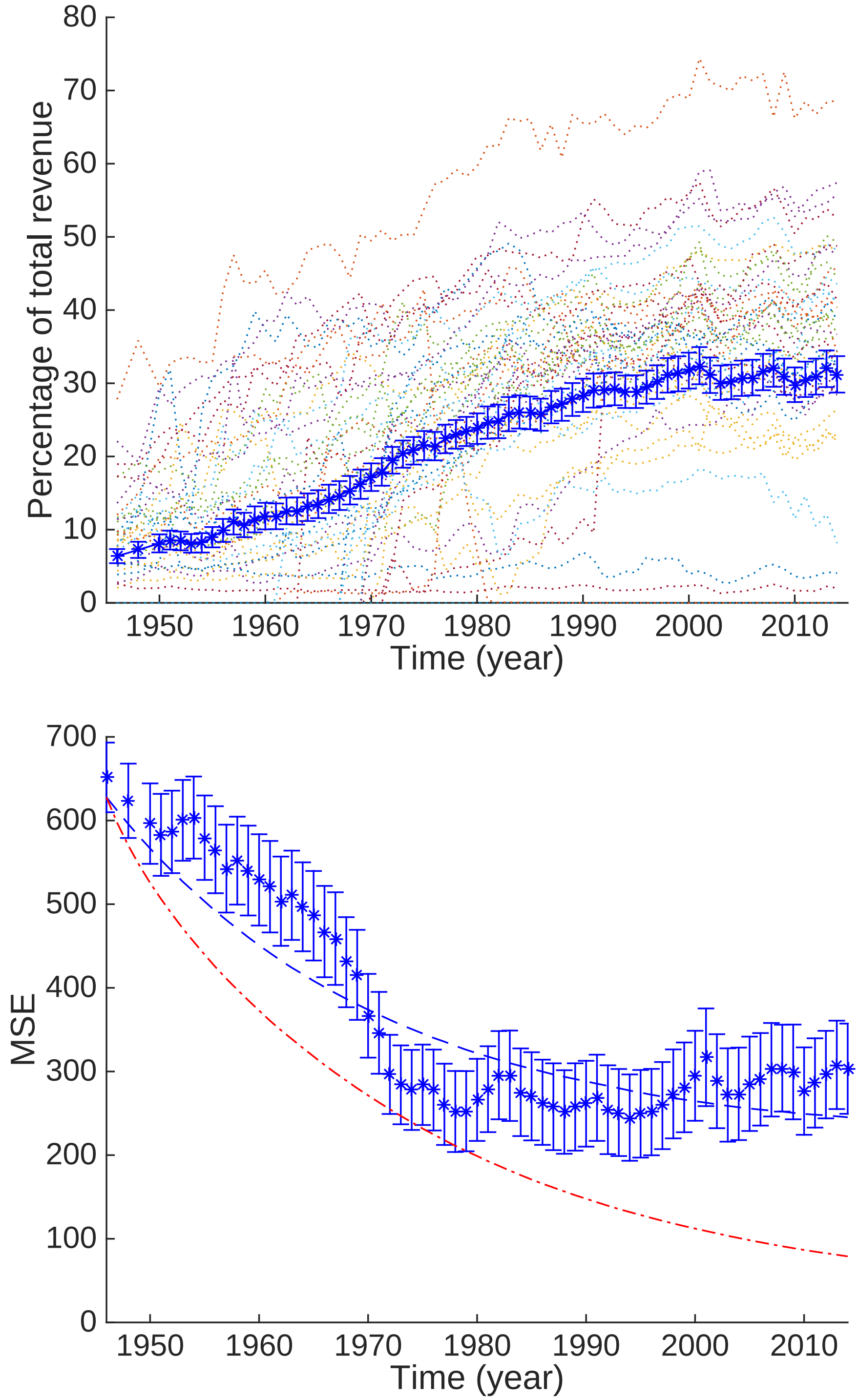


Figure 4.14: **Optimal taxation: individual income taxes (T40)**.

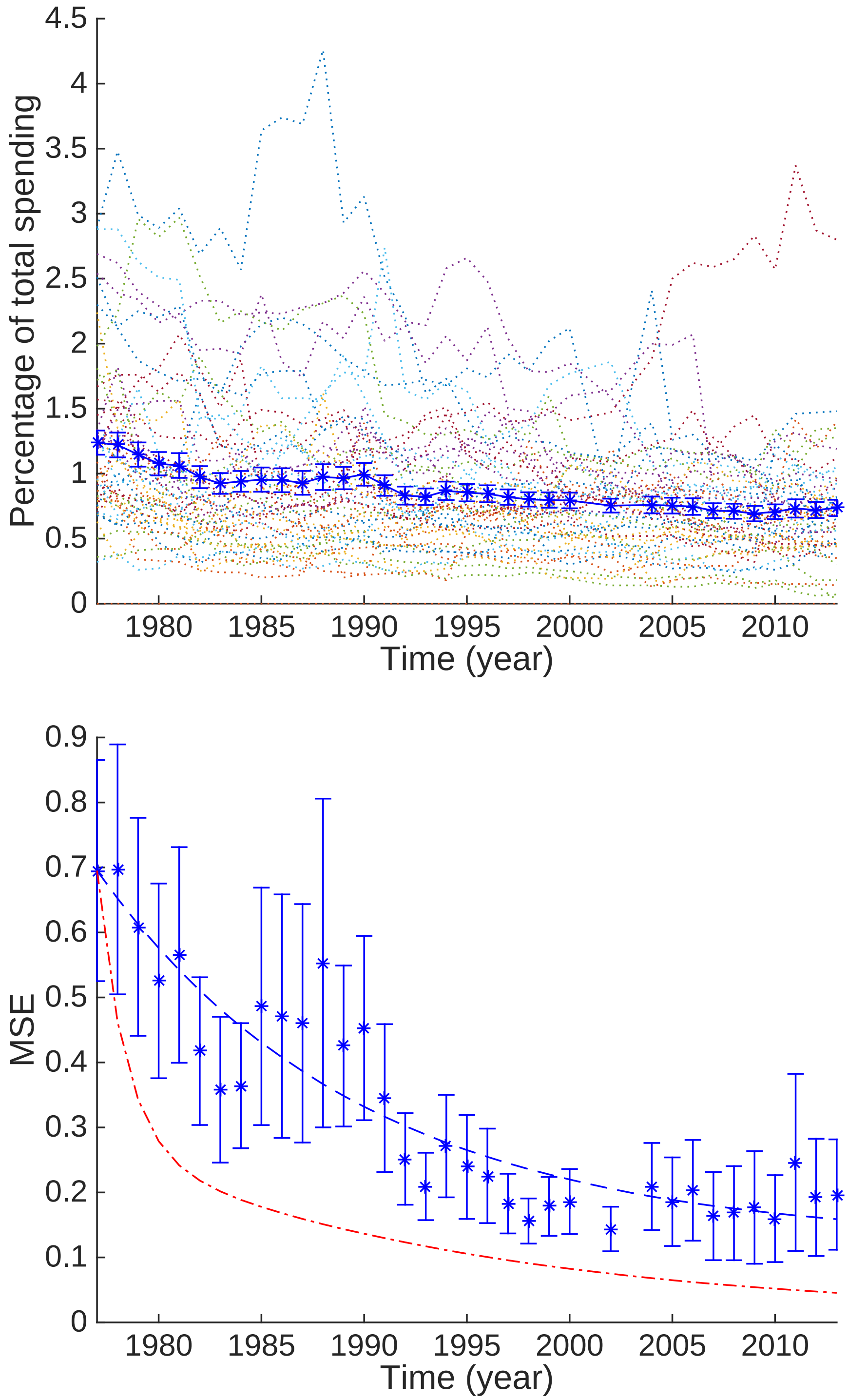


Figure 4.15: **Optimal spending: expense item "(E037) Educ-NEC-Direct Expend"**.

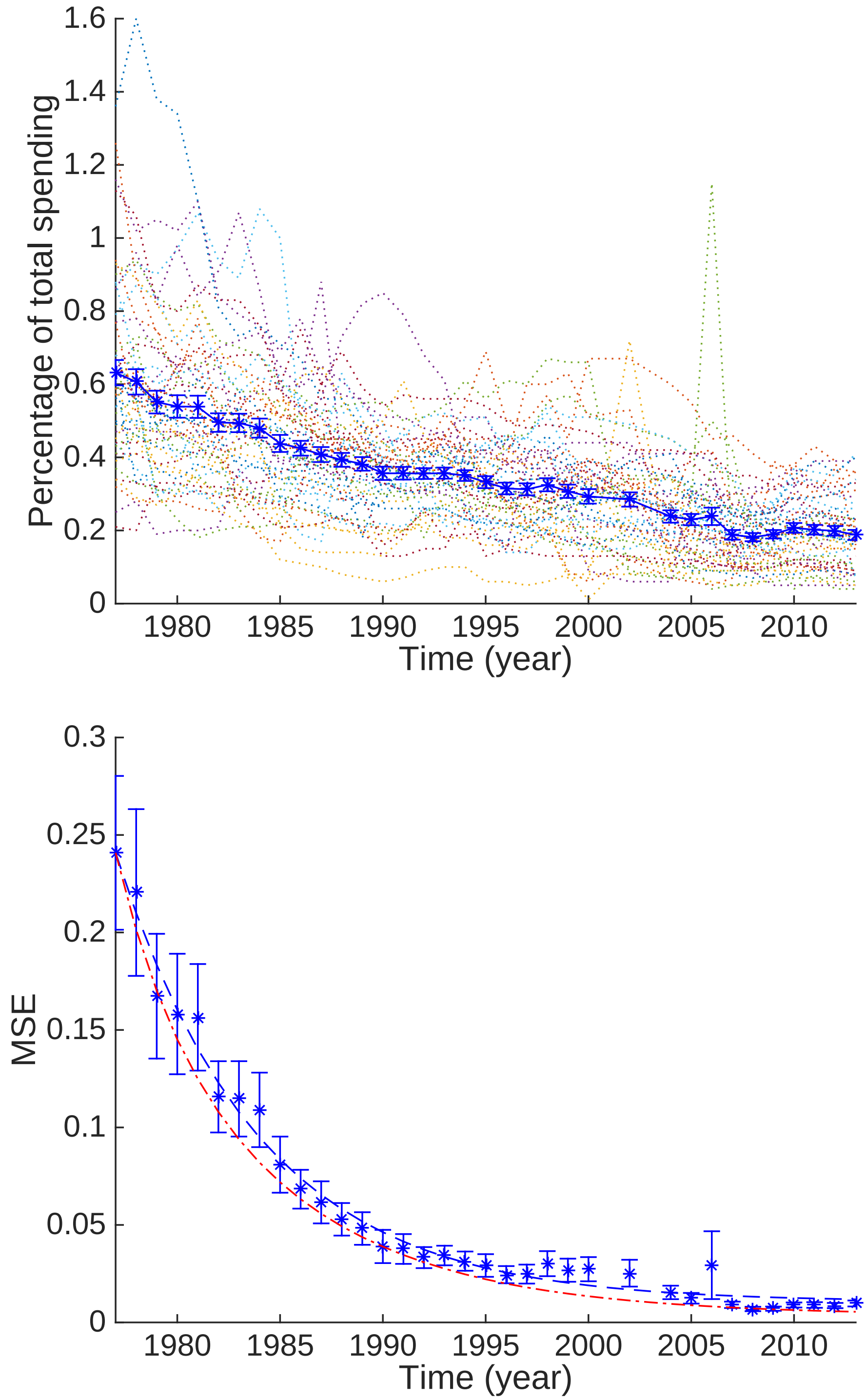


Figure 4.16: **Optimal spending: expense item "(E040) Emp Sec Adm-Direct Exp".**

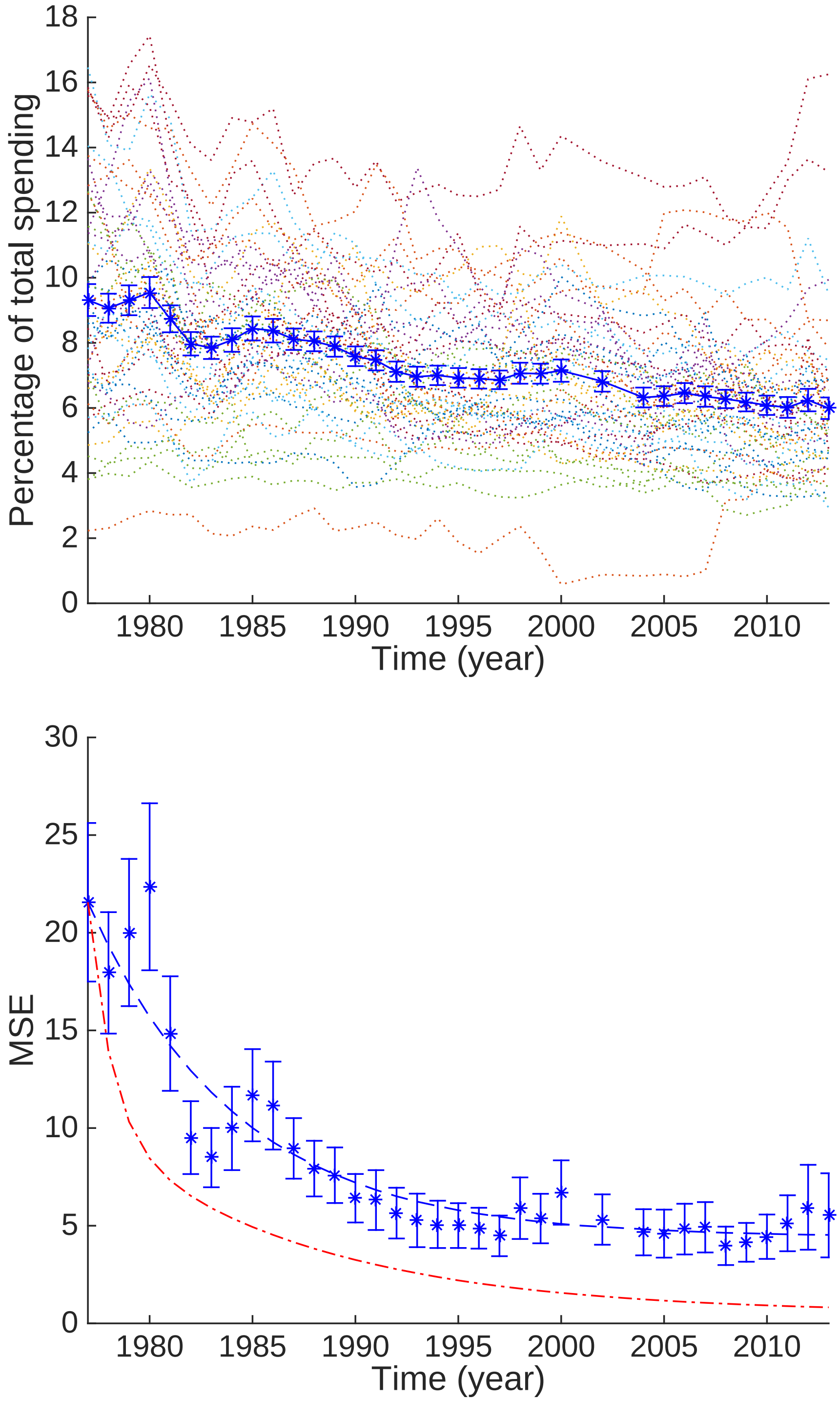


Figure 4.17: **Optimal spending: expense item "(E065) Total Highways-Dir Exp"**.

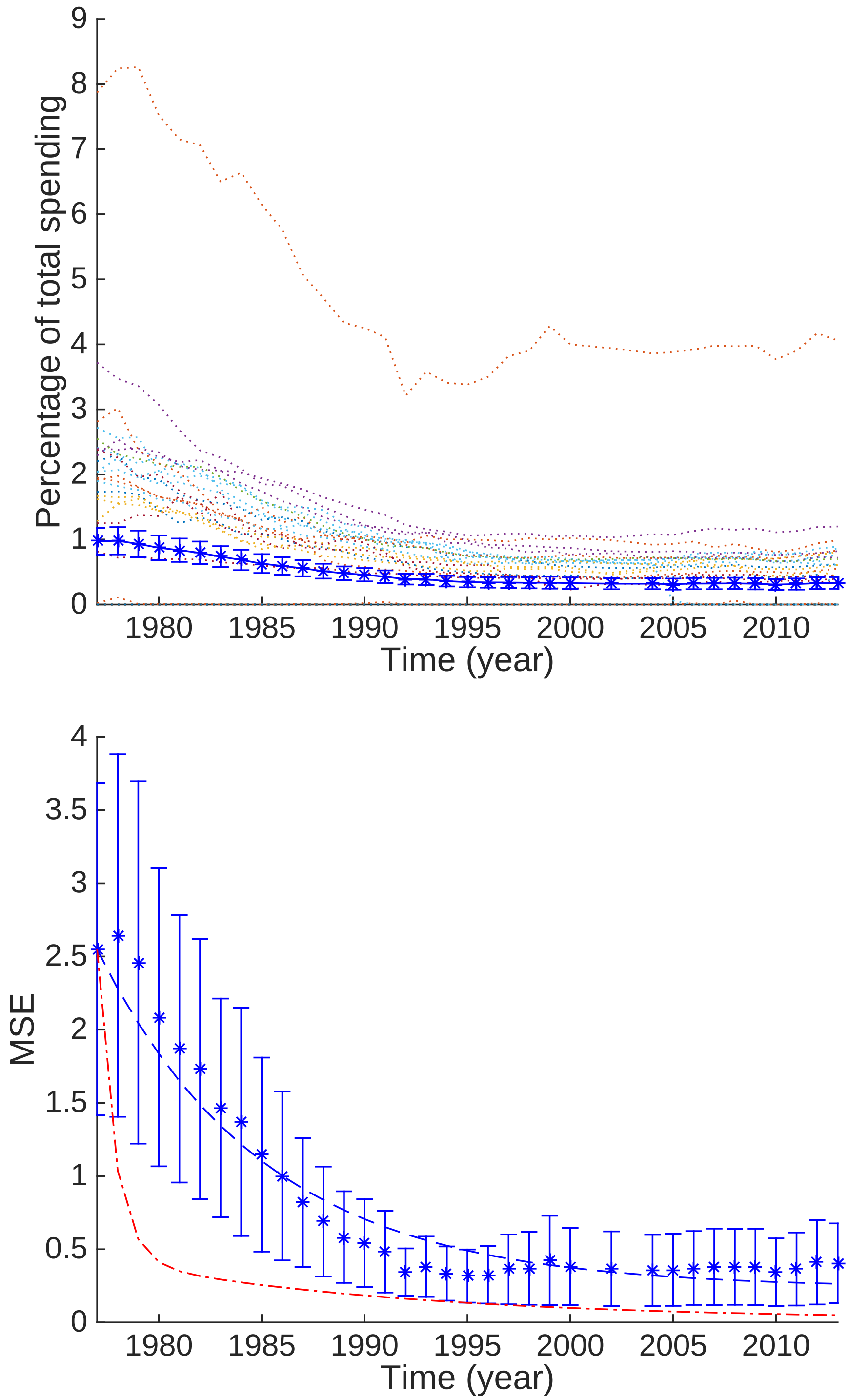


Figure 4.18: **Optimal spending: expense item "(E107) Liquor Stores-Tot Exp"**.

4.4 Multi-Product Revenue Management

4.4.1 Background

Consider the following multi-product pricing problem: A company like Amazon sells n *similar* products. For the i-th product, the (noisy) demand curve $\tilde{d}_i(p_i)$, i.e., how the market reacts to the price of the i-th product p_i, is unknown. The optimal price p_i^* that maximizes the expected revenue $\mathbb{E}\big[\tilde{r}_i(p_i)\big] = p_i\mathbb{E}\big[\tilde{d}_i\big]$ might differ from product to product (or from market to market for the same product). How can we coordinate the individual revenue management processes to maximize the cumulative revenue over a finite time horizon?

4.4.2 Model

We adopt the linear demand model and the pricing terminology from [Besbes and Zeevi, 2015].

Definition 4.2 (linear demand $\tilde{d}_i$ and its revenue $\tilde{r}_i$)**.** Linear demand is a commonly used model in economics:

$$\tilde{d}_i(p_i) \equiv d_i(p_i) + \nu_i \equiv a_i - c_i p_i + \nu_i, \tag{4.20}$$

where the demand parameters a_i and $c_i \geq 0$ differ among products. Note that the actual demand is subject to disturbances $\nu_i \in \mathcal{N}(0, \sigma_\nu^2)$. The optimal price that maximizes the expected revenue is $p_i^* = a_i/2c_i$. We rewrite the expected revenue in terms of the decision error $x_i(t) = p_i(t) - p_i^*$ as follows:

$$\begin{aligned}
\mathbb{E}[\tilde{r}_i(t)] &= p_i(t) d_i\big(p_i(t)\big), \\
&= p_i(t)\big(a_i - c_i p_i(t)\big), \\
&= \big(x_i(t) + p_i^*\big)\Big[a_i - c_i\big(x_i(t) + p_i^*\big)\Big], \\
&= \left(x_i(t) + \frac{a_i}{2c_i}\right)\left[a_i - c_i\left(x_i(t) + \frac{a_i}{2c_i}\right)\right], \\
&= \frac{a_i^2}{4c_i} - c_i x_i(t)^2.
\end{aligned} \tag{4.21}$$

Definition 4.3 (regret v_i)**.** In revenue management, it is more common to use the regret as a metric of performance. Regret is the difference between the current revenue and the maximum expected revenue based on the optimal price:

$$v_i(t) \equiv \max_p \mathbb{E}[\tilde{r}_i(t;p)] - \mathbb{E}[\tilde{r}_i\big(t;p_i(t)\big)] = c_i x_i(t)^2. \tag{4.22}$$

It represents the gain in revenue if the i-th product is priced optimally.

The average regret of the n products is a weighted MSE:

$$v(t) \equiv \frac{1}{n}\sum_i v_i(t) = \frac{1}{n}\sum_i c_i x_i(t)^2. \tag{4.23}$$

If $c_i \approx c$ values do not differ significantly from each other, we can approximate $v(t)$ directly as a function of MSE, i.e., $v(t) \approx c\,\mathrm{MSE}(t)$. The average revenue is a function of the average regret:

$$\begin{aligned} y(t) &\equiv \frac{1}{n}\sum_i \tilde{r}_i(t), \\ &= y^* - v(t) + \frac{1}{n}\sqrt{\sum_i p_i(t)^2}\cdot \nu(t), \\ &\approx y^* - v(t), \end{aligned} \tag{4.24}$$

where $y^* \equiv \frac{1}{n}\sum_i a_i^2/4c_i$ is the maximum expected average revenue and random variable $\nu(t)$ is identical to the previously defined $\nu_i(t)$. As $n \gg 1$, the term containing $\nu(t)$ can be safely omitted. Similarly, the time average of cumulative revenue per product can be approximated as

$$\begin{aligned} Y(T) &\equiv \frac{1}{nT}\sum_{t=0}^{T-1}\sum_{i=1}^{n} \tilde{r}_i(t), \\ &\approx y^* - V(T), \end{aligned} \tag{4.25}$$

where $V(T) \equiv \frac{1}{T}\sum_{t=0}^{T-1} v(t)$ is the time average of cumulative regret per product. Note that one can only sample $y(t)$ and $Y(T)$ but not $v(t)$ or $V(T)$ at each time point $t = 0, 1, 2, \ldots$ or for each length of time $T = 1, 2, 3, \ldots$.

We first assume identical and linear learning function, i.e., $g(x) \equiv gx + \omega$, and that c_i values are close, i.e., $v(t) \approx c\,\mathrm{MSE}(t)$. We approximate the regret dynamics to be

$$v(t+1) \approx m^2 v(t) + (1-\beta)^2 c\sigma_\omega^2, \tag{4.26}$$

where $m \equiv (1-\beta)g + \beta$. Thus, we have the following equations:

$$v(t) \approx m^{2t} v(0) + \frac{(1-m^{2t})(1-\beta)^2}{1-m^2} c\sigma_\omega^2, \tag{4.27}$$

$$y(t) = y^* - v(t), \tag{4.28}$$

$$V(T) \approx \frac{1}{T}\frac{1-m^{2T}}{1-m^2} v(0) + \left[\frac{1}{1-m^2} - \frac{1}{T}\frac{1-m^{2T}}{(1-m^2)^2}\right](1-\beta)^2 c\sigma_\omega^2, \tag{4.29}$$

$$Y(T) = y^* - V(T). \tag{4.30}$$

Especially in the open loop setting where $\beta = 0$, one can shorten (4.29) and (4.30) above as

$$Y(T) \approx y_\infty - \frac{1-g^{2T}}{T}\Delta, \tag{4.31}$$

where $y_\infty \equiv y^* - v_\infty$, $\Delta \equiv \big(v(0) - v_\infty\big)/(1-g^2)$, and $v_\infty \equiv c\sigma_\omega^2/(1-g^2)$. By fitting $Y(T)$ data observed for different time horizon T values, one can solve uniquely y_∞, Δ, and g. Unfortunately, $v(0)$ and v_∞ are not linearly independent in those equations and cannot be uniquely recovered from this method. In the next section, we will approach this identification problem from analyzing price data.

4.4.3 Multi-Product Revenue Optimization Algorithm

Our algorithm is inspired by the Besbes-Zeevi (BZ) dynamic pricing model [Besbes and Zeevi, 2015] and the James-Stein (JS) estimator [Stein, 1956; James and Stein, 1961; Efron and Morris, 1977]. In the context of soft regulation, the BZ model represents individual open loop dynamics and the JS estimator introduces soft feedback.

The BZ model is straightforward (Algorithm 1, p. 89): For the i-th product, the algorithm first starts with an initial guess $p_i(0)$, then sets some new prices, observes the demand, estimates the optimal pricing, and repeats. Before the algorithm advances to the t-th stage[1], the price has changed $2t-2$ times.At the t-th stage, the product manager now sets two new prices. One is the current estimate of the optimal price $p_i(2t-1) = \hat{p}_i^*$ (if this is the beginning, the initial guess is used). The other is a perturbation to it $p_i(2t) = \hat{p}_i^* + \delta(t)$.

[1] With an abuse of notation, we use t to indicate the iteration number (or the t-th stage), instead of the number of times when the price is set; similarly, T here is the total number of stages. The 0-th stage marks the initial guess.

Algorithm 1 Besbes-Zeevi Dynamic Pricing Model

$\hat{p}_i^* \leftarrow p_i(0)$

for $t = 1, \ldots, T$ **do**

  for $i = 1, \ldots, n$ **do**

    $\hat{p}_i^* \leftarrow \mathrm{BZ}(\hat{p}_i^*, t)$

  end for

end for

function $\mathrm{BZ}(\hat{p}_i^*, t)$

  $p_i(2t-1) = \hat{p}_i^*$

  $p_i(2t) = \hat{p}_i^* + \delta(t)$

  $(\hat{a}_i, \hat{c}_i) \leftarrow \arg\min_{a,c} \sum_{\tau=1}^{2t} \left[\tilde{d}_i(\tau) - \big(a - cp_i(\tau)\big)\right]^2$

  return $\hat{p}_i^* \leftarrow \min\left(\overline{p}, \max(\underline{p}, \frac{\hat{a}_i}{2\hat{c}_i})\right)$

end function

By linearly fitting all observed demands including the ones with newly set prices, one can estimate the linear demand curve, thereby updating the estimate of the optimal price $\hat{p}_i^*$ (capped by the lower bound $\underline{p}$ and upper bound $\overline{p}$). The algorithm repeats for T stages in total. The perturbation $\delta(t)$ shrinks with time with a step size coefficient ρ. For illustration purposes, we follow the numerical example from [Besbes and Zeevi, 2015] and set $\delta(t) = \rho t^{-1/4}$.

The JS estimator describes a phenomenon that when estimating a set of random variables, one can improve the overall estimate by shrinking the individual estimators towards their grand average [Efron and Morris, 1977]. For instance, the least mean square estimator of the expectation of a single random variable is its sample mean. If the number of random variables to be estimated is larger than two and performance of an estimator is measured by its MSE, then the JS estimator would perform no worse than the sample mean. This is an interesting result because the expectations need not be the same. They do have to be close in order for the JS estimator to significantly outperform sample mean. The resemblance between the JS estimator and soft regulation makes us wonder if we can relax the constraint of identical solution (Definition 1.1, p. 6). Based on the result of Section 3.5 (p. 49), we propose Algorithm 2 (p. 90) to improve the performance of Algorithm 1 (p. 89) for multi-product revenue management.

Algorithm 2 Multi-Product Revenue Management with Soft Regulation

for $t = 1, \dots, T$ **do**
  $u \leftarrow \frac{1}{n} \sum_i \hat{p}_i^*$
  for $i = 1, \dots, n$ **do**
    $\hat{p}_i^* \leftarrow \mathrm{BZ}(\hat{p}_i^*, t)$
  end for
  if $t \geq 3$ **then**
    for $i = 1, \dots, n$ **do**
      $(\hat{k}_i, \hat{s}_i) \leftarrow \arg\min_{k,s} \sum_{\tau=1}^{t-1} \left[p_i(2\tau+1) - \left(k p_i(2\tau-1) + s \right) \right]^2$
      $\hat{g}_i \leftarrow \min\left(\overline{g}, \max(\underline{g}, \hat{k}_i)\right)$
      $\hat{\theta}_i^* \leftarrow \min\left(\overline{p}, \max(\underline{p}, \frac{\hat{s}_i}{(1-\hat{k}_i)})\right)$
      $(\hat{k}_i, \hat{s}_i) \leftarrow \left(\hat{g}_i, (1-\hat{g}_i)\hat{\theta}_i^*\right)$
      $\hat{\sigma}_{\omega_i}^2 \leftarrow \frac{1}{t-1} \sum_{\tau=1}^{t-1} \left[p_i(2\tau+1) - \left(\hat{k}_i p_i(2\tau-1) + \hat{s}_i \right) \right]^2$
    end for
    $\hat{\sigma}_\omega^2 \leftarrow \frac{1}{n} \sum_i \hat{\sigma}_{\omega_i}^2$
    $\hat{\sigma}_\theta^2 \leftarrow \frac{1}{n} \sum_i \left(\hat{\theta}_i^* - \frac{1}{n} \sum_j \hat{\theta}_j^* \right)^2$
    $\sigma_1(\beta) \leftarrow \left\| (1-\beta)\,\mathrm{diag}\left(\hat{g}_1, \dots, \hat{g}_n\right) + \beta \frac{1}{n} \mathbf{1}\mathbf{1}^\top \right\|_2$
    $\hat{\beta}^* \leftarrow \arg\min_{0 \leq \beta < 1} \frac{(1-\beta)\hat{\sigma}_\omega + \beta \hat{\sigma}_\theta}{1 - \sigma_1(\beta)}$
    for $i = 1, \dots, n$ **do**
      $\hat{p}_i^* \leftarrow (1-\hat{\beta}^*)\hat{p}_i^* + \hat{\beta}^* u$
    end for
  end if
end for

4.4.4 Results and Discussion

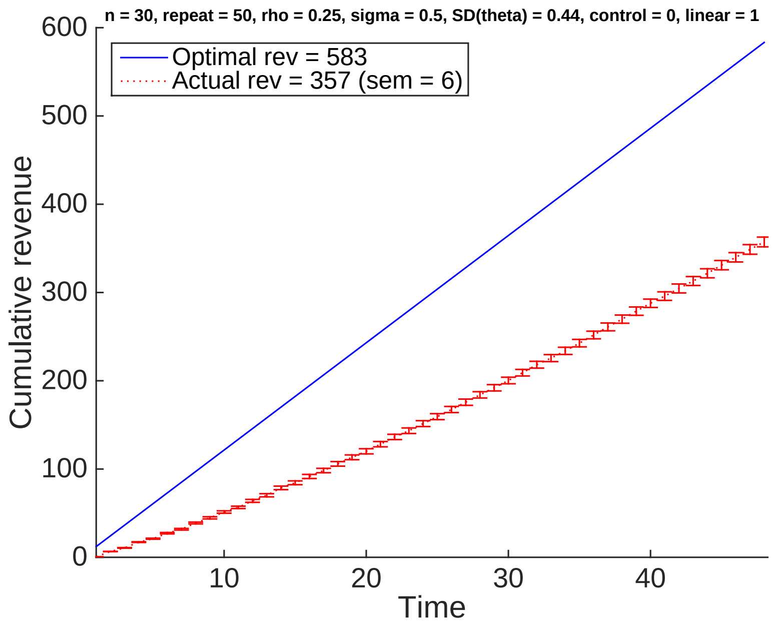
We include the numerical simulation results in Table 4.5 (p. 91). The simulation considers the following three factors that affect the performance of Algorithm 2 (p. 90): number of stages T, demand noise σ_ν, and step size ρ. The corresponding revenues are averaged over a set of 50 repetitions ($\pm$ reflects the standard error of the mean).

The demand curve used in this simulation follows [Besbes and Zeevi, 2015], i.e., $\tilde{d}_i(p_i) \equiv a_i - c_i p_i + \nu_i$ where $a_i \sim \mathcal{U}(0.8, 1)$, $c_i \sim \mathcal{U}(0.2, 1)$, and $\nu_i \sim \mathcal{N}(0, \sigma_\nu^2)$.

We measure the improvement with respect to the regret (Definition 4.3, p. 86) difference. For instance, a regret difference $\Delta V(T) = 20.2\%$ indicates that Algorithm 2 (p. 90) reduces the regret from Algorithm 1 (p. 89) by 20.2%. From this simple exercise, we find a significant improvement in performance by partially setting prices towards the grand average. We plot the revenue simulation and MSE time series for both algorithms in Fig. 4.19 (p. 92) and Fig. 4.20 (p. 93).

Table 4.5: Multi-product revenue management simulation result

			Revenue			
		ρ	Maximum	Algorithm 1	Algorithm 2	$\Delta V(T)$
$T = 12$	$\sigma_\nu = 0.25$	0.25	292	188 ± 3	209 ± 3	20.2%
		0.5		206 ± 3	209 ± 3	3.49%
	$\sigma_\nu = 0.5$	0.25		152 ± 3	182 ± 3	21.4%
		0.5		175 ± 3	194 ± 3	16.2%
$T = 24$	$\sigma_\nu = 0.25$	0.25	583	440 ± 7	472 ± 7	22.4%
		0.5		454 ± 7	455 ± 7	0.775%
	$\sigma_\nu = 0.5$	0.25		357 ± 6	418 ± 6	27.0%
		0.5		398 ± 7	428 ± 7	16.2%



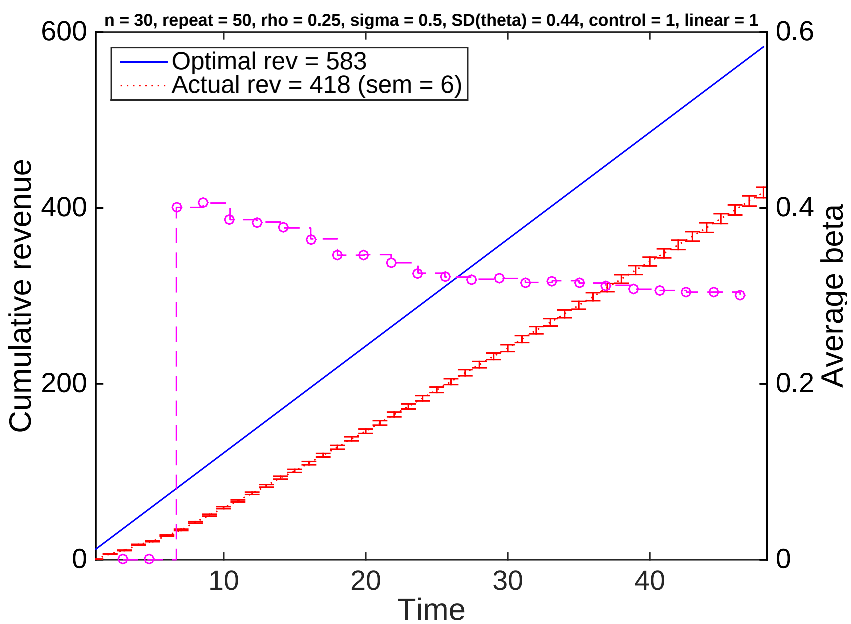


Figure 4.19: **Multi-product total revenue time series**. Top: Algorithm 1 (p. 89). Bottom: Algorithm 2 (p. 90)

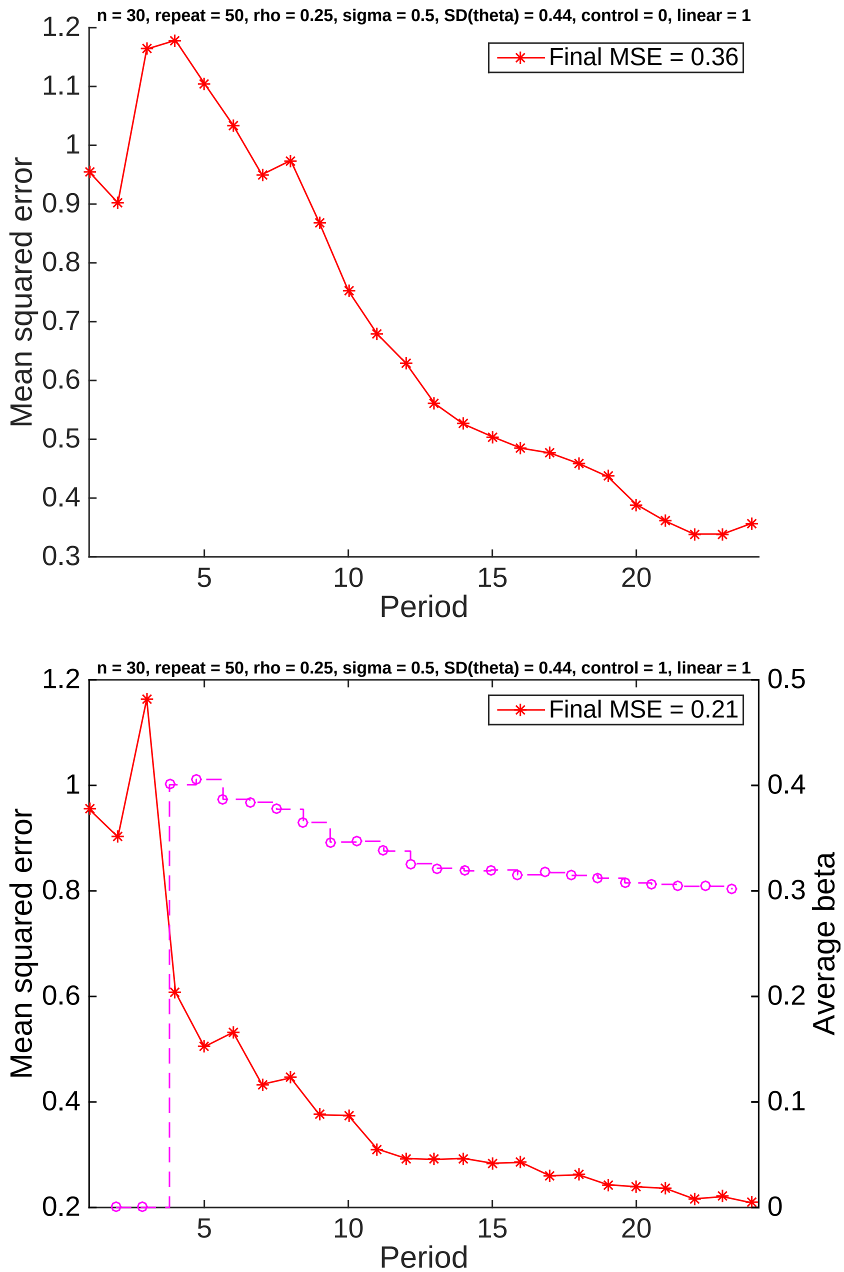


Figure 4.20: **Multi-product MSE time series**. Top: Algorithm 1 (p. 89). Bottom: Algorithm 2 (p. 90)

Chapter 5

Conclusion

This cross-disciplinary work addresses a simple yet important question: How could a group collectively become better problem-solvers? Our analysis suggests that intelligent individuals, solving the same problem (or similar problems), could do much better by adaptively adjusting their decisions towards the population average. Both theory, simulations, and an experiment with human subjects confirm the validity of our coordination mechanism. This work conveys the following important messages.

First, we lay down both the theoretical and empirical grounds for improving collective problem-solving processes. We illustrate the potential far-reaching impact of this work on case studies involving regulation of emerging industry and business, health optimization, optimal levels of U.S. state taxation and expenditure, and multi-product revenue management. Even a small improvement, magnified by the scale of affected population or volume, could have significantly promoted the overall welfare of a system.

Second, we quantitatively justify the unique and critical roles of social influence in collective intelligence. Scholars have long regarded social influence as detrimental to the accuracy of polls and other prediction instruments. And yet, we are relying on polling such as online review systems evermore nowadays. The polls appear to be working in identifying good options even though votes are correlated. Our model implies that social influence helps self-interested individuals better solve optimization problems subject to uncertainty.

Third, we discover from our experiment with human subjects that the participants were able to reach the theoretically predicted maximum performance by selectively filtering the average opinion. This discovery suggests a new coordination mechanism for enhancing individual decision making. Potential applications of this mechanism include topics as diverse as mobile health, taxation, and urban planning.

Fourth, this work extends multi-agent optimal control to include human behaviors. It bridges engineering and social sciences in a way that can quantitatively describe the collective dynamics of interacting and intelligent individuals. Such multidisciplinary effort would have positive implications for both research communities. For it shows the possibility of incorporating human decision-making in engineered systems. Techniques such as optimal control would also be readily transferable to solving high-impact problems that have a strong social significance.

Last but not least, to our best knowledge, this is the first time that a human subject experiment was used to test a control-theoretic hypothesis. We faced many challenges in the process such as designing the experiment and analyzing the data. Our methodology would be helpful for future empirical research on multi-agent control in sociotechnical systems.

Bibliography

[Abbe *et al.*, 2012] Emmanuel A Abbe, Amir E Khandani, and Andrew W Lo. Privacy-preserving methods for sharing financial risk exposures. *The American Economic Review*, 102(3):65–70, 2012.

[Aharony *et al.*, 2011] Nadav Aharony, Wei Pan, Cory Ip, Inas Khayal, and Alex Pentland. Social fmri: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing*, 7(6):643–659, 2011.

[Alvarez, 2011] George A Alvarez. Representing multiple objects as an ensemble enhances visual cognition. *Trends in cognitive sciences*, 15(3):122–131, 2011.

[Apple, 2015] Apple. Researchkit for developers, 2015.

[Apple, 2016] Apple. Researchkit and carekit, 2016.

[Bergman and Donner, 1964] Göran Bergman and Kai Otto Donner. *An analysis of the spring migration of the common scoter and the long-tailed duck in southern Finland*. Societas pro fauna et flora Fennica, 1964.

[Besbes and Zeevi, 2015] Omar Besbes and Assaf Zeevi. On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science*, 61(4):723–739, 2015.

[Bloomberg and Krupp, 2014] Michael Bloomberg and Fred Krupp. The right way to develop shale gas, 2014.

[Boltzmann, 1877] Ludwig Boltzmann. Sitzungsberichte de kaiserlichen akademie der wissenschaften. *Wien*, 75:67, 1877.

[Bookstaber *et al.*, 2015] Richard Bookstaber, Paul Glasserman, Garud Iyengar, Yu Luo, Venkat Venkatasubramanian, and Zhizun Zhang. Process systems engineering as a modeling paradigm for analyzing systemic risk in financial networks. *The Journal of Investing*, 24(2):147–162, 2015.

[Brotman, 2014] Barbara Brotman. With energy reports, it's game on, 2014.

[Browder, 1965] Felix E Browder. Fixed-point theorems for noncompact mappings in hilbert space. *Proceedings of the National Academy of Sciences*, 53(6):1272–1276, 1965.

[Carver and Scheier, 1982] Charles S Carver and Michael F Scheier. Control theory: A useful conceptual framework for personality–social, clinical, and health psychology. *Psychological bulletin*, 92(1):111, 1982.

[Challenger and Rogers, 1986] Presidential Commission On Space Shuttle Challenger and WP Rogers. Report of the presidential commission on the space shuttle challenger accident, 1986.

[Champernowne and Cowell, 1998] David Gawen Champernowne and Frank Alan Cowell. *Economic inequality and income distribution*. Cambridge University Press, 1998.

[Champernowne, 1953] David G Champernowne. A model of income distribution. *The Economic Journal*, 63(250):318–351, 1953.

[Congress, 1998] US Congress. Digital millennium copyright act. *Public Law*, (105-304):112, 1998.

[Conradt and Roper, 2005] Larissa Conradt and Timothy J Roper. Consensus decision making in animals. *Trends in ecology & evolution*, 20(8):449–456, 2005.

[Craighill and Clement, 2014] Peyton M. Craighill and Scott Clement. What's right with political polling (a response to nate silver). *The Washington Post*, 2014.

[De Condorcet, 2014] Nicolas De Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Cambridge University Press, 2014.

[Edwards-Levy and Jackson, 2016] Ariel Edwards-Levy and Natalie Jackson. Do polls still work if people don't answer their phones? *The Huffington Post*, 2016.

[Efron and Morris, 1977] Bradley Efron and Carl Morris. Stein's paradox in statistics. *Scientific American*, 236:119–127, 1977.

[Galton, 1907a] Francis Galton. The ballot-box. *Nature*, 75:509–510, 1907.

[Galton, 1907b] Francis Galton. Vox populi (the wisdom of crowds). *Nature*, 75:450–451, 1907.

[Gold, 2014] Russell Gold. *The Boom: How Fracking Ignited the American Energy Revolution and Changed the World*. Simon and Schuster, 2014.

[Goldstone and Gureckis, 2009] Robert L Goldstone and Todd M Gureckis. Collective behavior. *Topics in cognitive science*, 1(3):412–438, 2009.

[Han *et al.*, 2006] Jing Han, Ming Li, and Lei Guo. Soft control on collective behavior of a group of autonomous agents by a shill agent. *Journal of Systems Science and Complexity*, 19(1):54–62, 2006.

[Harvey and Fischer, 1997] Nigel Harvey and Ilan Fischer. Taking advice: Accepting help, improving judgment, and sharing responsibility. *Organizational Behavior and Human Decision Processes*, 70(2):117–133, 1997.

[Higgins, 2014] Parker Higgins. How drm harms our computer security, 2014.

[Huizingh, 2011] Eelko KRE Huizingh. Open innovation: State of the art and future perspectives. *Technovation*, 31(1):2–9, 2011.

[James and Stein, 1961] William James and Charles Stein. Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379, 1961.

[Kandel and Lazear, 1992] Eugene Kandel and Edward P Lazear. Peer pressure and partnerships. *Journal of political Economy*, pages 801–817, 1992.

[Kennedy, 2010] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, 2010.

[Kennedy, 2011] James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.

[Kiefer *et al.*, 1952] Jack Kiefer, Jacob Wolfowitz, et al. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

[Kittur and Kraut, 2008] Aniket Kittur and Robert E Kraut. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 37–46. ACM, 2008.

[Krause *et al.*, 2010] Jens Krause, Graeme D Ruxton, and Stefan Krause. Swarm intelligence in animals and humans. *Trends in ecology & evolution*, 25(1):28–34, 2010.

[Krishnamurthy and Poor, 2014] Vikram Krishnamurthy and H Vincent Poor. A tutorial on interactive sensing in social networks. *Computational Social Systems, IEEE Transactions on*, 2014.

[Krupp, 2008] Fred Krupp. The making of a market-minded environmentalist, 2008.

[Leveson, 2011] Nancy Leveson. *Engineering a safer world: Systems thinking applied to safety*. Mit Press, 2011.

[Lim and O'Connor, 1995] Joa Sang Lim and Marcus O'Connor. Judgemental adjustment of initial forecasts: Its effectiveness and biases. *Journal of Behavioral Decision Making*, 8(3):149–168, 1995.

[Lorenz *et al.*, 2011] Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. How social influence can undermine the wisdom of crowd effect. *Proceedings of the National Academy of Sciences*, 108(22):9020–9025, 2011.

[Luo *et al.*, 2016] Yu Luo, Garud Iyengar, and Venkat Venkatasubramanian. Soft regulation with crowd recommendation: Coordinating self-interested agents in sociotechnical systems under imperfect information. *PLoS ONE*, 11(3):e0150343, 2016.

[Mani *et al.*, 2013] Ankur Mani, Iyad Rahwan, and Alex Pentland. Inducing peer pressure to promote cooperation. *Scientific reports*, 3, 2013.

[Motsch and Tadmor, 2014] Sebastien Motsch and Eitan Tadmor. Heterophilious dynamics enhances consensus. *SIAM review*, 56(4):577–621, 2014.

[Moussaïd *et al.*, 2013] Mehdi Moussaïd, Juliane E Kämmer, Pantelis P Analytis, and Hansjörg Neth. Social influence and the collective dynamics of opinion formation. *PLoS One*, 8(11):e78433, 2013.

[Nedic and Ozdaglar, 2009] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *Automatic Control, IEEE Transactions on*, 54(1):48–61, 2009.

[Olfati-Saber and Murray, 2002] Reza Olfati-Saber and Richard M Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*, volume 15, pages 242–248. Citeseer, 2002.

[Olfati-Saber *et al.*, 2007] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[Olver, 2008] Peter J. Olver. *Numerical Analysis Lecture Notes.* Peter J. Olver, 2008.

[Perthame and Tadmor, 1991] Benoît Perthame and Eitan Tadmor. A kinetic equation with kinetic entropy functions for scalar conservation laws. *Communications in Mathematical Physics*, 136(3):501–517, 1991.

[Phillips, 2012] Macon Phillips. Obama administration responds to we the people petitions on sopa and online piracy, 2012.

[Piketty *et al.*, 2014] Thomas Piketty, Arthur Goldhammer, and LJ Ganser. Capital in the twenty-first century. 2014.

[Powers, 1973] William T Powers. Feedback: beyond behaviorism stimulus-response laws are wholly predictable within a control-system model of behavioral organization. *Science*, 179(4071):351–356, 1973.

[Quinn and Bederson, 2011] Alexander J Quinn and Benjamin B Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1403–1412. ACM, 2011.

[Rai, 1999] Arti K Rai. Intellectual property rights in biotechnology: Addressing new technology. *Wake Forest L. Rev.*, 34:827, 1999.

[Rawlings and Mayne, 2009] James Blake Rawlings and David Q Mayne. *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[Rendell *et al.*, 2010] Luke Rendell, Robert Boyd, Daniel Cownden, Marquist Enquist, Kimmo Eriksson, Marc W Feldman, Laurel Fogarty, Stefano Ghirlanda, Timothy Lillicrap, and Kevin N Laland. Why copy others? insights from the social learning strategies tournament. *Science*, 328(5975):208–213, 2010.

[Reynolds, 1987] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH computer graphics*, 21(4):25–34, 1987.

[Saez and Zucman, 2014] Emmanuel Saez and Gabriel Zucman. Wealth inequality in the united states since 1913: Evidence from capitalized income tax data. Technical report, National bureau of economic research, 2014.

[Salganik *et al.*, 2006] Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *science*, 311(5762):854–856, 2006.

[Samuelson, 1966] Paul Samuelson. Science and stocks. *Newsweek, September*, 19:92, 1966.

[Sandholm, 2010] William H Sandholm. *Population games and evolutionary dynamics*. MIT press, 2010.

[Seborg *et al.*, 2010] Dale E Seborg, Duncan A Mellichamp, Thomas F Edgar, and Francis J Doyle III. *Process dynamics and control*. John Wiley & Sons, 2010.

[Shmueli *et al.*, 2014] Erez Shmueli, Vivek K Singh, Bruno Lepri, and Alex Pentland. Sensing, understanding, and shaping social behavior. *Computational Social Systems, IEEE Transactions on*, 2014.

[Simons, 2004] Andrew M Simons. Many wrongs: the advantage of group navigation. *Trends in ecology & evolution*, 19(9):453–455, 2004.

[Smith, 2011] Lamar Smith. Stop online piracy act. *US Government*, 2011.

[Soll and Larrick, 2009] Jack B Soll and Richard P Larrick. Strategies for revising judgment: How (and how well) people use others opinions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(3):780, 2009.

[Stein, 1956] Charles Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley symposium on mathematical statistics and probability*, volume 1, pages 197–206, 1956.

[Sumpter and Pratt, 2009] David JT Sumpter and Stephen C Pratt. Quorum responses and consensus decision making. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1518):743–753, 2009.

[Sumpter, 2006] David JT Sumpter. The principles of collective animal behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 361(1465):5–22, 2006.

[Sunstein and Hastie, 2014] Cass Sunstein and Reid Hastie. *Wiser: Getting Beyond Groupthink to Make Groups Smarter*. Harvard Business Review Press, 2014.

[Surowiecki, 2005] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[Tadmor, 2015] Eitan Tadmor. Mathematical aspects of self-organized dynamics consensus, emergence of leaders, and social hydrodynamics. *SIAM News*, 48, 2015.

[Tanner, 1984] R Michael Tanner. Explicit concentrators from generalized n-gons. *SIAM Journal on Algebraic Discrete Methods*, 5(3):287–293, 1984.

[Teel, 2004] Andrew R Teel. Discrete time receding horizon optimal control: Is the stability robust? In *Optimal Control, Stabilization and Nonsmooth Analysis*, pages 3–27. Springer, 2004.

[Thaler and Sunstein, 2003] Richard H Thaler and Cass R Sunstein. Libertarian paternalism. *American Economic Review*, pages 175–179, 2003.

[Thaler and Sunstein, 2008] Richard H Thaler and Cass R Sunstein. *Nudge: Improving decisions about health, wealth, and happiness*. Yale University Press, 2008.

[Tribe, 2011] Laurence H Tribe. The "stop online piracy act" (sopa) violates the first amendment, 2011.

[Trochim *et al.*, 2006] William M Trochim, Derek A Cabrera, Bobby Milstein, Richard S Gallagher, and Scott J Leischow. Practical challenges of systems thinking and modeling in public health. *American Journal of Public Health*, 96(3):538, 2006.

[Venkatasubramanian and Zhang, 2016] Venkat Venkatasubramanian and Zhizun Zhang. Tecsmart: A hierarchical framework for modeling and analyzing systemic risk in sociotechnical systems. *AIChE Journal*, 2016.

[Venkatasubramanian *et al.*, 2015] Venkat Venkatasubramanian, Yu Luo, and Jay Sethuraman. How much inequality in income is fair? a microeconomic game theoretic perspective. *Physica A: Statistical Mechanics and its Applications*, 435:120–138, 2015.

[Wang *et al.*, 2015] Wei Wang, David Rothschild, Sharad Goel, and Andrew Gelman. Forecasting elections with non-representative polls. *International Journal of Forecasting*, 31(3):980–991, 2015.

[West, 2012] Geoffrey West. The mathematical regularities underlying both biological and social systems, 2012.

[Yaniv and Kleinberger, 2000] Ilan Yaniv and Eli Kleinberger. Advice taking in decision making: Egocentric discounting and reputation formation. *Organizational behavior and human decision processes*, 83(2):260–281, 2000.

[Yaniv, 2004] Ilan Yaniv. Receiving other people's advice: Influence and benefit. *Organizational Behavior and Human Decision Processes*, 93(1):1–13, 2004.

[Zhang and Parkes, 2008] Haoqi Zhang and David C Parkes. Value-based policy teaching with active indirect elicitation. In *AAAI*, volume 8, pages 208–214, 2008.

Appendix A

Agent-Based Simulation with NetLogo

A.1 NetLogo Simulation Environment (p. 105) We briefly introduce the NetLogo simulation environment here. This simulation exercise corresponds to the results discussed in Section 4.1 (p. 53).

A.2 Script "SR.nlogo" (p. 106) This single NetLogo script includes the model of the agent-based simulation in Section 4.1 (p. 53).

A.1 NetLogo Simulation Environment

NetLogo offers a graphical user face for agent-based simulation (Fig. A.1, p. 105). This object-oriented programming language is widely used for motion-based simulation such as moving particles or flocking birds. Even though our simulation in Section 4.1 (p. 53) does not involve actual movement, we chose NetLogo for its ease of coding and the capability of handling a large number of parallel computations. The simulation interface can be partitioned into two. The left panel contains all the parameters and controls. The right panel contains all the plots.

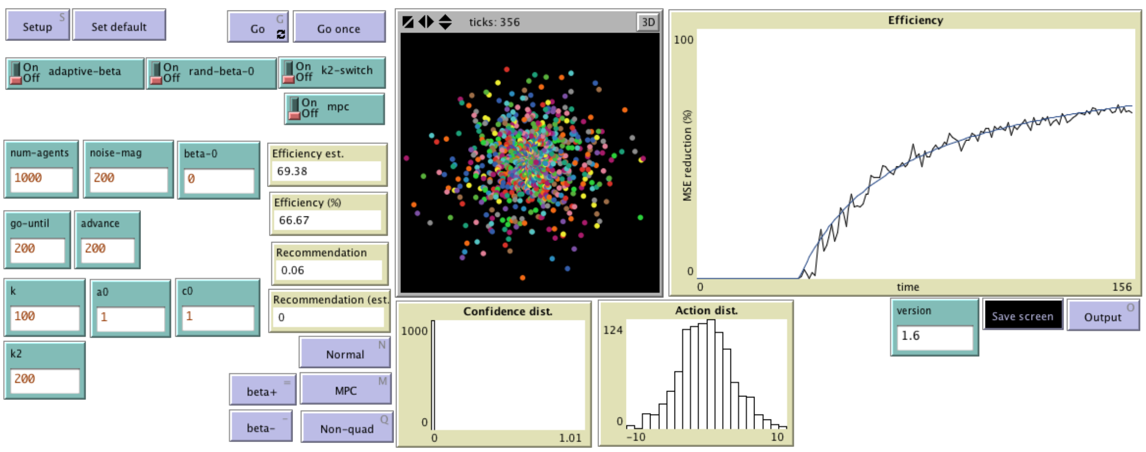


Figure A.1: **NetLogo interface**.

A.2 Script "SR.nlogo"

```
turtles-own [x beta x-delta beta-delta]
globals [u u-est u-0 a c MSE MSE-est MSE-0]

;; Basic functions

to default
  set num-agents 1000
  set go-until 200 ;; The max number of steps one simulation runs
  set noise-mag 200 ;; The uniform equivalent to a Gaussian noise (same variance)
  set adaptive-beta false
  set rand-beta-0 false ;; Initialize agents betas from a uniform distribution
  set k2-switch false ;; Piecewise quadratic
  set beta-0 0
  set k 100
  set k2 200 ;; Piecewise quadratic coefficient 2
  set a0 1 ;; KW parameter
  set c0 1
  set advance 200
  set mpc false
end

to setup
  clear-all
  reset-ticks
  random-seed 0

  setup-patches
  setup-turtles

  random-seed new-seed

  set u mean [x] of turtles
  set u-0 u
  set u-est u-0

  set MSE get-MSE
  set MSE-0 MSE
```

```
  set MSE-est MSE-0

  tick-advance advance
end

to setup-patches
  ask patches [ set pcolor black ]
end

to setup-turtles
  create-turtles num-agents
  ask turtles [
    set shape "dot"
    set x random-float 10
    ifelse rand-beta-0 [set beta random-float 1][set beta beta-0]
    ]
  move-turtles
end

to go
  ifelse ticks < go-until + advance
  [
    kw

    move-turtles ;; to current x_t

    set u-est get-u-est ;; estimate u_t
    set MSE get-MSE ;; compute MSE_t

    recommend ;; simulate u_t

    ask turtles
    [
      prepare
      update ;; get x_{t+1}
    ]

    set MSE-est get-MSE-est ;; estimate MSE_{t+1}
    tick ;; ticks ++ and update plots for ticks (to compare MSE_t and MSE-est_t)
```

```
  ][stop]
end

to move-turtles
  ask turtles [
      let deg who / num-agents * 360
      let r abs x
      if (abs x) > 10 [set r 10]
      setxy (r * cos(deg)) (r * sin(deg))
      ]
end

;; Defined functions

to-report f [arg-x]
  ifelse (k2-switch) and (arg-x > 0) [report 0 - k2 * arg-x ^ 2][report 0 - k * arg-x ^ 2]
end

to-report f-tilde [arg-x];; arg-beta arg-x-reg]
  ;;let omega (2 * (random-float 1) - 1) * noise-mag
  let omega random-normal 0 (noise-mag / (sqrt 3))
  report (f arg-x) + omega
end

to-report get-MSE
  report sum map [? ^ 2 / num-agents] [x] of turtles
end

to-report get-MSE-est
  ifelse ticks > 0
  [
    ifelse beta-0 = 1 [report MSE-0]
    [
    let sigma noise-mag / (sqrt 3)

    let num 2 * a ^ 2 * (1 - beta-0) ^ 2 * sigma ^ 2 / c ^ 2 + beta-0 * (2 * (1 - beta-0) * (1 - 4 * k * a) + 1) * u-est ^ 2
    let den 1 - (1 - beta-0) ^ 2 * (1 - 4 * k * a) ^ 2
```

```
    if mpc
    [
      set num a ^ 2 / c ^ 2 * (1 - beta-0) ^ 2 * ((1 - beta-0) ^ 2 * (1 - 4 * k * a) ^ 2 * sigma ^ 2 * 2 / num-agents + 2 * sigma ^ 2)
      ]

    ifelse den = 0 [report MSE-0][report num / den]
    ]
  ]
  [
    report MSE-0
  ]
end

to-report get-u-est
  let closed-form false
  ifelse closed-form
  [
    ifelse ticks - advance < 1
  [
    report u-0
    ]
  [
    let time n-values (ticks - advance) [? + advance]
    let prod reduce * map [1 - 4 * k * (get-a ?) * (1 - beta-0)] time
    report u-0 * prod
    ]
  ]
  [report u-0 * (ticks  / advance) ^ (0 - 4 * k * a0 * (1 - beta-0))]
end

to prepare
  set x-delta a * ((f-tilde (x + c)) - (f-tilde (x - c))) / c
end

to update
  ifelse adaptive-beta
  [
      set beta exp(0 - (x - u) ^ 2)
      set x (x + (1 - beta) * x-delta + beta * (u - x))
```

```
  ]
  [
    set x (x + (1 - beta) * x-delta + beta * (u - x))
  ]
end

to recommend
  ifelse ((not mpc) or (beta-0 = 0)) [set u mean [x] of turtles][set u 0 - (1 - beta-0) / beta-0 * (1 - 4 * k * a) * mean [x] of turtles]
end

to-report get-a [t]
  report a0 / t
end

to-report get-c [t]
  report c0 / t ^ (1 / 3)
end

to kw
  set a get-a ticks
  set c get-c ticks
end

to output-data
  set-current-directory "/Users/l16cn/Dropbox/DropWork/workspace/NetLogo/SoftReg/output"

  let beta-text beta-0
  let adaptive-text "" ;; adaptive beta
  let random-initial-beta-text "" ;; random beta-0
  let nonquad-text "" ;; asymmetric k
  let mpc-text "" ;; mpc switch

  if beta-0 = 0 [set beta-text "0.0"]
  if adaptive-beta [set adaptive-text "Adp"]
  if rand-beta-0 [
    set random-initial-beta-text "Unif"
    set beta-text ""
    ]
  if k2-switch [set nonquad-text "NQ"]
```

```
  if mpc [set mpc-text "MPC"]

  export-plot "Efficiency" (word adaptive-text random-initial-beta-text nonquad-text mpc-text "b=" beta-text ".csv" )
end
```

Appendix B

Amazon Mechanical Turk Experiment

B.1 Institutional Review Board (p. 113) We include the institutional review board (IRB) details here for conducting experiments with human subjects.

B.2 Amazon Mechanical Turk (p. 116) We include the Amazon Mechanical Turk details here such as the ad and sample result page.

B.1 Institutional Review Board

This human subject experiment is approved by the Columbia University Institutional Review Board. Included next are the approval letter and the information sheet of the experiment.

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

COLUMBIA INSTITUTIONAL REVIEW BOARD
MORNINGSIDE OFFICE

October 29, 2015

Venkat Venkatasubramanian
ENG Chemical Engineering - 521100X
819 S.W. Mudd
4721

Protocol Number: IRB-AAAQ2603
Title: Experiment to Test the Effectiveness of Soft Regulation with "Crowd Recommendation"
Approval Date: 10/19/2015   Expiration Date: 10/18/2020

Dear Professor Venkatasubramanian,

On October 19, 2015, the Chair of the Columbia University Institutional Review Board (IRB) reviewed the above mentioned protocol and determined that this research meets the criteria for exemption under category 2:

(2) Research involving the use of educational tests (cognitive, diagnostic, aptitude, achievement), survey procedures, interview procedures or observation of public behavior unless: (i) information obtained is recorded in such a manner that human subjects can be identified, directly or through identifiers linked to the subjects; and (ii) any disclosure of the human subjects' responses outside the research could reasonably place the subjects at risk of criminal or civil liability or be damaging to the subjects' financial standing, employability, or reputation.

Therefore, this research is exempt from further IRB review in accordance with the Department of Health and Human Service (HHS) regulations at 45 CFR Part 46.101(b)(2).

If you propose to change the protocol in any manner such that the criteria listed above no longer apply, you must submit a modification with the proposed changes to the IRB for review and approval prior to the implementation of the revised protocol.

For tracking purposes, you will be required to submit an abbreviated status report to the IRB prior to the expiration date listed above. The purpose of the submission is to confirm that the research is ongoing, which will facilitate accurate accounting at the University of all active research projects involving human subjects.

If you have any questions, please call Gloria Gaines at (212) 851-7043.

Columbia University appreciates your commitment towards the ethical conduct of human research.

Sincerely,

Annie Barry
Assistant Manager
Human Research Protection Office (HRPO)-Morningside

Electronically signed by: Barry, Annie

615 West 131st Street, 3rd Floor   New York, NY 10027
212-851-7040   Fax 212-851-7044   http://www.columbia.edu/cu/irb

Columbia University IRB
Approved for
use until: 10/18/2020

Figure B.1: **IRB approval letter**.

Columbia University in the City of New York
Institutional Review Board – Research Participant Consent Form

I. Purpose of the research study
You are invited to participate in a research study conducted by Yu Luo, Garud Iyengar, and Venkat Venkatasubramanian. The purpose of this study is to understand how certain information influences decision for a multi-player optimization problem.

II. What you will be asked to do
If you decide to be in this study, you will be asked to interact with an online interface. Your participation in this study will take approximately 10 minutes.

III. Foreseeable risks or discomforts
There is minimal risk to participating in this study. You may feel anxious if game score is low. You are free to withdraw from participation at any time.

IV. Benefits
While there may be no direct benefit to you from participating in this study, the indirect benefit of participating will be knowing that you helped researchers better understand how people respond to certain feedbacks during decision making processes.

V. Confidentiality
Any information provided by you will be kept in a password-protected Google Sheet. All data collected from you will be coded with a random alpha-numeric value. Your MTurk ID will never be recorded or shared.

VI. Compensation
If you participate in the study and successfully complete all sessions, the researcher will give you $0.50 to $5.00 based on your performance through your MTurk account.

VII. Voluntary nature of this research
Your participation in this study is completely voluntary. You do not have to do this, and you can quit at any time.

VIII. Contact information
If you have any question about this research, you may contact
softgamequestions@outlook.com

Figure B.2: **IRB information sheet**.

B.2 Amazon Mechanical Turk

Included below are the Amazon Mechanical Turk ad and result page. The payment rules are described as follows:

- Reward = $1.25 base pay upon approval + bonus (up to $3.75) based on performance
- Combined scores of the open loop and closed loop settings are ranked within each batch
- $5, $4, and $3 are awarded to the top 3 players
- $2 or higher if the player is ranked above the median
- $1.5 or higher if the player submits a valid confirmation code

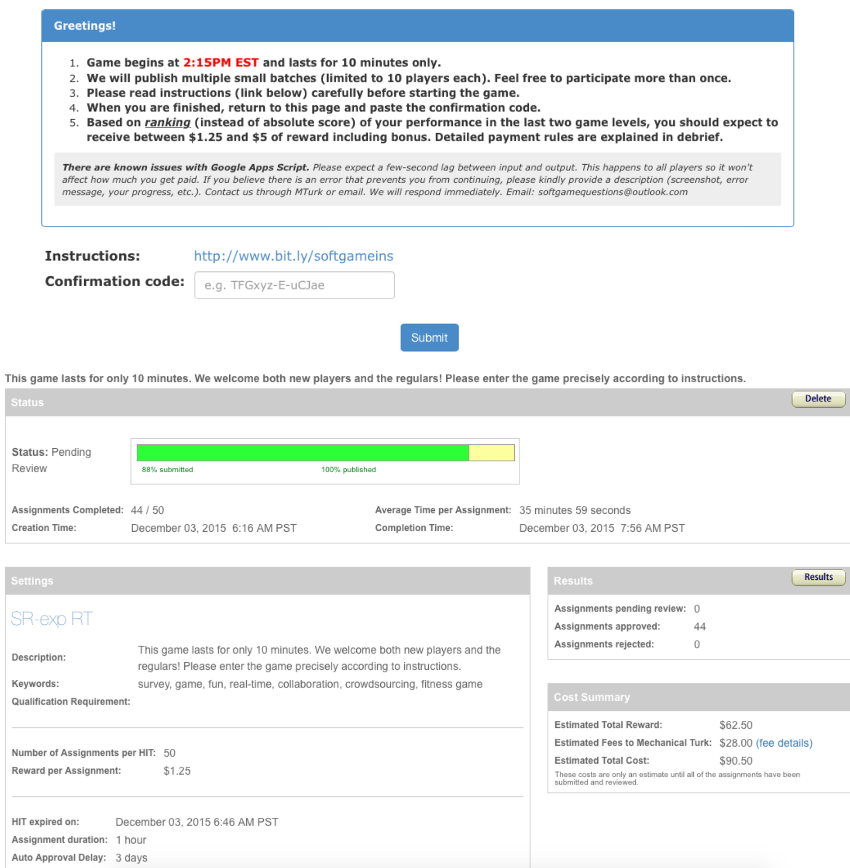


Figure B.3: **Amazon Mechanical Turk**.

Appendix C

Online Experiment with Google Apps Script

C.1 Google Apps Script (p. 118) We briefly introduce the Google Apps Script language here. It was used to program the "Fitness Game" interface in Section 4.2 (p. 65).

C.2 Script "Code.gs" (p. 119) This is the main code that includes the interactive features of the "Fitness Game."

C.3 Script "Engine.gs" (p. 133) This is the main function that computes the fitness level, score, feedback, etc.

C.1 Google Apps Script

Google Apps Script (GAS) is a scripting language for light-weight application development in the Google Apps platform. In the case of the "Fitness Game," we used GAS to create the game interface and Google Sheets to store data.

There are two scripts for the interface: "Code.gs" encodes the overall structure and "Engine.gs" encodes the core computations and data processing.

There are two Google Sheets that the game operates on: "Dashboard" (not included in this appendix) and "Data" (not included).

C.2 Script "Code.gs"

```
var version = "Software version 5.19"; //revert back to the base case, published

var modelPars = {
  thetaRange: 500,
  thetaLowerBound: 2000,
  roundingPrecision: 100,
  baseReward: 98,
  curvature: 1/2500,
  noiseMag: 2
}

var messageStyleOptions = {
  welcome:{background:'black',color:'white'},
  wait:{background:'gray',color:'white'},
  idle:{background:'white',color:'black'},
  error:{background:'red',color:'yellow'},
  outputLow:{background:'brown',color:'white'},
  outputOk:{background:'green',color:'white'},
  scoreBoard:{background:'white',color:'#0000A0'}
}

var messageOptions = {
  messageTextInput:'Enter a dietary suggestion between '+modelPars.thetaLowerBound+' and '
    +String(modelPars.thetaLowerBound+modelPars.thetaRange)+' for me',
  messageWaiting:'Calculating, please wait. Do not close this window! If the game freezes for more than a minute, contact us immediately.',
  messageNotS:" You just completed the previous session. <i>In this session, the ideal diet for me is different.</i> "+
  "We made the first guess of <b>2000 kcal</b> for you. ",
  messageB:"<p><b>You will also see a recommendation we calculate in real time from all players,"+
    "</b> which says <i>WE RECOMMEND (some value) kcal.</i></p>",
  messageSessionCompleted:'<p>Game over. Thanks for your participation.</p>',
  messageCloseReminder:'<p>(Click anywhere outside of it to close this dialog box)</p>',
  messageSubjectChats:['Please advise me a new diet!','Enter a new value please.','Other Turkers are ahead of you. New value please!',
    'Chop chop, new value please!','Tell me how much I should eat.','Game proceeds <i>after</i> you enter a new value. Be quick!',
    "Don't stop!",'I am waiting for a new value. Please write it down now.','Are you still there? Please enter a new value.',
    "So far so good. Keep going, Don't stop!"]
}
```

```
var chartOptions = {
  width:400*1.5,
  height:400,
  vAxis:{title:"My fitness",minValue:80,maxValue:100}
}

var scatterOptions = {
  title:"My fitness vs diet (10 most recent results)",
  series:{
    0:{pointSize:7.5,pointShape:'star',color:'green'},
    1:{pointSize:7.5,pointShape:'star',color:'brown'}},
  vAxis:{
    title:chartOptions.vAxis.title,
    gridlines:{count:5},
    minValue:chartOptions.vAxis.minValue,
    maxValue:chartOptions.vAxis.maxValue},
  hAxis:{
    title:'Diet (kcal)',
    minValue:modelPars.thetaLowerBound,
    maxValue:modelPars.thetaLowerBound+modelPars.thetaRange,gridlines:{count:6}}
}

var monitorOptions = {
  title:"My fitness progress (10 most recent results)",
  series:{
    0:{pointSize:3,color:'black'},
    1:{pointSize:3,color:'green',pointsVisible:false, lineDashStyle:[2,2]}
  },
  hAxis:{title:'Time (day)',minValue:1,maxValue:5,gridlines:{count:0}},
  vAxis:{title:chartOptions.vAxis.title,gridlines:{count:5},minValue:chartOptions.vAxis.minValue,maxValue:chartOptions.vAxis.maxValue}
}

var htmlOptions = {
  background:'white',
  color:'black',
  width:'400',
  height:'175'
}
```

```
var popupOptions = {
  style:{
    width:'400',
    height:'200',
    zIndex:'2',
    background:'orange'
  },
  position:{
    left:75,
    top:75
  }
}

function doGet(){

  var app = UiApp.createApplication().setTitle("Fitness game");
  var panel = app.createFocusPanel();
  var grid = app.createGrid(4, 2).setId('myGrid');


  var sheetDashboard = SpreadsheetApp.openById("19ql0qeSrzoj7rRHvdKlePe-rAMKT5y7lcyBMVWXA7gE").getSheetByName("Dashboard");
  var pars = sheetDashboard.getRange(1,2,12,1).getValues();
  var progOptions = {
    outputTarget:99,
    databaseId:sheetDashboard.getRange(12,2).getValue(),

    uniqueSessionId:pars[0][0],
    isSoftReg:eval(pars[2][0]),
    isLockActive:eval(pars[3][0]),
    isGameAccessible:eval(pars[4][0]),

    sessionSStartTime:pars[6][0],
    sessionAStartTime:pars[7][0],
    sessionBStartTime:pars[8][0],
    sessionAcceptNewEntryPeriod:Number(pars[9][0])
  }
```

```
var timestamp = new Date();
var sessionType = getSessionType(timestamp,progOptions).type;
var sessionProgress = getSessionType(timestamp,progOptions).progress;
var sessionTimeLeft = getSessionType(timestamp,progOptions).timeLeft;
var sessionSubject = getSessionType(timestamp,progOptions).subject;


var inputTextBox = app.createTextBox().setWidth('400px').setName('input').setId('input').setText(messageOptions.messageTextInput);
var userMessageHtml = app.createHTML('').setId('userMessage').setWidth('400px').setHeight('75px').setStyleAttributes(messageStyleOptions.welcome);
var subjectMessageHtml = app.createHTML(sessionSubject+': "Hello!"').setId('subjectMessage').setWidth('400px')
.setHeight('15px').setStyleAttributes(messageStyleOptions.welcome);
var progressBarImage = app.createHTML(sessionProgress).setId('progressBarImage')
var progressBarHtml = app.createHTML(sessionTimeLeft).setId('progressBarHtml')
var versionLabel = app.createLabel(version);

var randomStringValue = makeId();
var randomStringCallbackElement = app.createHidden('randomStringCallbackElement',randomStringValue).setId('randomStringCallbackElement');
app.add(randomStringCallbackElement);
var thetaValues = [precision(modelPars.thetaLowerBound + Math.random() * modelPars.thetaRange),//sandbox session
                   precision(modelPars.thetaLowerBound + Math.random() * modelPars.thetaRange),//session A
                   precision(modelPars.thetaLowerBound + Math.random() * modelPars.thetaRange)];//session B
var thetaCallbackElement = app.createHidden('thetaCallbackElement', JSON.stringify(thetaValues)).setId('thetaCallbackElement');
app.add(thetaCallbackElement);
var datasource = [];
var datasourceCallbackElement = app.createHidden('datasourceCallbackElement',JSON.stringify(datasource)).setId('datasourceCallbackElement');
app.add(datasourceCallbackElement);
var progOptionsCallbackElement = app.createHidden('progOptionsCallbackElement',JSON.stringify(progOptions)).setId('progOptionsCallbackElement');
app.add(progOptionsCallbackElement);
var allScores = [0,0,0];
var allScoresCallbackElement = app.createHidden('allScoresCallbackElement',JSON.stringify(allScores)).setId('allScoresCallbackElement');
app.add(allScoresCallbackElement);

var scoreProgressHtml = app.createHTML(getScoreHtml(allScores)).setId('scoreProgressHtml')
.setStyleAttributes(messageStyleOptions.scoreBoard).setWidth('400px');

var uniqueSubSessionId = progOptions.uniqueSessionId+'-'+sessionType;

if(sessionType!='S'){getPopup(app,'Warning: The previous session has expired.',false);return app}
else{getPopup(app,getCurrentPanelMessage(timestamp,progOptions),true)}
```

```
if(!progOptions.isGameAccessible && timestamp.valueOf() < Date.parse(progOptions.sessionSStartTime)){
  getPopup(app,
           'The game has not begun yet! Please come back later or refresh the page at '+
           Utilities.formatDate(progOptions.sessionSStartTime, "America/New_York", "h:mm a' Eastern Time'")+'.',false);
  return app
}


var scatterData = Charts.newDataTable()
.addColumn(Charts.ColumnType.NUMBER, 'input')
.addColumn(Charts.ColumnType.NUMBER, 'Above target')
.addColumn(Charts.ColumnType.NUMBER, 'Below target')
.addRow([0,'',''])
.build();
var scatterChart = Charts.newScatterChart()
.setDataTable(scatterData)
.setDimensions(chartOptions.height,chartOptions.height)
.setTitle(scatterOptions.title)
.setOption('series',scatterOptions.series)
.setOption('hAxis',scatterOptions.hAxis)
.setOption('vAxis',scatterOptions.vAxis)
.setLegendPosition(Charts.Position.BOTTOM)
.build();

var monitorData = Charts.newDataTable()
.addColumn(Charts.ColumnType.NUMBER,'time')
.addColumn(Charts.ColumnType.NUMBER,'Fitness')
.addColumn(Charts.ColumnType.NUMBER,'Target')
.addRow([1,'',''])
.build();
var monitorChart = Charts.newLineChart()
.setDataTable(monitorData)
.setDimensions(chartOptions.width,chartOptions.height)
.setTitle(monitorOptions.title)
.setLegendPosition(Charts.Position.BOTTOM)
.setOption('series',monitorOptions.series)
.setOption('hAxis',monitorOptions.hAxis)
.setOption('vAxis',monitorOptions.vAxis)
```

```
.build();

var textboxHandler = app.createServerHandler('insertInSS')
.addCallbackElement(thetaCallbackElement)
.addCallbackElement(randomStringCallbackElement)
.addCallbackElement(datasourceCallbackElement)
.addCallbackElement(progOptionsCallbackElement)
.addCallbackElement(allScoresCallbackElement)
.addCallbackElement(panel)
.validateLength(inputTextBox,4,4).validateRange(inputTextBox, modelPars.thetaLowerBound, modelPars.thetaLowerBound+modelPars.thetaRange)
.validateNotMatches(userMessageHtml,messageOptions.messageWaiting);

var textboxOutOfBoundInputHandler = app.createClientHandler()
.forTargets(userMessageHtml).setStyleAttributes(messageStyleOptions.error)
.setHTML('Invalid input!<br>Please enter a diet between '+modelPars.thetaLowerBound+' and '+
  String(modelPars.thetaLowerBound+modelPars.thetaRange)+'.')
.validateLength(inputTextBox,4,4).validateNotRange(inputTextBox,modelPars.thetaLowerBound, modelPars.thetaLowerBound+modelPars.thetaRange);
var textboxOutOfBoundInputClearHandler = app.createClientHandler()
.forEventSource().setText('')
.validateLength(inputTextBox,4,4).validateNotRange(inputTextBox,modelPars.thetaLowerBound, modelPars.thetaLowerBound+modelPars.thetaRange);

var textboxClearOnClickHandler = app.createClientHandler()
.forEventSource().setText('');

var textboxFreezeAfterInputHandler = app.createClientHandler()
.forEventSource().setEnabled(false).setStyleAttributes(messageStyleOptions.wait)
.forTargets(userMessageHtml).setText(messageOptions.messageWaiting).setStyleAttributes(messageStyleOptions.wait)
.forTargets(subjectMessageHtml).setStyleAttributes(messageStyleOptions.wait)
.validateLength(inputTextBox,4,4).validateRange(inputTextBox, modelPars.thetaLowerBound, modelPars.thetaLowerBound+modelPars.thetaRange);

inputTextBox
.addKeyUpHandler(textboxHandler)
.addKeyUpHandler(textboxFreezeAfterInputHandler)
.addKeyUpHandler(textboxOutOfBoundInputHandler)
.addKeyUpHandler(textboxOutOfBoundInputClearHandler)
.addClickHandler(textboxClearOnClickHandler);

grid
.setWidget(0,0,userMessageHtml)
```

```
  .setWidget(0,1,scoreProgressHtml)//progressBarImage)
  .setWidget(1,0,subjectMessageHtml)
  .setWidget(1,1,progressBarHtml)
  .setWidget(2,0,inputTextBox)
  .setWidget(2,1,versionLabel)
  .setWidget(3,1,monitorChart)
  .setWidget(3,0,scatterChart);

  panel.add(grid);app.add(panel);

  return app;
}

function insertInSS(e){


  var app = UiApp.getActiveApplication();
  var timestamp = new Date();
  var progOptions = JSON.parse(e.parameter.progOptionsCallbackElement);
  var allScores = JSON.parse(e.parameter.allScoresCallbackElement);
  var databaseId = progOptions.databaseId;
  var sheetDb = SpreadsheetApp.openById(databaseId);
  var sheetData = sheetDb.getSheetByName('Data');
  var sheetUsers = sheetDb.getSheetByName('Users');

  var uniqueSessionId = progOptions.uniqueSessionId;
  var sessionType = getSessionType(timestamp,progOptions).type;
  var sessionProgress = getSessionType(timestamp,progOptions).progress;
  var sessionTimeLeft = getSessionType(timestamp,progOptions).timeLeft;
  var sessionIndex = getSessionType(timestamp,progOptions).index;
  var sessionSubject = getSessionType(timestamp,progOptions).subject;


  var randomStringValue = e.parameter.randomStringCallbackElement;
  var uniqueSubSessionId = uniqueSessionId+'-'+sessionType;
  var uniqueUserId = uniqueSubSessionId+'-'+String(randomStringValue);
  var isSR = false;if (sessionType == 'S'){isSR = progOptions.isSoftReg};if (sessionType == 'B'){isSR = true};
```

```
if(sessionType=='E'){
  var combinedScore = allScores[1]+allScores[2];
  var endMessage = messageOptions.messageSessionCompleted + '<p>MTurk confirmation code is <b>' + uniqueUserId +
    '. Please use right click to copy the code. Ctrl+C does not work here.</b></p> <p>You scored '+
      combinedScore+
      ' point(s) in total from the last two levels in the game. Once approved, your participation will be compensated based on its ranking.</p>';
  getPopup(app,endMessage,false);
  return app;
}else{



  var input = Number(e.parameter.input);app.getElementById('input').setValue('');
  var thetaValue = JSON.parse(e.parameter.thetaCallbackElement)[sessionIndex];

  try{
    var datasource = JSON.parse(e.parameter.datasourceCallbackElement);
    }
  catch(e){
    Utilities.sleep(1000)

    var datasource = [];
  }

  try{
    var answers = engine(timestamp,uniqueUserId,uniqueSubSessionId,input,thetaValue,sessionType,
      progOptions,datasource,sheetData,sheetUsers,allScores);
    datasource = answers.datasource;
    allScores = answers.allScores;
    app.getElementById('datasourceCallbackElement').setValue(JSON.stringify(datasource));
    app.getElementById('allScoresCallbackElement').setValue(JSON.stringify(allScores));
    var output = answers.output;
    var monitorChart = answers.monitorChart;
    var scatterChart = answers.scatterChart;
    var softRecommendation = precision(Math.min(modelPars.thetaLowerBound+modelPars.thetaRange,
      Math.max(modelPars.thetaLowerBound,answers.recUser)));
```

```
var dayCounter = answers.dayCounter;
var scoreCounter = answers.scoreCounter;
var error = answers.error;
if((sessionType!='S')&&(dayCounter==1)){
  getPopup(app,getCurrentPanelMessage(timestamp,progOptions),true);
  input = 2000;
}

var inputAverageMessage = '';
if(isSR){inputAverageMessage = '<span style="background-color: #0000FF"><b><i>WE RECOMMEND '+softRecommendation+' kcal</i></b></span>'};


var userMessage = "<b>Day "+dayCounter+"</b>:";
userMessage = userMessage + "<br>Diet = "+input+" kcal "+inputAverageMessage+" <br>Fitness = "+output+"&#37;";
if (output<progOptions.outputTarget){
  userMessage=userMessage+" (below target "+progOptions.outputTarget+"&#37;)"
  app
  .getElementById('userMessage')
  .setHTML(userMessage)
  .setStyleAttributes(messageStyleOptions.outputLow)
}
else{
  userMessage=userMessage+" (above target "+progOptions.outputTarget+"&#37;)"
  app
  .getElementById('userMessage')
  .setHTML(userMessage)
  .setStyleAttributes(messageStyleOptions.outputOk)
}
userMessage = userMessage +"<br>Score = "+scoreCounter;
var subjectMessage = sessionSubject+': "'+messageOptions.messageSubjectChats[Math.floor(Math.random() *
  messageOptions.messageSubjectChats.length)]+'"';
if(error){
  app.getElementById('userMessage')
  .setStyleAttributes(messageStyleOptions.error)
  .setHTML('Uh oh, what a coincidence! It seems that another Turker is trying to access the database at EXACTLY the same time as you.'
         + ' Unfortunately, your latest input <b>'+input+'</b> is not saved. Could you enter it again?')
}
else{app.getElementById('userMessage').setHTML(userMessage);}
app.getElementById('subjectMessage').setHTML(subjectMessage).setStyleAttributes(messageStyleOptions.welcome);
```

```
      app.getElementById('input').setEnabled(true).setText('').setStyleAttributes(messageStyleOptions.idle).setFocus(true);
      app.getElementById('progressBarHtml').setHTML(sessionTimeLeft);
      app.getElementById('scoreProgressHtml').setHTML(getScoreHtml(allScores))//.setHTML(sessionProgress);
      app.getElementById('myGrid').setWidget(3,1,monitorChart);
      app.getElementById('myGrid').setWidget(3,0,scatterChart);
    }catch(e){
      Utilities.sleep(1000)
      app.getElementById('userMessage').setStyleAttributes(messageStyleOptions.error)
      .setHTML('Oops, '+(e.message)+'.<br>Your latest input <b>'+input+'</b> is not saved. Could you enter it again?');
      var subjectMessage = sessionSubject+': "'+messageOptions.messageSubjectChats[Math.floor(Math.random() *
        messageOptions.messageSubjectChats.length)]+'"';
      app.getElementById('subjectMessage').setHTML(subjectMessage).setStyleAttributes(messageStyleOptions.welcome);
      app.getElementById('input').setEnabled(true).setText('').setStyleAttributes(messageStyleOptions.idle).setFocus(true);
      return app;
    }


    return app;

  }

}

function artificialLock(){
  var lock = LockService.getScriptLock();
  lock.waitLock(1000)
  Utilities.sleep(25000);
  lock.releaseLock();
}


//misc functions
function precision(x){
  return Math.round(x * modelPars.roundingPrecision) / modelPars.roundingPrecision
}
function f(x,thetaValue){
  return precision(modelPars.baseReward - modelPars.curvature * (x - thetaValue) *
    (x - thetaValue) + modelPars.noiseMag * (2 * Math.random() - 1));
}
```

```
function makeId()
{
  var text = "";
  var possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";

  for( var i=0; i < 5; i++ )
    text += possible.charAt(Math.floor(Math.random() * possible.length));

  return text;
}


function getDuration(time){
  if (time > 24*60*60*1000){return 'in about ' + Math.floor(time/(24*60*60*1000))+' day(s).'}
  else{
    if (time > 60*60*1000){return 'in about ' + Math.floor(time/(60*60*1000))+' hour(s).'}
    else{
      if (time > 60*1000){return 'in about '+Math.floor(time/(60*1000))+' minute(s).'}
      else{
        if (time > 1000){return 'in about '+Math.floor(time/1000)+' second(s).'}
        else{
          if (time > 0){return 'in about a second. Enter a new value now!'}
          else{return 'now!'}
        }
      }
    }
  }
}

function getSessionType(timestamp,progOptions){

  var timeValue = timestamp.valueOf();
  var sessionSStart = Date.parse(progOptions.sessionSStartTime);
  var sessionAStart = Date.parse(progOptions.sessionAStartTime);
  var sessionBStart = Date.parse(progOptions.sessionBStartTime);
  var sessionBEnd = Date.parse(progOptions.sessionBStartTime)+progOptions.sessionAcceptNewEntryPeriod*60*1000;

  var sessionSDuration = sessionAStart - sessionSStart;
  var sessionADuration = sessionBStart - sessionAStart;
```

```
var sessionBDuration = sessionBEnd - sessionBStart;
var line = '&#8226;&#8226;&#8226;&#8226;&#8226;&#8226;&#8226;&#8226;&#8226;&#8226;';

if(timeValue < sessionSStart){
  return {
    type:'S',
    name:'Level 0 (Practice Mode)',
    subject:'Joe',
    index:0,
    progress:'Level 0'+line+'0&#37;'+'<br>'
    +'Level 1'+line+'0&#37;'+'<br>'
    +'Level 2'+line+'0&#37;',
    timeLeft:'Level 0 ends '+getDuration(sessionAStart - timeValue)
  }
}else{
  if(timeValue < sessionAStart){
    return {
      type:'S',
      name:'Level 0 (Practice Mode)',
      subject:'Joe',
      index:0,
      progress:'Level 0'+line+String(Math.round((timeValue-sessionSStart)/sessionSDuration*100))+'&#37;'+'<br>'
      +'Level 1'+line+'0&#37;'+'<br>'
      +'Level 2'+line+'0&#37;',
      timeLeft:'Level 0 ends '+getDuration(sessionAStart - timeValue)
    }
  }else{
    if(timeValue < sessionBStart){
      return {
        type:'A',
        name:'Level 1',
        subject:'Andy',
        index:1,
        progress:'Level 0'+line+'100&#37;'+'<br>'
        +'Level 1'+line+String(Math.round((timeValue-sessionAStart)/sessionADuration*100))+'&#37;'+'<br>'
        +'Level 2'+line+'0&#37;',
        timeLeft:'Level 1 ends '+getDuration(sessionBStart - timeValue)
      }
    }else{
```

```
        if(timeValue < sessionBEnd){
          return {
            type:'B',
            name:'Level 2',
            subject:'Ben',
            index:2,
            progress:'Level 0'+line+'100&#37;'+'<br>'
            +'Level 1'+line+'100&#37;'+'<br>'
            +'Level 2'+line+String(Math.round((timeValue-sessionBStart)/sessionBDuration*100))+'&#37;',
            timeLeft:'Level 2 ends '+getDuration(sessionBEnd - timeValue)
          }
        }else{
          return {
            type:'E',
            name:'End of game',
            subject:'Ben',
            index:2,
            progress:'Level 0'+line+'100&#37;'+'<br>'
            +'Level 1'+line+'100&#37;'+'<br>'
            +'Level 2'+line+'100&#37;',
            timeLeft:'You have completed this experiment.'
          }
        }
      }
    }
  }
}

function getPopup(app,message,autoHide){
  var popupMessageHtml = app.createHTML(message).setStyleAttributes(htmlOptions);
  var popupMessagePanel = app.createPopupPanel(false, true);
  popupMessagePanel
  .add(popupMessageHtml)
  .setPopupPosition(popupOptions.position.left,popupOptions.position.top).setStyleAttributes(popupOptions.style)
  .setGlassEnabled(false).setAnimationEnabled(true).setAutoHideEnabled(autoHide)
  .show();

  if(!autoHide){app.getElementById('myGrid').setVisible(false)}
}
```

```
function getCurrentPanelMessage(timestamp,progOptions){
  var thisSession = getSessionType(timestamp,progOptions);
  var panelHtml = "<p><b>"+thisSession.name+"</b></p><br> Hello, my name is "+thisSession.subject+
    ". Welcome to "+thisSession.name+"! Please help me maintain a high fitness.";
  if (thisSession.type!='S'){panelHtml=panelHtml+messageOptions.messageNotS};
  if (thisSession.type=='B'){panelHtml=panelHtml+messageOptions.messageB};
  panelHtml = panelHtml + messageOptions.messageCloseReminder;
  return panelHtml;
}

function getScoreHtml(allScores){
  var message = "YOUR SCORES<br>Level 0:     <b>"+allScores[0]+
    "</b><br>Level 1:     <b>"+allScores[1]+
      "</b><br>Level 2:     <b>"+allScores[2]+"</b>";
  return message
}
```

C.3 Script "Engine.gs"

```
function engine(timestamp,uniqueUserId,uniqueSubSessionId,input,thetaValue,sessionType,progOptions,datasource,sheetData,sheetUsers,allScores){
  try{
    //Reset history in a new session
    if(datasource.length>0){
      if(datasource[datasource.length-1].uniqueSubSessionId!=uniqueSubSessionId){
        datasource = [];
        input = 2000;
      }
    }

    //Chart datatable
    var scatterData = Charts.newDataTable()
    .addColumn(Charts.ColumnType.NUMBER, 'input')
    .addColumn(Charts.ColumnType.NUMBER, 'Above target')
    .addColumn(Charts.ColumnType.NUMBER, 'Below target');
    var monitorData = Charts.newDataTable()
    .addColumn(Charts.ColumnType.NUMBER,'time')
    .addColumn(Charts.ColumnType.NUMBER,'Fitness')
    .addColumn(Charts.ColumnType.NUMBER,'Target');

    //Initialize variables assuming empty datasource
    var output = f(input,thetaValue);
    var isSR = false;if (sessionType == 'S'){isSR = progOptions.isSoftReg};if (sessionType == 'B'){isSR = true};
    var prevInput = '';
    var prevInputAverage = '';
    var outputTarget = progOptions.outputTarget;
    var dayCounter = 1;
    if(output>=outputTarget){var scoreCounter = 1;scatterData.addRow([input,output,''])}
    else{var scoreCounter = 0;scatterData.addRow([input,'',output])}

    //Populate chart data from the latest 10 entries
    if(datasource.length>0){

      for (i=0;i<datasource.length;i++){

        if (datasource [i].output>=outputTarget){scatterData.addRow([datasource [i].input+thetaValue ,datasource [i].output,''])}
        else{scatterData.addRow([datasource [i].input+thetaValue ,'',datasource [i].output])}
```

```
    monitorData.addRow([datasource [i].dayCounter,datasource [i].output,outputTarget])

  }

  prevInput = datasource [datasource.length-1].input;
  dayCounter = datasource[datasource.length-1].dayCounter + 1;
  if (output>=outputTarget){scoreCounter = datasource[datasource.length-1].scoreCounter + 1}
    else{scoreCounter = datasource[datasource.length-1].scoreCounter};
  if(isSR){prevInputAverage = datasource[datasource.length-1].inputAverage;}

}
monitorData.addRow([dayCounter,output,outputTarget]);
scatterData.build();monitorData.build();

if(sessionType=='S'){allScores[0]=scoreCounter}
if(sessionType=='A'){allScores[1]=scoreCounter}
if(sessionType=='B'){allScores[2]=scoreCounter}

var monitorChart = Charts.newLineChart()
.setDataTable(monitorData)
.setDimensions(chartOptions.width,chartOptions.height)
.setTitle(monitorOptions.title)
.setLegendPosition(Charts.Position.BOTTOM)
.setOption('series',monitorOptions.series)
.setOption('hAxis',monitorOptions.hAxis)
.setOption('vAxis',monitorOptions.vAxis)
.build();

var scatterChart = Charts.newScatterChart()
.setDataTable(scatterData)
.setDimensions(chartOptions.height,chartOptions.height)
.setTitle(scatterOptions.title)
.setOption('series',scatterOptions.series)
.setOption('hAxis',scatterOptions.hAxis)
.setOption('vAxis',scatterOptions.vAxis)
.setLegendPosition(Charts.Position.BOTTOM)
.build();
```

```
//Soft regulation
var inputAverage = '';
if(isSR){

  var usersLastRow = sheetUsers.getLastRow();
  var userIds = [].concat.apply([],sheetUsers.getRange(1,2,usersLastRow,1).getValues());
  var subSessionIds = [].concat.apply([],sheetUsers.getRange(1,3,usersLastRow,1).getValues());
  var inputs = [].concat.apply([],sheetUsers.getRange(1,5,usersLastRow,1).getValues());

  //Initialize variables assuming empty users list
  var newUser = true;
  var inputSum = 0;
  var inputAverage =  input - thetaValue; //calibrated input
  var numOfUsers = 0;

  if(userIds.length>1){

    for (i=1;i<userIds.length;i++){
      if(subSessionIds[i]==uniqueSubSessionId){
        numOfUsers = numOfUsers + 1;

        if(userIds[i]==uniqueUserId){
          newUser = false;
          inputSum = inputSum + input - thetaValue;
          var userRow = 1 + i;
        }
        else{inputSum = inputSum + inputs[i]}

      }
    }
  }

  if(numOfUsers > 0){inputAverage = inputSum / numOfUsers}
  if(newUser){inputAverage = (inputSum + input - thetaValue)/(numOfUsers + 1)}
  var writeUsers = [timestamp,uniqueUserId,uniqueSubSessionId,thetaValue,input - thetaValue,output,inputAverage,dayCounter,scoreCounter];
  if(!newUser){sheetUsers.getRange(userRow,1,1,writeUsers.length).setValues([writeUsers])}
}

//Update history
```

```
  var newDatasource = {
    uniqueSubSessionId:uniqueSubSessionId,
    input:(input - thetaValue),
    output:output,
    dayCounter:dayCounter,
    scoreCounter:scoreCounter,
    inputAverage:inputAverage,
    dayCounter:dayCounter,
    scoreCounter:scoreCounter
  }

  if(datasource.length>8){datasource.shift()}
  datasource.push(newDatasource);
}catch(e){
  Utilities.sleep(1000);
  return {datasource:[]}
}


var appendContent = [timestamp,uniqueUserId,uniqueSubSessionId,thetaValue,input - thetaValue,output,inputAverage,dayCounter,scoreCounter,
prevInput,input-thetaValue,prevInputAverage,input-thetaValue];

try{
  var isLockActive = true;
  //Only when population is small
  //Utilities.sleep(4000);

  if(isLockActive){
    var lock = LockService.getScriptLock();
    lock.waitLock(5000)}

  sheetData.appendRow(appendContent);
  if(isSR & newUser){sheetUsers.appendRow(writeUsers)}
  SpreadsheetApp.flush();

  if(isLockActive){lock.releaseLock()}

  return {
    datasource:datasource,
```

```
        monitorChart:monitorChart,
        scatterChart:scatterChart,
        output:output,
        recUser:(inputAverage + thetaValue),
        error:false,
        dayCounter:dayCounter,
        scoreCounter:scoreCounter,
        allScores:allScores
      }

    }
    catch(e){

      return {
        datasource:datasource,
        monitorChart:monitorChart,
        scatterChart:scatterChart,
        output:output,
        recUser:(inputAverage + thetaValue),
        error:false,
        dayCounter:dayCounter,
        scoreCounter:scoreCounter,
        allScores:allScores
      }
    }
  }
```

Appendix D

Data Analysis with MATLAB

D.1 Script "readme.m" (p. 139) This is an inventory of the scripts, functions, and important variables used for the data analysis in Section 4.2 (p. 65). There are many scripts used here. We only include the most important two in the following sections.

D.2 Script "meanValue.m" (p. 142) This script computes the average MSE from the five replications within an experiment set. It also passes the processed data to "sysid.m" for further system identification and optimal control procedures.

D.3 Script "sysid.m" (p. 149) This script takes in the average MSE data and estimate the learning function, noise, and social influence. It also estimates the optimal social influence that would maximally reduce the expected cumulative MSE.

D.1 Script “readme.m”

```
%% SR2 Manuscript Data Analysis Documentation
% Last updated 10/8/16

%% Scripts based on execution order (B series)
% # *script_importData*: import 'all_data.csv' (with imputation) into *DataTable* in 'all_data_table.mat'
% # *script_importDataTax*: import 'STC_Historical_DB.csv' and
% 'STC_Historical_DB_Description.csv' and plot the time series ('TaxFig1##.pdf')
% # *script_importDataExp*: import 'TPC_Expenditure_DB.csv' and
% 'TPC_Expenditure_Description.csv' and plot the time series ('ExpFig1###.pdf')
% # *script_phaseDiagram*: plot robust *betaRobust* as a phase diagram in
% different combinations of noise ratio, time horizon, and learning gain
% ('BasicFig5#.pdf')
% # *script_meanValue*: plot the *x* time series as well as *u*
% ('BasicFig11.pdf' and 'BasicFig12.pdf'); store MSE data in
% 'meanValue_Basic.mat'
% # *script_sysid*: system identification and optimal control for the
% experiment data ('BasicFig3#.pdf')
% # *script_sysidTax*: system identification and optimal control for the
% tax data ('TaxFig3##.pdf')
% # *script_sysidExp*: system identification and optimal control for the
% expenditure data ('ExpFig3###.pdf')

%% Raw data attributes ('all_data.csv')
% # *batch*: name of the batch/series ('Basic', 'Noisy', 'Small')
% # *DataTable*: table of all data
% # *day*: number of guesses (1, 2, ...)
% # *gameId*: gameName + session ('F50b12-A', ...)
% # *gameName*: 'F' + nSessionCapacity + index ('F50b12', ...)
% # *score*: number of score points (0, 1, ...)
% # *session*: name of session ('S', 'A', 'B')
% # *theta*: ideal diet in kcals (2481, 2300, ...)
% # *time*: number of seconds in this session (0.04, 1.02, ...)
% # *timestamp*: timestamp ('08-Jun-2016 10:02:00', ...)
% # *u*: recommendation (0.77, ...)
% # *uPrev*: previous recommendation
% # *user*: gameName + session + uniqueCode ('F50b12-A-qJu2O', ...)
% # *x*: state of current guess = guess - theta (-5.4, 20, ...)
```

```
% # *xCurr*: same as x
% # *xPrev*: previous state
% # *y*: fitness from current state (96.4, ...)

%% Library (pkg.*)
% # *vq = digitize(x, v, xq)*: return digitized/discretized list vq based
% on original (x, v) series and evaluate at xq
% # *betaBest = findBetaBest(noiseRatio, gFunGain, nTime, isMseSum)*: find
% the *betaRobust* based on the noise ratio, learning gain, and time
% horizon
% # *[h, Data] = plotExp(expId, ExpData, ExpName)*: plot 'ExpFig1###.pdf'
% # *[h, Data] = plotTax(taxId, TaxData, TaxName)*: plot 'TaxFig1##.pdf'
% # *y = polyvalAr(p, y0, nTime)*: return y(nTime) based on y(t+1) =
% polyval(p, y(t)) and y(1) = y0
% # *yInf = polyvalArInf(p)*: return y(inf) based on y(t+1) =
% polyval(p, y(t))
% # *r2 = rsq(p,x,y)*: return *rsq* based on model and observed data
% # *r2 = rsqEmpirical(y, yFitted)*: return *rsq* based on model data and
% observed data
% # *[mseList, betaDynList, stateMatrix] = sysval(vars, beta, tau, Pars)*:
% Monte Carlo simulation of the MSE time series based on system parameters


%% Frequently used non-structure variables
% # *batchKey*: which batch/set to be analyzed ('Basic', ...)
% # *beta*: social influence
% # *betaId*: estimated social influence based on experiment data
% # *betaBest*: optimal social influence based on MC simualtion (true best)
% # *betaRobust*: optimal social influence based on robsut control
% # *betaDyn*: optimal dynamic social influence that changes over time
% # *diagonalPoly*: [0, 0, ..., 0, 1, 0] represents f(x) = x diagonal
% # *gain*: first derivative of *gFun*, i.e., the learning gain
% # *gameList*: a cell to store *Game* structures
% # *gameNameListUnique*: list of games (usually 5 in a batch)
% # *gFun*: (linear) learning function
% # *mse*: mean squared error
% # *nTime*: number of discretized time steps in equal increments
% # *nPlayer*: number of players (in a batch, or in a game)
% # *nPolyOrder*: polyfit highest order (linear or quadratic)
```

```
% # *nSample*: number of Monte Carlo samples
% # *opiDist*: opinion distance
% # *phat*: (usually) polyfit parameters
% # *playerList*: a cell to store *Player* structures
% # *playerNameListUnique*: list of the last five characters of each unique *user*
% # *progress*: progress temporary integer
% # *tau*: exponential beta profile parameter
% # *tauBest*: estiamted optimal *tau* based on MC simulation (true best)
% # *tauId*: estimated *tau* based on experiment data
% # *timeList*: list of equally sized time increments
% # *sigmaOmega*: standard deviation of the white noise
% # *steadyState*: steady state value for the autoregression
% # *vars*: optimization solution

%% Frequently used structure variables
% # *Data*: data table variable
% # *Game*: data table for this game
% # *GroupedData*: data summary table
% # *Player*: data table for this player
% # *Result*: structure for containing results
% # *Session*: data of a current *session* (either *ol* or *sr*)
% # *Summary*: data table for output
% # *Table*: data table for individual players or games

%% Updates history
% * *10/8/2016*: document for manuscript
% * *7/28/2016*: reverse engineer the MSE plot and solve for g(x), the
% simulation results appear to be consistent with experiment results
% * *7/27/2016*: impute day 1 data with x = 2000 - theta
% * *7/24/2016*: a revised meanValue that also computes the AR for mean x
% without square
% * *7/14/2016*: used xPrev^2 and xCurr^2 to fit a second order polynomial,
% the result appears to match the MSE result
% * *7/8/2016*: reconstructed x and u data based on xPrev and uPrev, added
% primary keys to the master data table
% * *7/7/2016*: cleaned up a few scripts, added *script_betaProfile* that
% is similar to our previous attempt in Excel (except that we don't use
% the unsupervised algorithm this time)
```

D.2 Script “meanValue.m”

```
% Script for the time progression of mean squared errors
% Last updated: 10/8/2016

clear; clc;
load all_data_table
import pkg.*

%% Script parameters
batchKey = 'Basic';
nTime = 30;
nPolyOrder = 1; % polyfit order
timeList = linspace(0,240,nTime)';

%% Query
Data = DataTable(strcmp(DataTable.batch, batchKey), :);
gameNameListUnique = unique(Data.gameName);
nGame = numel(gameNameListUnique);
nPlayerTotal = numel(unique(Data.playerName));
sessionList = {'Ol' 'A';'Sr' 'B'};

%% Table
Result.Ol.Mse.data = zeros(nTime, nGame);
Result.Sr = Result.Ol;
Result.Ol.xDigitized = []; % all data digitized into nTime samples
Result.Sr.xDigitized = [];

Summary = table();
gameList = cell(1, nGame);

%% Computation
progress = 0;
```

```
diagonalPoly = zeros(1, nPolyOrder + 1); diagonalPoly(end - 1) = 1;

% game 1 to 5
h11 = figure(11);
clf;
hold on

h12 = figure(12);
clf;
hold on
for i = 1:nGame

    Game = struct();
    Game.name = gameNameListUnique(i);
    Game.Data = Data(strcmp(Data.gameName, Game.name), :);
    playerNameListUnique = unique(Game.Data.playerName);
    nPlayer = numel(playerNameListUnique );
    Game.playerList = cell(1, nPlayer);

    Game.Ol.Se.data = [];
    Game.Ol.Se.mean = [];
    Game.Sr = Game.Ol;

    Game.Ol.nData = height(Game.Data(strcmp(Game.Data.session, 'A'), :));
    Game.Sr.nData = height(Game.Data(strcmp(Game.Data.session, 'B'), :));

    % player 1 to nPlayer
    for j = 1:nPlayer

        progress = progress + 1;
        progressPercent = progress / nPlayerTotal * 100

        Player = struct();
        Player.name = playerNameListUnique(j);
        Player.Data = Game.Data(strcmp(Game.Data.playerName, Player.name), :);

        % session A to B
        for k = 1:2
```

```
        sessionName = sessionList(k,1); % session Ol/Sr
        sessionLetter = sessionList(k,2); % session A/B

        Session = struct();
        Session.Data = Player.Data(strcmp(Player.Data.session, sessionLetter), :);
        Session.size = height(Session.Data);

        if Session.size >= 1

            if k == 1
                figure(11)
                plot(Session.Data.time, Session.Data.x, ':')
                Result.Ol.xDigitized = [Result.Ol.xDigitized, digitize(Session.Data.time, Session.Data.x, timeList)];
            else
                figure(12)
                plot(Session.Data.time, Session.Data.x, ':')
                Result.Sr.xDigitized = [Result.Sr.xDigitized, digitize(Session.Data.time, Session.Data.x, timeList)];
            end

            Session.seList = digitize(Session.Data.time, Session.Data.x .^ 2, timeList);

            Player.(sessionName{1}) = Session;
            Game.(sessionName{1}).Se.data = [Game.(sessionName{1}).Se.data, Session.seList];


        end

    end

    Game.playerList{j} = Player;

end


Game.Ol.Se.mean = mean(Game.Ol.Se.data, 2);
Game.Sr.Se.mean = mean(Game.Sr.Se.data, 2);

Game.Table = table();
Game.Table.gameName = Game.name;
```

```
    Game.Table.nValidPlayer = [size(Game.Ol.Se.data, 2), size(Game.Sr.Se.data, 2)];
    Game.Table.nTime = [Game.Ol.nData, Game.Sr.nData] ./ Game.Table.nValidPlayer;

    % autoregressive fitting
    phat(1, :) = polyfit(Game.Ol.Se.mean(1:end-1), Game.Ol.Se.mean(2:end), nPolyOrder);
    if nPolyOrder > 1
        Game.Table.a2(1) = phat(1, end - 2);
    else
        Game.Table.a2(1) = 0;
    end
    Game.Table.a1(1) = phat(1, end - 1);
    Game.Table.a0(1) = phat(1, end);
    Game.Table.steadyState(1) = polyvalArInf(phat(1, :));
    Game.Table.rsq(1) = rsq(phat(1, :), Game.Ol.Se.mean(1:end-1), Game.Ol.Se.mean(2:end));

    phat(2, :) = polyfit(Game.Sr.Se.mean(1:end-1), Game.Sr.Se.mean(2:end), nPolyOrder);
    if nPolyOrder > 1
        Game.Table.a2(2) = phat(2, end - 2) ;
    else
        Game.Table.a2(2) = 0;
    end
    Game.Table.a1(2) = phat(2, end - 1);
    Game.Table.a0(2) = phat(2, end);
    Game.Table.steadyState(2) = polyvalArInf(phat(2, :));
    Game.Table.rsq(2) = rsq(phat(2, :), Game.Sr.Se.mean(1:end-1), Game.Sr.Se.mean(2:end));

    gameList{i} = Game;

    Summary = [Summary; Game.Table];

    Result.Ol.Mse.data(:, i) = Game.Ol.Se.mean;
    Result.Sr.Mse.data(:, i) = Game.Sr.Se.mean;

end

figure(11);
l11 = errorbar(timeList, mean(Result.Ol.xDigitized, 2), std(Result.Ol.xDigitized, 0, 2) / sqrt(size(Result.Ol.xDigitized, 2)), 'b*-');
figure(12);
l12 = errorbar(timeList, mean(Result.Sr.xDigitized, 2), std(Result.Sr.xDigitized, 0, 2) / sqrt(size(Result.Sr.xDigitized, 2)), 'rx-');
```

```
% legend(l11, 'Average')
% legend(l12, 'Average')

Result.Ol.time = timeList;
Result.Sr.time = timeList;
Result.Ol.Mse.mean = mean(Result.Ol.xDigitized .^ 2, 2);
Result.Sr.Mse.mean = mean(Result.Sr.xDigitized .^ 2, 2);
Result.Ol.Mse.sem = std(Result.Ol.xDigitized .^ 2, 0, 2) / sqrt(size(Result.Ol.xDigitized, 2));
Result.Sr.Mse.sem = std(Result.Sr.xDigitized .^ 2, 0, 2) / sqrt(size(Result.Sr.xDigitized, 2));

Game = struct();
Game.Table = table();
Game.Table.gameName = 'Average';

xq = (0:nTime - 1)';

phat(1, :) = polyfit(Result.Ol.Mse.mean(1:end-1), Result.Ol.Mse.mean(2:end), nPolyOrder);
if nPolyOrder > 1
    Game.Table.a2(1) = phat(1, end - 2);
else
    Game.Table.a2(1) = 0;
end
Game.Table.a1(1) = phat(1, end - 1);
Game.Table.a0(1) = phat(1, end);
Game.Table.steadyState(1) = polyvalArInf(phat(1, :));
Game.Table.rsq(1) = rsq(phat(1, :), Result.Ol.Mse.mean(1:end-1), Result.Ol.Mse.mean(2:end));
vq1 = polyvalAr(phat(1, :), Result.Ol.Mse.mean(1), nTime);
legendText1 = ['MSE^+ = ',...
    num2str(Game.Table.a2(1), 2), 'MSE^2 + ',...
    num2str(Game.Table.a1(1), 2), 'MSE + ',...
    num2str(Game.Table.a0(1), 2),...
    ' (r^2 = ', num2str(Game.Table.rsq(1), 2), ')'];

phat(2, :) = polyfit(Result.Sr.Mse.mean(1:end-1), Result.Sr.Mse.mean(2:end), nPolyOrder);
if nPolyOrder > 1
    Game.Table.a2(2) = phat(2, end - 2) ;
else
    Game.Table.a2(2) = 0;
end
```

```
Game.Table.a1(2) = phat(2, end - 1);
Game.Table.a0(2) = phat(2, end);
Game.Table.steadyState(2) = polyvalArInf(phat(2, :));
Game.Table.rsq(2) = rsq(phat(2, :), Result.Sr.Mse.mean(1:end-1), Result.Sr.Mse.mean(2:end));
vq2 = polyvalAr(phat(2, :), Result.Sr.Mse.mean(1), nTime);
legendText2 = ['MSE^+ = ',...
    num2str(Game.Table.a2(2), 2), 'MSE^2 + ',...
    num2str(Game.Table.a1(2), 2), 'MSE + ',...
    num2str(Game.Table.a0(2), 2),...
    ' (r^2 = ', num2str(Game.Table.rsq(1), 2), ')']

Game.Table.nValidPlayer = mean(Summary.nValidPlayer);
Game.Table.nTime = mean(Summary.nTime);

Summary = [Summary; Game.Table];

%% Output

Summary

figure(11);
xlim([-1 250])
% title(['batchKey = ',batchKey,', session = A, and nTime = ',num2str(nTime)])
xlabel('Time (second)')
ylabel('Decision error')
hold off

figure(12);
xlim([-1 250])
% title(['batchKey = ',batchKey,', session = B, and nTime = ',num2str(nTime)])
xlabel('Time (second)')
ylabel('Decision error')
hold off

% h13 = figure(13);
% clf
% hold on
% errorbar(Result.Ol.time, Result.Ol.Mse.mean, Result.Ol.Mse.sem,'b*')
% plot(Result.Ol.time, vq1, 'b--')
```

```
% errorbar(Result.Sr.time, Result.Sr.Mse.mean, Result.Sr.Mse.sem,'rx')
% plot(Result.Sr.time, vq2, 'r-.')
% legend('Open loop', legendText1, 'Soft regulation', legendText2)
% xlim([-1 250])
% xlabel('Time (second)')
% ylabel('MSE')
% title(['batchKey = ',batchKey,' and nTime = ',num2str(nTime)])
% hold off
%
% h14 = figure(14);
% clf
% hold on
% plot(Result.Ol.time, vq1, 'b--')
% plot(Result.Sr.time, vq2, 'r-.')
% legend(['Open loop: ', legendText1], ['Soft regulation: ', legendText2])
% xlim([-1 250])
% xlabel('Time (second)')
% ylabel('MSE')
% title(['batchKey = ',batchKey,' and nTime = ',num2str(nTime)])
% hold off

mseMatrix.Ol = [Result.Ol.Mse.data, Result.Ol.Mse.mean];
mseMatrix.Sr = [Result.Sr.Mse.data, Result.Sr.Mse.mean];

switch batchKey
    case 'Basic'
        save meanValue_Basic Result nTime Summary mseMatrix
    case 'Noisy'
        save meanValue_Noisy Result nTime Summary mseMatrix
    case 'Small'
        save meanValue_Small Result nTime Summary mseMatrix
end

currDir = cd();

set(h11,'Units','Inches');
pos = get(h11,'Position');
set(h11,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3), pos(4)])
saveas(h11, [currDir '/output/' batchKey 'Fig' num2str(11) '.pdf']);
```

```
set(h12,'Units','Inches');
pos = get(h12,'Position');
set(h12,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3), pos(4)])
saveas(h12, [currDir '/output/' batchKey 'Fig' num2str(12) '.pdf']);

% set(h13,'Units','Inches');
% pos = get(h13,'Position');
% set(h13,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3), pos(4)])
% saveas(h13, [currDir '/output/' batchKey 'Fig' num2str(13) '.pdf']);
%
% set(h14,'Units','Inches');
% pos = get(h14,'Position');
% set(h14,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3), pos(4)])
% saveas(h14, [currDir '/output/' batchKey 'Fig' num2str(14) '.pdf']);
```

D.3 Script "sysid.m"

```
% system Id and control
% last updated: 10/8/16

clear;
clc;
import pkg.*

%% Script parameters
nSample = 5e3;
batchKey = 'Basic';
rng('default');
gFunDof = 1; % gFun degrees of freedom (3: quadratic, 2: linear slope + intercept, 1: linear slope only)
```

```
isSystemIdControl = false;
gameId = 0;


if gameId ~= 0
    isSystemIdControl = true;
end

%% Simulation parameters
switch batchKey
    case 'Basic'
        load meanValue_Basic
    case 'Noisy'
        load meanValue_Noisy
    case 'Small'
        load meanValue_Small
end
vars0 = [0 sqrt(Summary.a1(end, 1)) 0 sqrt(Summary.a0(end, 1))];
beta0 = 0.5;
tau0 = 0;
if gameId ~= 0
    nPlayer = round(mean(Summary(gameId, :).nValidPlayer));
    MseList.ol = mseMatrix.Ol(:, gameId);
    MseList.sr = mseMatrix.Sr(:, gameId);
    gameName = Summary.gameName{gameId};
else
    nPlayer = round(mean(Summary(end, :).nValidPlayer));
    MseList.ol = Result.Ol.Mse.mean;
    MseList.sr = Result.Sr.Mse.mean;
    gameName = Summary.gameName{end};
end
Pars.nTime = nTime;
Pars.nPlayer = nPlayer;
Pars.nSample = nSample;
timeList = linspace(0, 240, nTime)';
opiDistList = 0:500;
stateList = -500:500;
```

```
%% System Id and control
if isSystemIdControl

    Options.MaxIter = 30;
    Options.PlotFcns = {@optimplotx @optimplotfval};

    disp('vars...')
    if gFunDof == 1
        [vars, fval1] = fminsearch(@(x) 1 - pkg.rsqEmpirical(MseList.ol, pkg.sysval([0, x(1), 0, x(2)], 0, 0, Pars)),...
            [vars0(2), vars0(end)],...
            Options)
        vars = [0, vars(1), 0, vars(end)]
    else
        if gFunDof == 2
            [vars, fval1] = fminsearch(@(x) 1 - pkg.rsqEmpirical(MseList.ol, pkg.sysval([0, x], 0, 0, Pars)), vars0(2:end), Options)
            vars = [0, vars]
        else
            [vars, fval1] = fminsearch(@(x) 1 - pkg.rsqEmpirical(MseList.ol, pkg.sysval(x, 0, 0, Pars)), vars0, Options)
        end
    end
    disp('betaId...')
    [betaId, fval2] = fminsearch(@(x) 1 - pkg.rsqEmpirical(MseList.sr, pkg.sysval(vars, x, 0, Pars)), beta0, Options)
    disp('tauId...')
    [tauId, fval3] = fminsearch(@(x)  1 - pkg.rsqEmpirical(MseList.sr, pkg.sysval(vars, nan, x, Pars)), tau0, Options)

    disp('betaRobust...')
    betaRobust = findBetaBest(vars(end) ^ 2 / MseList.ol(1), vars(2), nTime, true)

    disp('betaDyn...')
    [Simulation.Mse.BetaDyn.result, betaDynList, stateMatrix] = sysval(vars, nan, nan, Pars);


    disp('betaBest...')
    [betaBest, fval4] = fminsearch(@(x) sum(pkg.sysval(vars, x, 0, Pars)) / sum(MseList.ol), beta0, Options)
    disp('tauBest...')
    [tauBest, fval5] = fminsearch(@(x) sum(pkg.sysval(vars, nan, x, Pars)) / sum(MseList.ol), tau0, Options)
```

```
    disp('Id and control done!')

    Solution.vars = vars; Solution.betaId = betaId; Solution.tauId = tauId;
    Solution.betaRobust = betaRobust; Solution.betaDynList = betaDynList;
    Solution.betaBest = betaBest; Solution.tauBest = tauBest;

else
    switch batchKey
        case 'Basic'
            load sysid_Basic
        case 'Noisy'
            load sysid_Noisy
        case 'Small'
            load sysid_Small
    end
    vars = Solution.vars; betaId = Solution.betaId; tauId = Solution.tauId;
    betaRobust = Solution.betaRobust; betaDynList = Solution.betaDynList;
    betaBest = Solution.betaBest; tauBest = Solution.tauBest;


    disp('Previous system Id and control results loaded')
end

%% Simulations
disp('MSE...')
gFun = vars(1:3); sigmaOmega = vars(end);
Simulation.Mse.GFun.result = sysval(vars, 0, 0, Pars);
Simulation.Mse.GFun.rsq = rsqEmpirical(MseList.ol, Simulation.Mse.GFun.result);
Simulation.Mse.GFun.legend = ['g(x) = '...
    num2str(gFun(1), 2) 'x^2 + '...
    num2str(gFun(2), 2) 'x + '...
    num2str(gFun(3), 2) ' ('...
    'r^2 = ' num2str(Simulation.Mse.GFun.rsq, 2)...
    ', \sigma_\omega = ' num2str(sigmaOmega, 2)...
    ', \sigma_\omega^2/MSE(0) = ' num2str(sigmaOmega ^ 2 / MseList.ol(1), 2)...
    ')'];

Simulation.Mse.BetaId.result = sysval(vars, betaId, 0, Pars);
Simulation.Mse.BetaId.rsq = rsqEmpirical(MseList.sr, Simulation.Mse.BetaId.result);
```

```
Simulation.Mse.BetaId.ratio = sum(Simulation.Mse.BetaId.result) / sum(Simulation.Mse.GFun.result);
Simulation.Mse.BetaId.legend = ['\beta = ' num2str(betaId, 2)...
    ' (r^2 = ' num2str(Simulation.Mse.BetaId.rsq, 2),...
    ', ' num2str((1 - Simulation.Mse.BetaId.ratio) * 100, 2), '% reduced)'];

Simulation.Mse.TauId.result = sysval(vars, nan, tauId, Pars);
Simulation.Mse.TauId.rsq = rsqEmpirical(MseList.sr, Simulation.Mse.TauId.result);
Simulation.Mse.TauId.ratio = sum(Simulation.Mse.TauId.result) / sum(Simulation.Mse.GFun.result);
Simulation.Mse.TauId.legend = ['\beta(d) = exp(-' num2str(tauId, 2) 'd)'...
    ' (r^2 = ' num2str(Simulation.Mse.TauId.rsq, 2),...
    ', ' num2str((1 - Simulation.Mse.TauId.ratio) * 100, 2), '% reduced)'];

Simulation.Mse.BetaBest.result = sysval(vars, betaBest, 0, Pars);
Simulation.Mse.BetaBest.ratio = sum(Simulation.Mse.BetaBest.result) / sum(Simulation.Mse.GFun.result);
    if Simulation.Mse.BetaBest.ratio > Simulation.Mse.BetaId.ratio
        Simulation.Mse.BetaBest = Simulation.Mse.BetaId;
        betaBest = betaId;
    end
Simulation.Mse.BetaBest.legend = ['\beta^\ast_T = ' num2str(betaBest, 2)...
    ' (' num2str((1 - Simulation.Mse.BetaBest.ratio) * 100, 2), '% reduced)'];

Simulation.Mse.BetaRobust.result = sysval(vars, betaRobust, 0, Pars);
Simulation.Mse.BetaRobust.ratio = sum(Simulation.Mse.BetaRobust.result) / sum(Simulation.Mse.GFun.result);
Simulation.Mse.BetaRobust.legend = ['\beta^\ast_R = ' num2str(betaRobust, 2)...
    ' (' num2str((1 - Simulation.Mse.BetaRobust.ratio) * 100, 2), '% reduced)'];

Simulation.Mse.TauBest.result = sysval(vars, nan, tauBest, Pars);
Simulation.Mse.TauBest.ratio = sum(Simulation.Mse.TauBest.result) / sum(Simulation.Mse.GFun.result);
Simulation.Mse.TauBest.legend = ['\beta^\ast_T(d) = exp(-' num2str(tauBest, 2) 'd)'...
    ' (' num2str((1 - Simulation.Mse.TauBest.ratio) * 100, 2), '% reduced)'];

Simulation.Mse.BetaDyn.ratio = sum(Simulation.Mse.BetaDyn.result) / sum(Simulation.Mse.GFun.result);
Simulation.Mse.BetaDyn.legend = ['Dynamic \beta^\ast_R(t) MSE'...
    ' (' num2str((1 - Simulation.Mse.BetaDyn.ratio) * 100, 2), '% reduced)'];

Simulation.g = polyval(gFun, stateList);

disp('beta...')
opiDistListShort = linspace(min(opiDistList), max(opiDistList), 10);
```

```
Simulation.Beta.BetaId.result = ones(1, numel(opiDistListShort)) * betaId;
Simulation.Beta.BetaId.legend = Simulation.Mse.BetaId.legend;
Simulation.Beta.BetaBest.result = ones(1, numel(opiDistListShort)) * betaBest;
Simulation.Beta.BetaBest.legend = Simulation.Mse.BetaBest.legend;
Simulation.Beta.BetaRobust.result = ones(1, numel(opiDistListShort)) * betaRobust;
Simulation.Beta.BetaRobust.legend = Simulation.Mse.BetaRobust.legend;

Simulation.Beta.TauId.result = exp(0 - tauId * opiDistList);
Simulation.Beta.TauBest.result = exp(0 - tauBest * opiDistList);
Simulation.Beta.TauId.legend = Simulation.Mse.TauId.legend;
Simulation.Beta.TauBest.legend = Simulation.Mse.TauBest.legend;
disp('Simulation done!')

%% Plots
if gameId ~= 0
    titleText = ['batchKey = ' batchKey ', gameName = ' gameName ',...
        nTime = ' num2str(nTime) ', nSample = ' num2str(nSample) ', nPlayer = ' num2str(nPlayer)];
else
    titleText = ['batchKey = ' batchKey ', nTime = ' num2str(nTime) ', nSample = ' num2str(nSample) ', nPlayer = ' num2str(nPlayer)];
end

h31 = figure(31);
clf
hold on
errorbar(Result.Ol.time, MseList.ol, Result.Ol.Mse.sem, 'b*')
plot(timeList, Simulation.Mse.GFun.result, 'b--')
errorbar(Result.Sr.time, MseList.sr, Result.Sr.Mse.sem, 'rx')
plot(timeList, Simulation.Mse.BetaId.result, 'r-.')
% legend('Open loop', Simulation.Mse.GFun.legend,...
%     'Soft regulation',...
%     Simulation.Mse.BetaId.legend);
xlabel('Time (second)')
ylabel('Mean squared error')
xlim([-1, 250])
ylim([0, 0.9e5])
% title(titleText)
hold off
```

```
h32 = figure(32);
clf
hold on
plot(timeList, Simulation.Mse.GFun.result, 'b*--')
[hAx,hLine1,hLine2] = plotyy(timeList, Simulation.Mse.BetaDyn.result, timeList, [betaDynList; betaDynList(end)], 'plot', 'stairs');
ylabel(hAx(1),'Mean squared error') % left y-axis
ylabel(hAx(2),'Social influence') % right y-axis
hAx(1).YColor = 'k';
hAx(2).YColor = 'k';
hLine1.LineStyle = '-.'; hLine1.Color = 'r'; hLine1.Marker = 'x';
hLine2.LineStyle = ':'; hLine2.Color = 'm'; hLine2.LineWidth = 2;
% legend('Open loop MSE', Simulation.Mse.BetaDyn.legend, 'Dynamic \beta^\ast_R(t) input', 'Location', 'northwest');
xlabel('Time (second)')
% title(titleText)
hold off

h33 = figure(33);
clf
hold on
plot(timeList, Simulation.Mse.GFun.result, 'b*--')
plot(timeList, Simulation.Mse.BetaId.result, 'rx-.')
plot(timeList, Simulation.Mse.TauId.result, 'r--')

plot(timeList, Simulation.Mse.BetaRobust.result, 'm:', 'LineWidth', 2)

plot(timeList, Simulation.Mse.BetaBest.result, 'k-.')
plot(timeList, Simulation.Mse.TauBest.result, 'k-')

% legend(...
%     Simulation.Mse.GFun.legend,...
%     Simulation.Mse.BetaId.legend, Simulation.Mse.TauId.legend,...
%     Simulation.Mse.BetaRobust.legend,...
%     Simulation.Mse.BetaBest.legend, Simulation.Mse.TauBest.legend);
xlabel('Time (second)')
ylabel('Mean squared error')
xlim([-1, 125])
ylim([0, 0.3e5])
% title(titleText)
hold off
```

```
%% Output

disp('Output...')

currDir = cd();
if gameId == 0
    dir = [currDir '/output/' batchKey];
else
    dir = [currDir '/output/games/' gameName];
end

set(h31,'Units','Inches');
pos = get(h31,'Position');
set(h31,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3), pos(4)])
saveas(h31, [dir 'Fig31.pdf']);


set(h32,'Units','Inches');
pos = get(h32,'Position');
set(h32,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3), pos(4)])
saveas(h32, [dir 'Fig32.pdf']);

set(h33,'Units','Inches');
pos = get(h33,'Position');
set(h33,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3), pos(4)])
saveas(h33, [dir 'Fig33.pdf']);

% set(h34,'Units','Inches');
% pos = get(h34,'Position');
% set(h34,'PaperPositionMode','Auto','PaperUnits','Inches','PaperSize',[pos(3), pos(4)])
% saveas(h34, [dir 'Fig34.pdf']);

if isSystemIdControl && gameId == 0
    switch batchKey
        case 'Basic'
            save sysid_Basic Pars Simulation Solution
        case 'Noisy'
            save sysid_Noisy Pars Simulation Solution
```

```
        case 'Small'
            save sysid_Small Pars Simulation Solution
    end
    disp('System Id and control results saved')
end

disp('Finished!')
```