

DESIGNING MARKETPLACES AND CIVIC ENGAGEMENT PLATFORMS:
LEARNING, INCENTIVES, AND PRICING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Nikhil Garg
June 2020

© 2020 by Nikhil Garg. All Rights Reserved.

Re-distributed by Stanford University under license with the author.



This work is licensed under a Creative Commons Attribution-3.0 United States License.

<http://creativecommons.org/licenses/by/3.0/us/>

This dissertation is online at: <http://purl.stanford.edu/mf806mq4601>

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Ashish Goel, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Ramesh Johari

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Balaji Prabhakar

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Dorsa Sadigh

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Many of our most crucial societal interactions are now mediated by algorithmic systems. We buy goods, find work and hire each other, discuss current events, and make public decisions through online platforms. Non-profit and government actors further use such systems to assign kids to schools, organs to patients, and food to food banks. Principled socio-technical system design requires formalizing an objective and understanding the incentives, behavioral tendencies, and capabilities of participants; in turn, the design influences participant behavior. In practice, design decisions are made jointly through the interplay of experimental and data-driven techniques on one hand, and theoretical modeling and insights on the other. In this dissertation, I describe work designing socio-technical systems in two domains, two-sided marketplaces and civic engagement platforms. I demonstrate how to leverage theory-motivated design and robust empirical analyses together to build better systems, filling in gaps at the interfaces of these approaches.

Part I considers the design of surge pricing that is incentive compatible for drivers in ride-hailing platforms. The work compares two potential driver payment policies for such platforms: a new driver surge mechanism (now deployed across the US), in which the surged component of a trip payment is additive (independent of trip length) as opposed to multiplicative (proportional to trip length), the historical standard. The paper presents the theoretical foundation that informed this change at Uber. We model surge evolution as a continuous-time Markov chain; we show that, with multiplicative pricing schemes, strategically rejecting certain trip requests may maximize an individual driver's earnings, to the detriment of others. We then develop an incentive compatible pricing scheme with an approximately affine, closed-form expression. Finally, we analyze counter-factual earnings from more than 500,000 ride-hailing trips, validating that our proposal would increase incentive compatibility and earnings stability in practice.

Part II tackles rating system inflation on online platforms, studying how to choose the multiple choice question asked of raters. Each potential question induces a joint distribution between the seller's underlying quality and the ratings they receive, and we develop a large deviations based framework to quantify how quickly the true ranking of sellers is recovered, given this joint distribution. With an experiment on a large online labor market, we show that various rating questions yield dramatically different rating distributions: while 80.6% of freelancers receive the best rating using

a traditional numeric scale, less than 35.8% receive top ratings using other designs. Our theoretical framework quantifies the resulting information gain and provides a principled way to choose among the scales given the behavioral data. We further show how informational priorities (identifying the absolute best items versus separating unacceptable from acceptable items) should affect design.

Part III considers the design and building of systems for participatory budgeting. A key challenge in such systems is to design the elicitation mechanism: participants must be able to share their opinions in a manner that is simple, expressive enough for decisions that lie in high-dimensional spaces, and yet enables provably efficient aggregation. First, I present a new method for people to collectively make a decision on a societal budget. In our method, voters are sequentially asked for their ideal budget within a constraint set determined by the previous voter’s answer. This process simulates stochastic gradient descent, and the asymptotic output provably maximizes societal welfare in certain settings. We test our method by building a intuitive user interface and running elections on Amazon Mechanical Turk. Second, we show how to optimize an existing elicitation mechanism – K Approval, in which each voter identifies their favorite K candidates – in a principled manner. With real voter data from over thirty elections, we demonstrate that many multi-candidate elections that select W winners are run sub-optimally: whereas voters are typically asked to identify their $K = W$ favorite candidates (e.g., $K = 1$ in a winner-takes-all election), it is learning rate optimal to ask voters to identify their favorite $M > K$ candidates.

Acknowledgments

To my advisors, Ashish Goel and Ramesh Johari. You've been the best advisors I could have imagined, as kind and patient as you are excellent scholars and teachers. You've taught me to be unafraid to tackle the problems I believe are important, and have let me shape the PhD that I wanted, even when that meant disappearing for months at a time working on some completely disparate project. I hope to be an advisor in your mold. Ashish, you've modeled taking risks, whether by becoming a pilot or by pivoting research fields and trying to fix democracy. Ramesh, you've modeled picking problems from across fields, serving the academic community, and taking care in all your teaching.

To my other research mentors. Hamid Nazerzadeh has been an intern mentor, co-author, job market advocate, and friend these last two years. Vijay Kamble took me under his wing when I was a first-year student, and has been a great friend and mentor since then, always ready to give advice and offer help when I need it.

To the other professors in the Stanford Society and Algorithms (SOAL) group in MS&E, my adopted community at Stanford. Itai Ashlagi wandered over to our desk area with something interesting to say, on a daily basis. Sharad Goel shared discerning takes on how to do useful research and manages a computing cluster that I (ab)used for each of my works. Irene Lo and Johan Ugander provided crucial advice at several stages of my PhD and were extremely generous with their time and feedback. Outside of MS&E, Michael Bernstein, Dan Jurafsky, Gabriel Weintraub, and James Zou provided guidance on research and beyond, and Dan and James were wonderful collaborators. My committee members Balaji Prabhakar, Daniela Saban, and Dorsa Sadigh also have been patient, and have provided a fun and useful dissertation process.

To my group-mates, lab-mates, SOAL student community, and fellow group meeting attendees and TAs: Nick Arnosti, Tum Chaturapruek, William Cai, Yiling Chen, Je-ok Choi, Lin Fan, Lodewijk Gelauff, Margalit Glasgow, Arpit Goel, Reyna Hulett, Vijay Kamble, Allison Koencke, Anilesh Krishnaswamy, Sanath Kumar, Hannah Li, Wanyi Li, Bar Light, Peter Lofgren, Rahul Makhijani, Nikki Nikolenko, Allison Park, Benjamin Plaut, Stephen Ragain, Geoff Ramseyer, Mohammad Rasouli, Carlos Riquelme, Sukolsak Sakshuwong, Sven Schmit, Virag Shah, Camelia Simoiu, Meltem Tutar, David Walsh, Carrie Wu, Linjia Wu, and Hongyang Zhang, and others I'm

sorry to have messed up in omitting here. I learned more from my peers than any other group, and y'all made it fun, even when you (Hannah, Sven) made fun of innocent me for no good reason.

To the rest of my Stanford friends, including Leighton Barnes, Daniel Hatchell, Andrew Naber, Nitish Padmanaban, Arjun Seshadri, Raghav Subramaniam, Sahaana Suri, Paroma Varma, Steve Yadlowsky, and friends throughout Electrical Engineering, the university, and AddisCoder. Our potlucks, board game nights, and endless chats made it more than bearable. These last few months in particular, when we've all been stuck inside in Covid-19 quarantine but connected virtually, have especially underscored the value of great friendships.

To my friends from grade-school through undergrad, too many to list here. I've realized recently how lucky I am to have a group of friends together from our time in elementary school, a group that has only grown over time in both size and love. Y'all have been wonderful, and our trips and long-distance video calls have meant more to me than I can express.

To all my co-authors: William Cai, Dorottya Demszky, Johann Gaebler, Lodewijk Gelauff, Matthew Gentzkow, Ashish Goel, Sharad Goel, Ramesh Johari, Dan Jurafsky, Vijay Kamble, David Marn, Kamesh Munagala, Hamid Nazerzadeh, Benjamin Plaut, Sukolsak Sakshuwong, Londa Schiebinger, Jesse Shapiro, Rob Voigt, James Zou. Research is best when it is collaborative.

To administrators and staff, and those at my steady rotation of coffee-shops: Roz Morf, Lori Cottle, Adam Bailey, Jenny Lam, David. Thank you for your support and for making the Stanford community what it is.

The specific works in this dissertation also benefited from work by and many conversations with Alice Lu, Carter Mundell, Cary Luu, Hayden Brown, Jake Edison, John Horton, Kane Sweeney, Leighton Barnes, Margaret Tian, Michael Bernstein, Michael Sheldon, Oliver Hinder, Peter Cohen, Qitang Wang, and Shane Kinder.

Academic research is a privilege afforded to us by a public that trusts that we are using their resources wisely, for a greater good. In that spirit, I'd like to thank the National Science Foundation, Office of Naval Research, Army Research Office, and Stanford Cyber Initiative for supporting me and my work.

Finally, to my family: Mom, Dad, and Nikhita, and all my family in India and the U.S.: you've always been there, providing invaluable love and support.

Contents

Abstract	v
Acknowledgments	vii
1 Introduction	1
1.1 Dissertation outline	4
1.2 Bibliographic notes	7
I Pricing in Online Marketplaces	9
2 Driver Surge Pricing	10
2.1 Introduction	10
2.1.1 Contributions	11
2.1.2 Related Work	12
2.2 Model, driver earnings, and platform objective	14
2.2.1 Model primitives	14
2.2.2 Driver strategies and earnings	15
2.2.3 Platform objective and constraints	18
2.2.4 Practical considerations	19
2.3 Incentive compatibility with affine pricing	19
2.3.1 Single-state model: multiplicative pricing is incentive compatible	20
2.3.2 Dynamic model: multiplicative pricing is not incentive compatible	21
2.3.3 Why is multiplicative surge pricing not incentive compatible?	22
2.4 Incentive Compatible Surge Pricing	23
2.4.1 Transition probabilities and expected time spent in each state	23
2.4.2 Incentive Compatible pricing in the dynamic model	24
2.4.3 Opportunity cost intuition for incentive compatible pricing	25
2.4.4 Proof sketch of Theorem 2.4.1	27
2.5 Numerics: Incentive Compatibility with Additive Surge	27

2.5.1	Computing optimal driver policies	28
2.5.2	Results	28
2.6	Empirical Comparison of Surge Mechanisms	30
2.6.1	Data setting and analysis description	31
2.6.2	Analysis and results: value of short and long trips	33
2.7	Conclusion	34
II	Designing Rating Systems in Online Marketplaces	36
3	Designing Informative Rating Systems	37
3.1	Introduction	37
3.2	Related literature	40
3.2.1	Platform measures to counter or encourage inflation	40
3.2.2	Survey design and rating inflation in other contexts	41
3.2.3	Theoretical analyses of ratings	42
3.3	Online labor market experiment description	42
3.3.1	Motivation and hypothesis	43
3.3.2	Empirical context	43
3.3.3	Method	44
3.4	Labor market test results	46
3.4.1	Verbal rating scales counter inflation	47
3.4.2	Verbal rating scales yield more informative ratings	49
3.4.3	Discussion	52
3.5	A framework to compare rating scales	52
3.5.1	Model	53
3.5.2	Quantifying design performance via convergence rate	56
3.5.3	Application to the online labor market	57
3.6	Conclusion and discussion	59
3.6.1	Challenges, opportunities, and limitations	59
3.6.2	Future work	61
4	Designing Optimal Binary Rating Systems	63
4.1	Introduction	63
4.2	Related work	65
4.3	Model and optimization	66
4.3.1	Model and problem specification	66
4.3.2	Large deviations & discretization	68
4.3.3	Solving the optimization problem	69

4.3.4	Visualization and discussion	71
4.4	Designing approximately optimal, implementable rating systems	72
4.5	Mechanical Turk experiment	73
III Designing Voting Mechanisms on Civic Engagement Platforms		76
5	Iterative Local Voting	77
5.1	Introduction	77
5.1.1	Contributions	79
5.2	Related Work	81
5.2.1	Stochastic Subgradient Method	81
5.2.2	Iterative Local Voting	81
5.2.3	Optimization without Gradients	82
5.2.4	Participatory Budgeting	83
5.2.5	Implicit Utilitarian Voting	83
5.3	Convergence Analysis	84
5.3.1	Spatial Utilities	84
5.3.2	Decomposable Utilities	86
5.3.3	Equivalence to Directional Equilibrium	87
5.4	Experiments with Budgets	88
5.4.1	Experimental Setup	89
5.4.2	Experimental Parameters	89
5.4.3	User Experience	90
5.5	Results and Analysis	91
5.5.1	Convergence	91
5.5.2	Understanding Voter Behavior	96
5.6	Conclusion	100
6	Optimizing Elections with Many Candidates	103
6.1	Introduction	103
6.2	Related work	104
6.3	Model	106
6.3.1	Model primitives	106
6.3.2	Asymptotic design invariance	107
6.4	Learning Rates and Optimal Design	109
6.4.1	Learning rates	109
6.4.2	Optimal design and discussion	111
6.5	Theoretical Design Insights	112

6.5.1	When does randomization help?	112
6.5.2	K -Approval for selecting W winners	114
6.6	Empirics and PB deployments	115
6.6.1	Data description	117
6.6.2	Model validation	117
6.6.3	K -Approval for selecting W winners	118
6.6.4	Randomization in practice	119
6.7	Discussion	119
A	Driver Surge Pricing	121
A.1	Additional discussion and information	121
A.1.1	Platform objective	121
A.1.2	Driver earnings in each state	123
A.1.3	Model's relationship to practice	124
A.1.4	Supplementary Figures	125
A.2	Extra empirical information	126
A.2.1	Additional results and facts	127
A.2.2	Empirical analysis additional information	132
A.3	Proofs of single state model results	136
A.3.1	Driver reward	136
A.3.2	Proof of Theorem 2.3.1	136
A.3.3	Proof of Proposition 2.3.1	139
A.3.4	Uniqueness of optimal policy for single-state model	140
A.4	Proofs of dynamic model results	141
A.4.1	Driver reward	142
A.4.2	Proof strategy for incentive compatible pricing and structural results	145
A.4.3	Necessary lemmas	146
A.4.4	Proofs of main results, Theorems 2.3.2 and 2.4.1	149
A.4.5	Optimal policies as depend on derivatives	154
A.4.6	Proofs of appendix-only lemmas	162
B	Designing Informative Rating Systems	169
B.1	Further analysis of the labor market test	169
B.1.1	Verifying randomization in allocation of clients	169
B.1.2	Robustness against high volume clients and allocation bug	170
B.1.3	Regressing treatment response with treatment cell and other covariates	171
B.1.4	More on inflation over time	172
B.1.5	Analysis of cell with randomized order of answer choices	173

B.1.6	Design approach using labor market data	173
B.2	Amazon Mechanical Turk synthetic experiment	174
B.2.1	Experiment description	175
B.2.2	Results	177
B.3	Proofs	179
C	Designing Optimal Binary Rating Systems	182
C.1	Mechanical Turk experiment, simulations, and results	182
C.1.1	Experiment description	182
C.1.2	Calculating optimal β and H	184
C.1.3	Simulation description	185
C.2	Supplementary theoretical information and results	189
C.2.1	Formal specification of system state update	189
C.2.2	Detailed algorithm	189
C.2.3	Formalization of effect of matching rates shifting	191
C.2.4	Limit of β as $M \rightarrow \infty$	191
C.2.5	Learning $\psi(\theta, y)$ through experiments	192
C.3	Proofs	193
C.3.1	Rate functions for $P_k(\theta_1, \theta_2)$	194
C.3.2	Laplace's principle with sequence of rate functions	195
C.3.3	Rate function for W_k	196
C.3.4	Proofs of Lemma 4.3.1 and Theorem 4.3.1	198
C.3.5	Additional necessary lemmas	201
C.3.6	Proof for Theorem 4.3.2	206
C.3.7	Proof of Theorem C.2.1	208
C.3.8	Kendall's tau and Spearman's rho related proofs	209
D	Iterative Local Voting	212
D.1	Mechanical Turk Experiment Additional Information	212
D.2	Indifference Regions Additional Information	213
D.3	Proofs	217
D.3.1	Known SSGM Results	217
D.3.2	Mapping ILV to SSGM	218
D.3.3	Proof of Theorem 5.3.1	218
D.3.4	Proof of Theorem 5.3.2	220
D.3.5	Proof of Propositions	220
D.3.6	Proof of Theorem 5.3.3	221
D.3.7	Proofs of Lemmas	222

E	Optimizing Elections with Many Candidates	228
E.1	Empirics additional information	228
E.2	Proofs	234
E.2.1	Asymptotic design-invariance	234
E.2.2	Learning rates	235
E.2.3	Design insights	239
	Bibliography	251

List of Tables

3.1	Treatments groups for labor market test	45
3.2	Number of clients and jobs in each cell, and mean treatment response	46
5.1	Summary of convergence results	84
B.1	Average treatment responses under different data policies	170
B.2	OLS Regression Results with covariate for previous number of treatment responses .	172
B.3	Optimal scores ϕ for each treatment, where the score of the top position is normalized to 5.	175
B.4	Large deviation learning rates for each treatment in the Mturk experiment, calculated using Equation (3.4) and the joint distributions generated using the training data plotted in Figure B.6. Optimal for each treatment corresponds to the highest learning rate among many random score functions tested.	178
E.1	List of election data that we use in Section 6.6. From PrefLib, we use all elections where full rankings are available and there are at least 5 candidates and 700 voters. Throughout, we ignore voters who did not submit full rankings (especially with high K -Ranking requested, this might only be a fraction of the total number of actual votes). Additionally, for the PB elections, we limit the data to those who submitted votes online rather than through paper ballots. Sources for the PrefLib datasets are: Mattei and Walsh (2013); O’Neill (2013); Popov et al. (2014); Regenwetter et al. (2007, 2008).	230

E.2	OLS Regression on the best K to use in K -Approval, by the number of candidates and desired winners. Standard errors are cluster standard errors, where each cluster is an election in our dataset.	232
E.3	Elections and goals where randomizing between two K -Approval mechanisms produces leads to faster learning than using either of the mechanisms separately. For several of these cases, randomization also beats the Approval rate optimal mechanism.	233
E.4	Where the constants such that $0 < \epsilon < a < \frac{T_2}{2} < T_2 < T_1 < T_1 + 2T_2 + a = 1$, i.e., the table describes a valid probability distribution. Row 10 is as follows: The first K candidates (the asymptotic winners) occupy the first K spots, in an order drawn uniformly at random. Similarly, The bottom $M - K$ candidates occupy the bottom $M - K$ spots, in an order drawn uniformly at random. This randomization ensures asymptotically design invariance.	247

List of Figures

2.1	Driver surge heatmaps with multiplicative and additive surge. On Uber, drivers see a heatmap of surge when they are logged in but not on a trip, guiding them to higher earning opportunities by signaling each location’s value (Lu et al., 2018). Structural simplicity is essential to clearly communicate payments to drivers, and additive and multiplicative surge represent the two simplest options.	11
2.2	The primitives are as follows: $\lambda_1 = \lambda_2 = 12, \lambda_{1 \rightarrow 2} = 1, \lambda_{2 \rightarrow 1} = 4$; in both states, trip lengths are distributed according to a Weibull distribution with shape 2 and mean $\frac{1}{3}$. These parameters reflect realistic average trip to wait time values, and that surge tends to be short-lived compared to non-surge times.	25
2.3	How C , the ratio R_1/R_2 at which IC pricing is feasible from Theorem 2.4.1, changes (1) with respect to the mean trip length, and (2) with respect to $\lambda_{i \rightarrow j}$. Except for those that are varied in each plot, the primitives are fixed to those used in Figure 2.2: $\lambda_1 = \lambda_2 = 12, \lambda_{1 \rightarrow 2} = 1, \lambda_{2 \rightarrow 1} = 4$ and, in both states, trip lengths are distributed according to a Weibull distribution with shape 2 and mean $\frac{1}{3}$	26

2.4	Incentive compatibility for each type of surge. The shaded regions are where the respective scheme is incentive compatible in the surge state ($\sigma_2 = (0, \infty)$ is optimal). When not varied, $\lambda_1 = \lambda_2 = 10, \lambda_{1 \rightarrow 2} = 1, \lambda_{2 \rightarrow 1} = 4, R_2 = 3.33, R_1 = 1$, and trip lengths in both states are distributed according to a Weibull distribution with shape 2 and mean 0.3. We assume every trip is accepted in the non-surge state.	29
2.5	Difference in earnings over the next 90 minutes for the driver of a given accepted trip request, and a matched driver who also was open nearby at the time of the request, conditional on surge factor (rounded to nearest 0.5) and length of trip. Error bars are 95% bootstrapped confidence intervals.	34
3.1	Labor market test marginal rating distributions. Error bars are 95% boot-strapped confidence intervals, where the bootstrapped sampling is done at the client level. . .	47
3.2	Likelihood that a client will rehire a freelancer during the time period of the test, given just the first rating the client gives that freelancer during the test period. Values are normalized by the overall mean rehire rate. Confidence intervals are 95% intervals with bootstrapped sampling done at the client level.	50
3.3	Joint distributions of freelancer quality vs. ratings in the <i>Average</i> and <i>Numeric</i> treatment cells, respectively. Low, Medium, and High quality sellers refer to those with other cell average ratings in $[0, 2), [2.5, 3.5)$ and $[4.5, 5]$, respectively. The Y axis is the probability that a freelancer of a given quality receives a rating at least as high as the X axis. Confidence intervals are 95% intervals with bootstrapped sampling done at the client level.	51
3.4	We apply and test our design approach using experimental data from our online labor market. Large deviation rates are calculated using Equation (3.4) and the joint distributions generated in Section 3.4.2. Optimal for each treatment corresponds to the highest learning rate among many random scores.	57
4.1	Optimal β (with $M = 200$) with various objective weight functions w and matching rates g	68
4.2	Experiment and simulation results	74
5.1	UI Screenshot for 1 set of the \mathcal{L}^2 Mechanism	90
5.2	Solution over time for each mechanism type	92
5.2	(Continued) Solution over time for each mechanism type	93
5.3	Net normalized movement in window of $N = 30$	94
5.3	(Continued) Net normalized movement in window of $N = 30$	95
5.4	Average movement in dimension over total movement for each voter, with dimensions sorted	97

5.5	Fraction of possible movement in each dimension in \mathcal{L}^∞ , conditioned on distance to ideal pt. The ‘All’ condition contains data from all three \mathcal{L}^∞ instances, whereas the others only from the instance that also did full elicitation.	98
5.6	Median time per page	99
5.7	Histogram of values from all full elicitation data. The red vertical lines indicate each slider’s default value (at the 2016 estimated budget).	101
5.8	Histogram of weights from all full elicitation data. The red vertical lines indicate the sliders’ default value of 5.	101
6.1	K -Approval rate optimal mechanism for the Mallows model as ϕ , number of candidates, and number of winners vary.	112
6.2	Validating model: comparing learning rates to empirical error, and showing approximate design invariance.	116
A.1	Fraction of time spent in surge state, $\mu_2(\sigma)$, with driver policy $\sigma = \{\sigma_1 = (0, \infty), \sigma_2\}$, where $\sigma_2 = (t, \infty)$, i.e., t is the minimum trip length accepted in the surge state. The primitives are as follows: $\lambda_1 = \lambda_2 = 12, \lambda_{1 \rightarrow 2} = 1, \lambda_{2 \rightarrow 1} = 4$; in both states, trip lengths are distributed according to a Weibull distribution with shape 2 and mean $\frac{1}{3}$. These parameters reflect realistic average trip to wait time values, and that surge tends to be short-lived compared to non-surge times. Note that the driver can increase the time spent in the surge state by rejecting short surge trips.	126
A.2	Using the same model primitives as in Figure A.1: the payment function $w_i(\tau)$ for various surge mechanisms plotted two ways, when $R_2 = 1$ and $R_1 = \frac{2}{3}$ for drivers who accept every trip.	126
A.3	Surge facts from RideAustin marketplace	128
A.4	Basic trip facts from RideAustin marketplace	129
A.5	Same as Figure 2.5, except with the surge factor flipped to simulate a world with frequent, valuable surge.	130
A.6	Histogram of per-shift driver earnings per hour. Note that the y-axis is in log scale.	131
A.8	Constructed payment function (Additive surge with base fare) vs the reverse engineered Status quo fare payments at the trip level. As expected, additive surge tends to pay higher for shorter trips and lower for longer trips.	133
A.9	Using next nearby driver with an accepted trip as the counter-factual match.	134
A.10	Using period length of next 1 hour (instead of 1.5 hours).	134
A.11	Starting measurement from dispatch time instead of trip start time, i.e., taking into account the first part of the trip that is unpaid for the driver.	135
A.12	With pure multiplicative and additive surge, respectively (no min fare).	135

B.1	Rating distributions for different client sampling techniques. As in the main text, the confidence intervals are 95% bootstrapped confidence intervals, with bootstrapped sampling at the client level.	171
B.2	Mean ratings for each treatment cell by the number of previous treatment responses given during the test period. Error bands are bootstrapped 95% confidence intervals.	173
B.3	Joint distributions of freelancer quality vs. ratings in the other treatment cells. Low, Medium, and High quality sellers refer to those with other cell average ratings in $[0, 2)$, $[2.5, 3.5)$ and $[4.5, 5]$, respectively.	174
B.4	Joint distributions, where Low, Medium, and High quality sellers refer to those with other cell average ratings in $[0, 2)$, $[2, 4)$ and $[4, 5]$, respectively.	175
B.5	Simulated performance over time with various other configurations. The “Worst” scoring rule corresponds to the rule ϕ with the smallest learning rate found for each treatment.	176
B.6	Joint distributions of rating and expert score on the MTurk training set, by treatment condition.	177
B.7	Simulated performance of each rating scale with Equally Spaced scores.	178
C.1	Additional information for MTurk experiment	183
C.2	Paragraph rating distribution – for paragraph θ and rating word y , the empirical $\hat{\psi}(\theta, y)$ is shown. Colors encode the true quality as rated by experts (light blue is best quality, dark blue is worst).	184
C.3	Optimal $H(y)$ varying by $w(\theta_1, \theta_2)$ using Mechanical Turk data	185
C.4	Simulations from data from Mechanical Turk experiment – Binary rating system	188
D.1	Page 1 – Welcome Page for all mechanisms	213
D.2	\mathcal{L}^2 Page 2 – Instructions	213
D.3	\mathcal{L}^2 Page 3 – Mechanism	214
D.4	Page 4 – Feedback for all mechanisms	214
D.5	Full Elicitation Page 3 – Mechanism	215
D.6	Fraction of possible movement in each dimension in \mathcal{L}^∞ , conditioned on distance to ideal pt. The ‘All’ condition contains data from all three \mathcal{L}^∞ instances, whereas the others only from the instance that also did full elicitation. This plot only includes those people who provided an explanation as long or longer than the median explanation provided (197 characters).	215

D.7	Fraction of possible movement in each dimension in \mathcal{L}^∞ , conditioned on distance to ideal pt. The ‘All’ condition contains data from all three \mathcal{L}^∞ instances, whereas the others only from the instance that also did full elicitation. This plot only includes those people who provided an explanation shorter than the median explanation provided (197 characters).	216
E.1	Average bootstrapped error (fraction of winning subset not identified) by the number of voters, compared to the errors implied by the (empirically calculated) learning rates. All mechanisms plotted have the same asymptotic winners.	229
E.2	More approximate design invariance plots	231
E.3	K -Approval rate optimal mechanism for the Mallows model as ϕ , number of candidates, and number of winners vary. This plot contains an empirical line, which is calculated using the coefficients in the regression contained in Table E.2.	232

Chapter 1

Introduction

Many of our most crucial societal interactions are now mediated by algorithmic systems. We buy goods, find work and hire each other, discuss current events, and make public decisions through online platforms. Non-profit and government actors further use such systems to assign kids to schools (Abdulkadiroğlu et al., 2005), organs to patients (Roth et al., 2004), and food to food banks (Prendergast, 2016). The promise of these socio-technical systems is that they enable coordination at scale. Each participant can act locally according to their own incentives, information, and constraints – and make global connections and impact. When designed correctly, the system helps people to together achieve some shared goal, and ensures that the benefits are divided fairly; meanwhile, bad designs waste resources and privilege some participants to the detriment of others.

Building these systems is challenging. The designer must decide who can participate, what participants can do, and how they communicate to each other and the platform – while respecting business, legal, and human constraints. Each decision affects the participant incentives, information, and constraints, and hence their behavior and ultimately system outcomes. The system’s specific context determines good design. For example, commodity markets such as ride-hailing centrally set prices, while ones with more heterogeneous products like lodging allow hosts to set their own; these decisions reflect the relative amount of information available to participants and the platform.

Many disciplines now consider the design of such systems, using a range of methods. On the day to day level, practitioners at large companies run thousands of experiments (A/B tests) a day, to evaluate everything from user interfaces and site design to algorithms for matching, recommendation, and pricing (Luca and Bazerman, 2020; Salganik, 2019). Whether a particular change is deployed depends on how it performs on a suite of metrics. Experimental and data-driven methods are best when theory and intuition is not precise enough to choose among similar designs. For example, Google tested 40 shades of blue for web links (Arthur, 2012).

However, data-driven methods are insufficient to design modern socio-technical systems. It is often too expensive or simply infeasible to experimentally consider too many options, and running

principled experiments in networked settings is difficult. Some changes may be too sensitive or risky to deploy, even in a limited environment. Such difficulties are especially compounded for changes that are public-facing or have effects that may not be apparent on the time scale of an experiment. For example, in ride-hailing markets, large changes to how drivers are paid should be paired with public communication, naturally limiting how often such changes can be made.

Theoretical mechanism and market design models shed insight in such cases; under assumptions on participant utility functions, they seek to predict behavior under large design changes. Such approaches are especially successful at constructing system designs that provably are optimal under some objective (such as welfare), with attractive properties (such as strategy-proofness). In this manner, different policies can be compared by connecting individual actions under the model to a global objective. Academic papers shed high-level insight on practical problems, and researchers apply their training to design mechanisms outside academia, across domains (Athey and Luca, 2019).

However, as the aphorism, “all models are wrong but some are useful” (Box, 1979) suggests, design optimality guarantees are useful to the extent that the underlying model approximates reality. Unfortunately, real-world behavior often differs substantially from that assumed by theoretical models, and mechanisms in practice often face subtle business, legal, and human constraints that are difficult to model. Such factors make it difficult to predict a mechanism’s practical impact *a priori*, before it is deployed or evaluated in a particular context.

These competing aspects of data-driven and theoretical approaches lead to a division of labor to design and build socio-technical systems across domains; the latter provides high-level guidance and ideas, and the former evaluates them and optimizes them in context. However, there are gaps at the interface of these approaches – where platforms must make fine-grained design choices that are infeasible to wholly evaluate experimentally, but for which coarse theoretical insights are insufficient:

- Optimal mechanisms may not be implementable in practice, for context-specific reasons. How do we test assumptions and analyze good-enough approximations for a given application?
- User interface and other design decisions typically made experimentally may have important implications for downstream decisions and platform objectives. How do we connect short-lived experimental measures to these long-run effects?

Motivated by such gaps, in this thesis I demonstrate several ways forward to connect the practically feasible to the theoretically optimal, in service of building useful systems and solving central socio-technical design challenges. One recurring lesson is that this approach requires focusing on a particular application and developing requisite domain knowledge; through collaborations with practitioners, my work has informed systems at Uber, a large online labor platform, and in participatory budgeting elections across the United States.

(Approach 1) Empirically designing implementable mechanisms that approximate ideal ones. Mechanism design solutions are often theoretically elegant but may violate real-world

constraints. For example, they may be too complicated to be understood by regular people, and so participants may not trust the system or act strategically (sub-optimally) even when the mechanism is provably “strategy-proof” (Hassidim et al., 2016). A platform designer must then implement a mechanism that obeys its constraints and best approximates the first-best solution, even when analyzing such mechanisms is theoretically intractable. One way forward is to first construct an optimal solution, and then empirically and numerically compare different approximations of it.

For example, ride-hailing platforms centrally set prices so as to reduce delays and matching frictions. Prior academic work considers how to set prices such that drivers drive when and where is most efficient, yielding important, influential insights (Besbes et al., 2018a; Bimpikis et al., 2016; Ma et al., 2018). In practice, however, driver payments must be transparent and communicated ahead of time – such as through a heat-map of current surge prices.

In Chapter 2, we start with a theoretical market design approach and design a general, incentive compatible payment scheme for drivers in the presence of dynamic demand. We then compare two simple, transparent pricing schemes used in practice, revealing under which market characteristics each scheme approximates the ideal one. Leveraging real ride-hailing data, we make a specific recommendation for such markets. The work grew out of my summer at Uber; I was a data science intern on the team building a new driver surge mechanism (now deployed across the US), in which the surged component of a trip payment is *additive* (independent of trip length) as opposed to *multiplicative* (proportional to trip length), the historical standard. The chapter presents the theoretical foundation that informed this change.

(Approach 2) Deriving principled outcome measurements for experiments. Some design challenges, especially concerning platform user interfaces, both (a) are so contingent on idiosyncratic human behavior that an experiment is the only path forward and (b) affect long-run platform objectives in ways that may not be transparent in a short experiment. In such cases, one approach is to construct an intermediate outcome that is both measurable in a short experiment and is theoretically sound in connecting with a long-run platform objective.

This approach is especially important for information elicitation systems, which are used to gather opinions from users; opinions are then aggregated and used to train down-stream models and make decisions. On online marketplaces and labor markets, for example, feedback ratings are used by the platform to recommend products and by future participants to vet purchases. On civic engagement platforms, votes (for candidates or issues) are used to make public decisions. In such systems, the specific question asked of respondents (for example, “How did this freelancer compare to others you’ve hired” versus, “Please rate the seller from 1 to 5 stars”) substantially affects the distribution of responses (as we demonstrate in this thesis), which can be measured in an experiment. However, absent an understanding of how the platform will use the collected data in downstream tasks, it is unclear which distribution of responses is “best.” A question that is effective at weeding out unacceptable products may not be useful in identifying top performers.

To address this challenge in various contexts, Chapters 3, 4, and 6 each start with theoretical models of downstream platform decisions and how they’re differentially affected by the distribution of responses from the user interface. This approach yields a statistic mapping such distributions to a long-run platform learning objective. Then, we use both experiments and historical data to compare specific user interface designs with respect to this statistic. Using this approach, these chapters identify effective opinion elicitation questions for a large online labor platform and for participatory budgeting elections.

(Approach 3) Building systems to evaluate usability and test assumptions. Finally, in some cases, a proposed mechanism simply needs to be tested, either in the lab or in a real-world setting. While the mechanism may “seem” simple and have desirable theoretical properties, the best proof of its practical usability is a working demonstration. Such end-to-end design and analysis of a new mechanism – deriving it with theory, building a usable system that implements it, and testing it through controlled experiments – is the promise of a truly interdisciplinary approach.

In this spirit, in Chapter 5 we propose a new voting method to vote in multi-dimensional continuous spaces – like budgets. We theoretically show that different variants of the mechanism work (find a societal optimal budget) under varying assumptions on analytic forms of the voters’ utility functions. We then implement the system and evaluate it experimentally on Amazon Mechanical Turk, showing that the required interface is indeed usable and that several of the mechanism variants – especially the ones that theoretically are the most flexible – work, i.e., consistently converge to the same outcome across tests.

This thesis presents approaches to overcome gaps at the interface between technical disciplines to design socio-technical systems. To close this part of the introduction, it is essential to acknowledge the importance of engaging with the social sciences and humanities in this venture; such an interdisciplinary direction is among the most promising future directions in the design of socio-technical systems, and ignoring those disciplines worsens outcomes, cf. Abebe et al. (2020); Hitzig (2018); Immorlica (2019). I have benefited from such collaborations, though the resulting work is not included in this thesis (Demszky et al., 2019; Fishkin et al., 2019; Garg et al., 2018b). I hope that this dissertation nevertheless demonstrates that such gaps between disciplines can be bridged without sacrificing the strengths of either, and that the approaches above serve as a guide to future work, in order to design better, more principled socio-technical systems.

1.1 Dissertation outline

This dissertation is organized as follows. Appendices containing proofs and other supporting material are all included at the end.

Part I, “Pricing in Online Marketplaces” This part is composed of a single chapter (Chapter 2), “Driver Surge Pricing,” which presents a surge pricing scheme for drivers in ride-hailing platforms. The work grew out of my summer at Uber; I was a data science intern on the team building a new driver surge mechanism (now deployed across the US), in which the surged component of a trip payment is *additive* (independent of trip length) as opposed to *multiplicative* (proportional to trip length), the historical standard.

The chapter presents the theoretical foundation that informed this change. Due to the temporal dynamics of surge – in which certain time periods are more valuable than other periods, to balance the supply to available drivers with the demand for rides – trips of different lengths have different driver opportunity costs. We model surge evolution as a continuous-time Markov chain; in our model, we show that, with traditional, multiplicative pricing schemes, strategically rejecting certain trip requests may maximize an individual driver’s earnings, to the detriment of riders and other drivers. For example, it may be advantageous to reject short trips during surge in the hopes of getting a longer surge trip. We then develop an incentive compatible pricing scheme with an approximately affine, closed-form expression. Such simplicity is important in practice to enable transparency and communication of surge prices to drivers through a heat-map, and stands in contrast to previous works which resolve such strategic concerns through prices that emerge from a global optimization framework. Finally, through both calibrated simulations and by analyzing counter-factual earnings from more than 500,000 ride-hailing trips, we validate that our proposal would increase incentive compatibility and driver earnings stability in practice.

Part II, “Designing Rating Systems in Online Marketplaces” In this part, we tackle the challenge of rating inflation in online platforms, where most sellers predominantly receive positive ratings (on AirBnB, e.g., almost 95% of hosts have an average rating of at least 4.5 out of 5 stars (Zervas et al., 2015)). Such inflation leads to uninformative rating systems in which noise dominates.

In Chapter 3, we study how the platform can choose the multiple choice *question* that it asks raters. Each potential question induces a joint distribution between the seller’s true underlying quality and the ratings they receive. For example, asking “How did this freelancer compare to others you’ve hired” versus, “Please rate the seller from 1 to 5 stars,” may yield different rating responses, for the same quality seller. We develop a large deviations based framework that quantifies how quickly the platform recovers the true ranking of sellers, given this joint distribution. This framework thus provides the ideal metric for an A/B test, connecting a design to a platform’s long-term goals, *without* needing to run a long experiment to measure the outcome directly.

We then run an experiment on a large online labor market and show that platforms can get informative ratings, by leveraging *positive-skewed, verbal* label scales (e.g., one scale ranges from *Below Average* to *Best Freelancer I’ve Hired*). In the experiment, clients rate freelancers through various verbal and numeric scales. These scales induce substantially different rating behaviors: while 80.6% of freelancers receive the best numeric rating, less than 35.8% receive top verbal ratings;

furthermore, clients are up to 31.8% more likely to rehire the freelancer after giving them a top rating on a verbal scale than after giving them the top numeric score. Our theoretical framework then quantifies the resulting information gain and provides a principled way to choose among the scales given the behavioral data. These results serve as a positive contrast to a long line of work proposing various changes that do not prevent inflation.

In Chapter 4, we show how a platform’s informational priorities should affect the rating system design. In commodity markets such as ride-hailing, it is essential to separate unacceptable from acceptable participants as quickly as possible. In superstar markets, on the other hand, fine differentiation among the best sellers is most important. We formalize such goals as weighted versions of Kendall τ distance between the estimated and true participant rankings. Then, in a setting in which rater responses are binary, we develop an efficient non-convex optimization algorithm to find the optimal joint distribution, i.e., the relationship between the participant’s quality and the probability at which they should receive a positive rating. This joint distribution maximizes the asymptotic weighted accuracy and the large deviation rate at which it is reached. Our algorithm exploits a dimensionality reduction in which only ranking mistakes between similar participants can dominate the large deviation rate at which the error decays. It finds, for example, that it is optimal for most participants to receive primarily positive ratings when the goal is to identify unacceptable ones.

Part III, “Designing Voting Mechanisms on Civic Engagement Platforms” Platforms can support far more than the exchange of individual goods and services. Civic engagement platforms enable people to collectively make complex, public decisions in an axiomatically fair way, applying the social choice tradition of Kenneth Arrow in a computational age. In participatory budgeting (PB), for example, people vote on how to allocate millions of dollars across many candidate projects.

A key challenge in such systems is to design the *elicitation mechanism*: participants must be able to share their opinions in a manner that is simple, expressive enough for decisions that lie in high-dimensional spaces, and yet enables provably efficient aggregation. The design space ranges from voters identifying their favorite project to negotiating with other voters directly. In this part, I develop such mechanisms in a theory-driven way, and then test them either through synthetic experiments or deployments, such as real municipal PB elections run on our group’s platform.¹ Such interdisciplinary work is essential to building the practical, large-scale, decentralized group decision-making systems that will be central to the next generation of online platforms and marketplaces.

In Chapter 5, we develop a new method for people to collectively make a decision on a societal budget. In our method, voters are sequentially asked for their ideal budget within a constraint set determined by the previous voter’s answer. This process simulates stochastic gradient descent, and the asymptotic output provably maximizes societal welfare in certain settings. In particular, if each voter’s dis-utility for a budget is its ℓ_p distance from their ideal budget, then asking each voter for

¹pbstanford.org. The platform has been used in over fifty elections, allocating tens of millions of dollars with tens of thousands of voters. I was not involved in initial platform development, but contribute to its continued use.

their favorite budget in a local ℓ_q dual ball provides a stochastic gradient for the societal welfare function.² Then, sequentially querying voters in this manner simulates stochastic gradient descent. We tested our method by running elections on Amazon Mechanical Turk,³ demonstrating that (a) one can build an intuitive user interface, and (b) the procedure converges to a consistent point across several runs, with a small number of voters. Overall, the work demonstrates that theory can be used to build new mechanisms that make complex collective decisions by asking voters simple questions.

In Chapter 6, we show how to optimize an existing elicitation mechanism – K Approval, in which each voter identifies their favorite K candidates – in a principled manner. We extend the approach of the work in Part II, showing how the value of K (e.g., eliciting 3 candidates versus 4) determines the large deviation rate at which the asymptotic outcome is learned, even when it does not change the outcome. Then, with real voter data from over thirty elections (including from our PB platform), we demonstrate that many multi-candidate elections that select W winners are run sub-optimally: whereas voters are typically asked to identify their $K = W$ favorite candidates (e.g., $K = 1$ in a winner-takes-all election), it is learning rate optimal to ask voters to identify their favorite $M > K$ candidates. This small change matters: in one election, asking for each voter’s two favorite candidates versus single favorite would have been the difference between identifying the ultimate winner with a 99.9% vs 80% probability after 400 voters. This rule-of-thumb has influenced our recommendations for the elections run by our partner cities, demonstrating the impact of combining theory with data analysis.

1.2 Bibliographic notes

Chapter 2 Joint with Hamid Nazerzadeh. To appear as an extended abstract at the ACM Conference on Economics and Computation (EC), in 2020 (Garg and Nazerzadeh, 2020). A journal version is in submission (Garg and Nazerzadeh, 2019).

Chapter 3 Joint with Ramesh Johari. To appear as an extended abstract at the ACM Conference on Economics and Computation (EC), in 2020 (Garg and Johari, 2020). A journal version is in submission (Garg and Johari, 2019b).

Chapter 4 Joint with Ramesh Johari. Published at the International Conference on Artificial Intelligence and Statistics (AISTATS), in 2019 (Garg and Johari, 2019a).

Chapter 5 Joint with Vijay Kamble, Ashish Goel, David Marn, and Kamesh Munagala. Journal version published at the Journal of Artificial Intelligence Research (JAIR), in 2019 (Garg et al., 2019b). Originally published in conference version form at the International Conference on World Wide Web (WWW), in 2017 (Garg et al., 2017): copyright 2017 International World Wide Web Conference Committee, published under Creative Commons CC By 4.0 License.

²For $p \in \{1, 2, \infty\}$. All $p, q \in [1, \infty) \cup \infty$ s.t. $1/p + 1/q = 1$ work for a more restricted voter behavior model.

³A demo of the system I built is available at <http://54.183.140.235/radius/50/mechanism/12/>.

Chapter 6 Joint with Lodewijk Gelauff, Sukolsak Sakshuwong, and Ashish Goel. Published at the AAAI Conference on Human Computation and Crowdsourcing (HCOMP), in 2019 (Garg et al., 2019a).

For each work, I contributed substantially to all aspects of the research and writing process.

Part I

Pricing in Online Marketplaces

Chapter 2

Driver Surge Pricing

2.1 Introduction

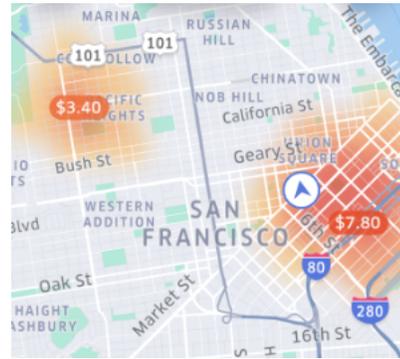
Ride-hailing marketplaces like Uber, Lyft, and Didi match millions of riders and drivers every day. A key component of these marketplaces is a *surge* (dynamic) pricing mechanism. On the rider side of the market, surge pricing reduces the demand to match the level of available drivers and maintains the reliability of the marketplace, cf., (Hall et al., 2015), and so allocates the rides to the riders with the highest valuations. On the driver side, surge encourages drivers to drive during certain hours and locations, as drivers earn more during surge (Chen and Sheldon, 2016; Hall et al., 2017; Lu et al., 2018). Castillo et al. (2017) show that surge balances both sides of this spatial market by moderating the demand and the density of available drivers, hence avoiding so called “Wild Goose Chase” equilibria in which drivers spend much of their time on long distance pick ups. Surge pricing – along with centralized matching technologies – is often considered the primary reason that ride-hailing marketplaces outperform traditional taxi services on metrics such as driver utilization and overall welfare (Ata et al., 2019; Buchholz, 2017; Cramer and Krueger, 2016).

However, variable pricing (across space and time) must be carefully designed, since it can create incentives for “cherry-picking” and rejecting certain trip requests. Such behavior increases earnings of strategic drivers at the expense of other drivers, who may then disproportionately receive such trip

This chapter is joint with Hamid Nazerzadeh. Published as an extended abstract at the ACM Conference on Economics and Computation (EC), in 2020 (Garg and Nazerzadeh, 2020). A journal version is in submission (Garg and Nazerzadeh, 2019). We would like to thank Uber’s driver pricing data science team, in particular Carter Mundell, Jake Edison, Alice Lu, Michael Sheldon, Margaret Tian, Qitang Wang, Peter Cohen, and Kane Sweeney for their support and suggestions without which this work would have not been possible. We also thank Leighton Barnes, Ashish Goel, Ramesh Johari, Vijay Kamble, Hannah Li, and Virag Shah. This work was funded in part by the Stanford Cyber Initiative, the Office of Naval Research grant N00014-15-1-2786, and National Science Foundation grants 1544548 and 1839229.



(a) *Multiplicative surge heatmap.* “1.6x” on the map means that the standard fares for trips from the corresponding area are increased by 60%.



(b) *Additive surge heatmap.* “\$7.8” on the map means that \$7.8 is added to each trip’s standard fare from the corresponding area.

Figure 2.1: Driver surge heatmaps with multiplicative and additive surge. On Uber, drivers see a heatmap of surge when they are logged in but not on a trip, guiding them to higher earning opportunities by signaling each location’s value (Lu et al., 2018). Structural simplicity is essential to clearly communicate payments to drivers, and additive and multiplicative surge represent the two simplest options.

requests after they are rejected by others, cf., (Cook et al., 2018). It also reduces overall platform reliability, inconveniencing riders who may have to wait longer before receiving a ride.

Uber recently revamped its driver surge mechanism, in an attempt to improve the driver experience and make earnings more dependable (Uber, 2019b). The main change is making surge “additive” instead of “multiplicative.” Under **multiplicative surge**, the driver payout from a surged trip scales with the length of the trip. In contrast, under **additive surge**, the surge component of the payout is constant (independent of trip length), with some adjustment for very long trips (Uber, 2019c). (Figure 2.1 depicts the heat-map of surge on the driver app for each type of surge.) We show that this change directly addresses the issue that drivers who strategically reject trip requests may earn more than drivers who do not, even as total payments remain the same.

2.1.1 Contributions

We consider the design of *incentive compatible* (IC) pricing mechanisms in the presence of surge. Trips differ by their length $\tau \in \mathbb{R}_+$, and the platform sets the payout $w(\tau)$ for each trip in each world state (i.e., surge vs non-surge). Drivers decide which trip requests $\sigma \subseteq \mathbb{R}_+$ to accept in each world state, in response to the payout function w .¹ The technical challenge is to design an IC pricing mechanism w , for which accepting all trips is an earning maximizing strategy for drivers over a long horizon, i.e., where $\sigma = (0, \infty)$ in each world state maximizes driver earnings.

¹Drivers’ level of sophistication and experience varies, cf. Cook et al. (2018). An IC mechanism aligns the incentives of drivers to accept all trips, for any level of strategic response to pricing strategies.

We first study a continuous-time, infinite horizon *single-state model*, where trip requests arrive over time according to a stationary Poisson process. We show that in this model, multiplicative pricing – where the payout of a trip is proportional to the length of that trip – is incentive compatible. To obtain this result, we show in Theorem 2.3.1 that the best response strategy of a driver to function w , to maximize earnings, is a threshold strategy where the driver accepts all trips with payout rate $\frac{w(\tau)}{\tau}$ above some threshold. Hence, a mechanism that equalizes the payout rate of all trips is incentive compatible.

We then present a model where the world state stochastically transitions over time between surge and non-surge states, with trip payments, distributions, and intensity varying between states. In such a *dynamic* system, completing a given trip affects a driver’s earnings beyond just the length of the trip, i.e., imposes a future-time externality on the driver that is a function of the trip length. The driver’s trip opportunity cost thus includes both what occurs during a trip, and a continuation value. This externality causes *multiplicative pricing to not be incentive compatible in the presence of surge* (Theorem 2.3.2), in contrast to the single-state model. Namely, drivers can benefit from rejecting long trips in a non-surge state, and short trips in the surge state.

In Theorem 2.4.1, our main result, we propose a class of incentive compatible pricing functions described in closed form of the model primitives. The prices incorporate driver temporal externalities: during surge, short trips pay more per unit time than do long trips.

Next, we study surge pricing in our model numerically, showing that additive surge is incentive compatible in more regimes of interest than is multiplicative surge. Finally, using RideAustin ride-hailing data, we show that our theoretical insights extend to practice: additive surge correctly values trips amid temporal externalities, unlike multiplicative surge.

To our knowledge, ours is the first ride-hailing pricing work to incorporate dynamic (non-constant), stochastic demand and pricing. This component is essential to uncover how a particular trip imposes substantial temporal externalities on a driver’s future earnings.

2.1.2 Related Work

We discussed some of the related work on surge pricing above. Here, we briefly review the lines of research closest to ours. We refer the reader to a recent survey by Korolko et al. (2018) for a broader overview of the growing literature on ride-hailing markets.

Driver spatio-temporal strategic behavior. Several works model strategic driver behavior in a spatial network structure, and across time in a single-state. Ma et al. (2018) develop spatially and temporally smooth prices that are welfare-optimal and incentive compatible in a deterministic model. Their prices form a competitive equilibrium and are the output of a linear program with integer solutions. We similarly seek to develop incentive compatible pricing schemes, and both works broadly construct VCG-like prices that account for driver opportunity costs. Our focus is on structural aspects (e.g., multiplicative in trip length) in a non-deterministic model.

Bimpikis et al. (2016) show how the platform would price trips between locations, taking into account strategic driver re-location decisions, in a stationary model with discrete locations. They show that pricing trips based on the origin location substantially improves surplus, as well as the benefits of “balanced” demand patterns. Besbes et al. (2018b) consider a continuous state space setting and show how a platform may optimally set prices across the space in reaction to a localized demand shock to encourage drivers to relocate; their model has driver cost to re-locate, but no explicit time dimension. They find that localized prices have a global impact, and, e.g., the optimal pricing solution incentivizes some drivers to move away from a demand shock. Afèche et al. (2018) consider a two state model with demand imbalances and compare platform levers such as limiting ride requests and directing drivers to relocate, in a two-state fluid model with strategic drivers. They upper-bound performance under these policies, and find that it may be optimal for the platform to reject rider demand even in over-supplied areas, to encourage driver movement. A similar insight is developed by Guda and Subramanian (2019) who explicitly model market response to surge pricing. Finally, Yang et al. (2018) analyze a mean-field system in which agents compete for a location-dependent, time-varying resource, and decide when to leave a given location. They leverage structural results—agents’ equilibrium strategies depend just on the current resource level and number of agents—to numerically study driver relocation decisions as a function of the platform commission structure.

Pricing in ride-sharing and service systems. There is a growing literature on queuing and service systems motivated in part by the ride-sharing market. For example, Besbes et al. (2018a) revisit the classic square root safety staffing rule in spatial settings, cf., Bertsimas and van Ryzin (1991, 1993). Much of the focus of this line of work is how pricing affects the arrival rate of (potentially heterogeneous) customers, and thus the trade-off between the price and rate of customers served in maximizing revenue.

Banerjee et al. (2015) consider a network of queues in which long-lived drivers enter the system based on their expected earnings but cannot reject specific trip requests. Under their model, dynamic pricing *cannot* outperform the optimal static policy in terms of throughput and revenue, but is more robust. Cachon et al. (2017) argue in contrast that surge pricing and payments are welfare increasing for all market participants when drivers decide when to work. Similar in spirit to our work, Chen and Hu (2018) consider a marketplace with forward-looking buyers and sellers who arrive sequentially and can wait for better prices in the future. They develop strategy-proof prices whose variation over time matches the participants’ expected utility loss incurred by waiting. Lei and Jasin (2016) consider a model where customers arrive over time and utilize a capacity constrained resource for a certain amount of time. They develop an asymptotically revenue-maximizing, dynamic, customer-side pricing policy, even when service times may be heterogeneous. Glazer and Hassin (1983) consider taxi-driver strategic responses to multiplicative and affine pricing, as we do, focusing on deviations in which a driver can take a circuitous route in order to increase the length of a trip.

One of the most related to our work in modeling approach, Kamble (2019) studies how a freelancer can maximize long-term earnings with job-length-specific prices, balancing on-job payments and utilization time. In his model, a freelancer sets their own prices for a discrete number of jobs of different lengths and, with assumptions similar to our single-state model, it is optimal for the freelancer to set the same price per hour for all jobs. We further discuss the relationship of this work to our single-state model below.

Organization. The rest of the paper is organized as follows. Section 2.2 contains our model; we further derive driver earnings as it depends on their strategy, and formalize the platform objective. In Section 2.3, we formulate a driver’s best response strategy to affine pricing functions in each model. In Section 2.4, we present incentive compatible pricing functions for our surge model. In Section 2.5, we numerically compare the IC properties of additive and multiplicative surge. Finally, in Section 2.6, we empirically compare additive and multiplicative surge using data from the RideAustin marketplace.

2.2 Model, driver earnings, and platform objective

We consider a large ride-hailing market with decoupled pricing, from the perspective of a *single driver*. This driver receives trip requests of various lengths, whose rate, distribution, and payment are known to the driver and determined exogeneously to decisions to accept or decline requests. We do not consider spatial heterogeneity in our setting, to focus on the temporal opportunity cost and continuation value based on a length of the trip.²

In this section, we first in Section 2.2.1 present the primitives of our two models, a single-state model and a dynamic model with surge pricing. Then in Section 2.2.2 we describe the driver’s strategy space and derive the driver reward in each model. Next, in Section 2.2.3, we formalize the platform objective and technical challenge solved in this work. We conclude with a short discussion on our model’s relationship to practice in Section 2.2.4.

2.2.1 Model primitives

We start with the model primitives in each model.

Single-state model

We start with a model where there is a single world state, i.e., all model components are constant over time. Time is continuous and indexed by t . At each time t , the driver is either *open*, or *busy*.

²We believe our insight can be extended to a spatial setting where the price can be decomposed to a time-based component, based on the length of the trip, and a spatial component based on the destination of the trip. However, this would be beyond the scope of this work, cf., Bimpikis et al. (2016).

While open, the driver receives job (trip) requests from riders according to a Poisson process at rate λ , i.e., the time between requests is exponential with mean $\frac{1}{\lambda}$. Job lengths, denoted by τ , are drawn independently and identically from a continuous distribution F .

If the driver accepts a job request of length τ at time t (as discussed below), they receive a payout of $w(\tau)$ at time $t + \tau$, at which time they become open again. Otherwise, the driver remains open. Except where specified, the only assumption on w is that it is asymptotically (sub)-linear: $\exists c : \liminf_{\tau \rightarrow \infty} \frac{w(\tau)}{\tau} \leq c$, which ensures that the driver reward is also bounded.

Dynamic model with surge pricing

A model with fixed pricing and arrival rates of jobs is not a realistic representation of ride-hailing platforms. In particular, rider demand (both in intensity and in distribution) may vary substantially over time, even within a day (cf. Appendix Figure A.4d). To study how this **dynamic** nature affects driver decisions, we consider a model with two states, $i \in \{1, 2\}$, where $i = 2$ denotes the *surge* state. (At a high level, the surge state provides a higher earnings rate to the driver. The precise definition is in Section 2.2.2, after we formulate the driver’s earnings rate in each state).

The world evolves stochastically between the two states, as a Continuous Time Markov Chain (CTMC). When the world is in state i , the state changes to j according to a fixed exponential clock that ticks at rate $\lambda_{i \rightarrow j}$, independently of other randomness.

When open in state i , the driver receives job requests at rate λ_i with lengths $\tau \sim F_i$, and collects payout according to payment function w_i , which is presumed to have the same properties as w in the single-state model. The state of the world may change while a driver is on trip. Crucially, the driver receives payments according to the state of the world i when the trip *begins*. We will use $w = \{w_1, w_2\}$ to denote the overall pricing mechanism.

2.2.2 Driver strategies and earnings

In our model, the driver can decide whether to accept the trip request, with no penalty.³

In the single-state model, let $\sigma \subseteq \mathbb{R}_+ \triangleq (0, \infty)$ denote the driver’s (fixed) strategy, where $\tau \in \sigma$ implies that a driver accepts job requests of length τ . In the dynamic model, the driver follows policy $\sigma = \{\sigma_1, \sigma_2\}$, where $\sigma_i \subseteq \mathbb{R}_+$ indicates the jobs accepted in state i . We assume that driver policies are measurable with respect to F (corresponding F_i in dynamic model); for technical reasons, in the dynamic model we also assume that σ_i consist of a union of open intervals, i.e., are open subsets of \mathbb{R}_+ . When we write equalities with policies σ , we mean equality up to changes of measure 0.

The driver is long-lived and aims to maximize their own lifetime average hourly earnings on the platform, including both open and busy times. Let $R(w, \sigma, t)$ denote the (random) total earnings from jobs accepted from time 0 up to time t if the driver follows policy σ and the payout function

³This assumption follows Uber’s current practice. We further discuss the driver’s information set in Section 2.2.4.

is w . Then, the driver's lifetime *earnings rate* is

$$R(w, \sigma) \triangleq \liminf_{t \rightarrow \infty} \frac{R(w, \sigma, t)}{t}.$$

This earnings rate is a deterministic (non-random) quantity, and is a function of the driver policy σ , pricing function w , and the primitives.

A driver policy σ^* is **optimal** (best-response) with respect to pricing function w if it maximizes the lifetime earnings rate of the driver among all policies: $R(w, \sigma^*) \geq R(w, \sigma)$, for all valid policies σ (i.e., measurable with respect to F or F_i , with σ_i open sets). Then, pricing function w is **incentive compatible (IC)** if accepting all job requests is optimal with respect to w , i.e., $\sigma = (0, \infty)$ in the single-state model or $\sigma = \{(0, \infty), (0, \infty)\}$ in the dynamic model is optimal with respect to w . In other words, payment function w is incentive compatible if an earnings-maximizing driver (who knows all the primitives, w , and the trip length τ at request time) accepts every trip request.

We now analyze the driver's lifetime earnings rate $R(w, \sigma)$ for each model.

Driver earnings in the single state model

In the single-state model, the primitives directly induce a renewal reward process, where a given renewal cycle is the time a driver is newly open to the time they are open again after completing a job. Let $W(\sigma)$ be the mean earnings on trips $\tau \in \sigma$, i.e., the expected earning in a renewal cycle; let $T(\sigma)$ be the sum of the expected wait time to an accepted trip and the expected length of a trip, and thus the expected renewal cycle length; let $F(\sigma)$ be the probability the driver receives a request in σ . Then, the lifetime driver mean hourly earnings (earnings rate) is

$$R(w, \sigma) = \frac{W(\sigma)}{T(\sigma)} = \frac{\frac{1}{F(\sigma)} \int_{\tau \in \sigma} w(\tau) dF(\tau)}{\frac{1}{F(\sigma)\lambda} + \frac{1}{F(\sigma)} \int_{\tau \in \sigma} \tau dF(\tau)}$$

The first equality follows from the renewal reward theorem, and holds with probability 1.

Driver earnings in the dynamic model

For the dynamic model, on the other hand, we cannot directly use the renewal reward theorem with a renewal cycle containing just a single trip. The driver's earning on a given trip is no longer independent of earnings on other trips: given a job that starts in the surge state, the driver's next job is more likely to also start in surge. Given whether each job started in the surge state, however, job earnings are independent. We can use this property to prove our next lemma, which gives the driver earnings rate in the dynamic model. Let $\mu_i(\sigma)$ be the fraction of time the driver spends either open state i or on a trip that *starts* in state i .

Lemma 2.2.1. *In the dynamic model, the earnings rate can be decomposed into each state i earnings*

rate $R_i(w_i, \sigma_i)$ and fraction of time $\mu_i(\sigma)$ spent in state i :

$$R(w, \sigma) = \mu_1(\sigma)R_1(w_1, \sigma_1) + \mu_2(\sigma)R_2(w_2, \sigma_2) \quad \text{with probability 1.}$$

As in the single-state model, $R_i(w_i, \sigma_i) = \frac{W_i(\sigma_i)}{T_i(\sigma_i)}$, where

$$W_i(\sigma_i) = \frac{1}{F_i(\sigma_i)} \int_{\tau \in \sigma_i} w_i(\tau) dF_i(\tau), \quad T_i(\sigma_i) = \frac{1}{\lambda_i F_i(\sigma_i)} + \frac{1}{F_i(\sigma_i)} \int_{\tau \in \sigma_i} \tau dF_i(\tau)$$

We prove the result by defining a new renewal process, in which a single reward renewal cycle is: the time between the driver is open in state 1 to the next time the driver is open in state 1 *after* being open in state 2 at least once. In other words, *each* renewal cycle is composed of some number (potentially zero) of sub-cycles in which the driver is open in state 1 and then is open in state 1 again after a completed trip; one sub-cycle starting with the driver open in state 1 and ending with being open in state 2 (either after a completed trip or a state transition while open); some number (potentially zero) of sub-cycles in which the driver is open in state 2 and then is open in state 2 again after a completed trip; and finally one sub-cycle starting in state 2 and ending with the driver open in state 1.

Given the number of such renewal reward cycles completed up to time t , the total earnings on trips starting in each state (earnings in each sub-cycle) are independent of each other, and then we use Wald's identity (Wald, 1973) to separate $\mu_i(\sigma)$ and $R_i(\sigma_i)$.

Note that $T_i(\sigma_i)$ is not exactly the expected length of time in a single sub-cycle in a state given σ_i , but rather is proportional to it; the multiplicative constant $\frac{1}{\lambda_i F_i(\sigma_i) + \lambda_{i \rightarrow j}}$ cancels out with the same constant in the expected earnings in a single sub-cycle in a state given σ_i . This constant emerges from the primitives: when the driver is open in state i , there are two competing exponential clocks (with rates $\lambda_i F_i(\sigma_i)$ and $\lambda_{i \rightarrow j}$, respectively) that determine whether the driver will accept a request before the world state changes.

What does $\mu_i(\sigma)$ look like? We defer showing the exact form to Section 2.4.1 in advance of developing incentive compatible pricing. Here, we provide some intuition: the trips that a driver accepts in each state determines the portion of their time spent on trips started in each state. If a driver never accepts trips in the non-surge state, they will be open and thus available for a trip as soon as surge begins. Inversely, if a driver accepts a long surge trip immediately before surge ends, they will be paid according to the surge payment function w_2 even though surge has ended. Surprisingly, given the complex formulation of the reward $R(w, \sigma)$ as it depends on $\sigma = \{\sigma_1, \sigma_2\}$, we find the structure of optimal policies as they depend on the pricing w_i , as well as incentive compatible pricing functions.

Finally, we can now precisely define what it means for $i = 2$ to be the surge state: it has a higher potential earning rate than state 1; $\exists \sigma_2$ such that $R_2(w_2, \sigma_2) > R_1(w_1, \sigma_1), \forall \sigma_1 \subseteq \mathbb{R}_+$.⁴

⁴This assumption is different than the statement $w_2(\tau) \geq w_1(\tau), \forall \tau$, and neither implies the other; it is a condition

2.2.3 Platform objective and constraints

Having derived the driver reward, we now describe the platform objective, setting up the technical challenge we solve in the rest of the work. Recall that our model is **decoupled**: rider and driver prices are determined separately. Under decoupled pricing, the platform has under its control both the price $p_i(\tau)$ charged to the rider and the payment $w_i(\tau)$ paid to the driver for a trip of length τ —and the two values *are not necessarily related at the trip level*. This modeling assumption follows the current practice (Uber, 2019e) and allows us to focus on the drivers’ perspective, without further complicating the analysis.⁵

What should be the role of driver payments with decoupled pricing? In practice, the platform quotes the rider a price and ‘guarantees’ fulfillment if a ride is requested; driver payments should thus primarily ensure that all requested rides are fulfilled, motivating our goal of designing incentive compatible prices. In Appendix Section A.1.1, we formalize this intuition by considering driver payments w as a sub-problem of the comprehensive platform challenge, involving jointly setting both rider prices and driver payments to maximize an objective (e.g., profit or welfare). We establish that – with decoupled pricing and an earnings-maximizing driver within our model – this joint problem can be decomposed into one in which the rider pricing (not considered in this work) determines the objective value, subject to finding a driver payment policy w that satisfies incentive compatibility and a driver *participation constraint*: that the driver earnings rate is higher than an outside option earnings rate (denoted R), i.e., $\max_{\sigma} R(w, \sigma) \geq R$.

In the dynamic model, we additionally consider *per-state driver earnings constraints*, $R_i(w_i, (0, \infty)) = R_i$, for some exogenous $R_2 > R_1$. This constraint comes from practice, via features not directly captured in our model. As detailed in Appendix Section A.1.2, currently platforms impose a business constraint to pass on rider revenue in each world state to the driver, i.e., the constraints R_i are determined by per-state revenue, a function of latent demand and rider prices. If the platform has more flexibility, R_i may also be optimized, for example to induce drivers to position themselves in areas with more frequent surges, cf. Asadpour et al. (2019); Besbes et al. (2018b); Lu et al. (2018); i.e., revenue during one spatio-temporal period may be used to subsidize driver payments in another period. We do not directly consider how the platform should set R_i (or R), as doing so depends heavily on rider pricing and resulting demand. However, our results do establish a range of R_i for which incentive compatible prices can be constructed.

This decomposition is how decoupled surge pricing is set in practice, and for the rest of this work we seek a payment policy that satisfies these conditions.

jointly on λ_i, F_i, w_i , (but not $\lambda_{i \rightarrow j}$), indicating that a driver can, following some policy, earn more while in the surge state than possible following any policy in the non-surge state.

⁵Coupled pricing imposes more constraints. Bai et al. (2018) and Bikhchandani (2020) both find that the platform should adjust its payout ratio with demand—an example of decoupling—to maximize profit or overall welfare.

2.2.4 Practical considerations

Our model is stylized in several important respects, and ride-hailing practice is not consistent across marketplaces, time, or geography. Our theoretical model reflects our view on the most relevant components from practice.

Driver heat-maps and affine pricing We are especially interested in *affine* pricing schemes, where $w_i(\tau) = m_i\tau + a_i$, with $m_i \geq 0$ (in the single-state model: $w(\tau) = m\tau + a$, with $m \geq 0$; we refer to the case with $a_i > 0$ ($a_i < 0$) as positive (negative) affine pricing). Such pricing functions can be communicated as time and distance rates (see, e.g., Uber (2019d)), and the surge component displayed on a heat-map. This simplicity is an important desiderata from practice, where payments should be clear to drivers.

Driver information structure: trip time and time to the rider. We assume that the platform reveals the total trip length to the driver at the time of request, and that the driver can freely reject it without penalty. Drivers often cannot see the rider’s destination or the trip length until they pick up the rider (but they can reject a request based on the pick-up time to the rider, without penalty).⁶ Some drivers call ahead to find out the rider’s destination or even cancel the trip at the pick-up location, creating negative experiences for both the rider and the driver.⁷ Our notion of incentive compatibility is *ex-post*, implying that drivers would accept all trips, even if the trip length is not revealed, and so this setting from practice is covered as well. Furthermore, in practice, jobs have two components: the time it takes to pick up the rider, and the time while the rider is in the driver’s vehicle – and the former component is typically unpaid.⁸ Our model combines these two components into an overall trip length, which determines payments.

Markovian surge and model limitations. In practice, surge has strong intra-day patterns – for example, rush hours have higher average surge values, cf. Appendix Figure A.3b. However, evolution of surge on finer time scales, on the level of drivers’ individual trip decisions, is more volatile and believably Markovian, cf. Appendix Figure A.3c. Our theoretical model assumes that surge is Markovian and binary and the response of a single driver, and further ignores spatial effects. We discuss such issues in Sections A.1.3 and A.2.1, and our empirical analysis in Section 2.6 provides evidence that our insights extend to practice despite these theoretical limitations.

2.3 Incentive compatibility with affine pricing

In this section, we study the incentive compatibility of affine pricing. In Section 2.3.1, we first characterize the driver’s best-response strategy with respect to any pricing function w in the single-state model. We then observe that multiplicative pricing, a special case of affine pricing where

⁶This practice is not consistent across marketplaces and locations. For example, in California as of January 2020, Uber shows the driver the destination and payment estimate at request time.

⁷We note that destination discrimination is against Uber’s guidelines and could lead to deactivation (Uber, 2019a).

⁸Lyft has recently experimented with paying drivers for the time it takes to pick up the rider (Auerbach, 2019).

$w(\tau) = m\tau$, is incentive compatible. In contrast, in Section 2.3.2, we show that in the dynamic model, multiplicative pricing may no longer be incentive compatible. We further derive the structure of optimal driver policies in each state with respect to affine or multiplicative pricing, which will enable numerical study of the incentive compatibility properties of additive and multiplicative surge in Section 2.5. Section 2.3.3 discusses the key differences in the two models, setting up Section 2.4 where we derive incentive compatible pricing functions for the dynamic model.

2.3.1 Single-state model: multiplicative pricing is incentive compatible

Our first result is a simple optimal driver policy in the single-state model.

Theorem 2.3.1. *With a single state, for each w there exists a constant $c_w \in \mathbb{R}_+$ such that the policy $\sigma^* = \left\{ \tau : \frac{w(\tau)}{\tau} \geq c_w \right\}$ is optimal for the driver with respect to w .*

Theorem 2.3.1 establishes that, in a single-state model with Poisson job arrivals, the *length* of the job is not important, only the hourly rate while busy on the job. The optimal c_w in the policy is not necessarily $c_w = \sup \frac{w(\tau)}{\tau}$: drivers must trade off the earnings rate while on a trip with their utilization rate; the more trips that a driver rejects, the longer the wait for an acceptable trip. In the appendix we prove the result by, starting at an arbitrary policy σ , making changes to the policy that increase the earnings rate while on a job *without decreasing the utilization rate*. Thus, each such change improves the reward $R(w, \sigma)$, and the sequence of changes results in a policy of the above form, for some threshold c' . Then, this threshold c' can be optimized, leading to an optimal policy of this form.

An immediate corollary of Theorem 2.3.1 is that $w(\tau) = m\tau$, for $m > 0$, is IC. In other words, if the platform pays a constant rate $\frac{w(\tau)}{\tau} = m$ to busy drivers, then in the single-state model it is in the driver's best interest to accept every trip. This result is driven by the following insight for Poisson arrivals: while *receiving* long trip requests is more beneficial to drivers in the single-state setting as they increase one's utilization rate (the driver is busy for a longer time until the next open period), *rejecting* short trips to cherry-pick long trips decreases utilization by the same amount.⁹ Further note that, given an earnings rate target R , calculating the multiplier m and thus an IC pricing policy is trivial.

On the other hand, affine pricing may not be incentive compatible because short trips are worth more per unit time than are long trips: $\frac{w(\tau)}{\tau} = m + \frac{a}{\tau}$. The optimal policy may thus be to accept trips in $\sigma^* = (0, T)$ for some T . However, our next proposition establishes that affine pricing is incentive compatible if the additive component stays small enough as a function of the request arrival rate:

With a single state, $w(\tau) = m\tau + a$ is incentive compatible if $0 \leq a \leq \frac{m}{\lambda}$.

⁹This insight is similar to a result of Kamble (2019); however, in our setting the driver's strategy σ is a *subset* of \mathbb{R}_+ denoting the job requests accepted, as opposed to a discrete set of prices charged. Further, in our settings the driver responds to the platform's prices instead of setting prices, enabling a wider range of IC pricing mechanisms.

The sufficient condition has a simple intuition: when open, the expected amount of time the driver must wait for the next request is $\frac{1}{\lambda}$; if on-trip time is valued at m per unit-time, then with $a = \frac{m}{\lambda}$ the additive component can be interpreted as paying for the driver's expected waiting time. Thus, while a driver may earn more per hour for a short trip than a long trip with affine pricing, such a short trip is not worth the time the driver must wait for the next trip request. We further note that the condition in the proposition is not a necessary one; however, deriving necessary and sufficient conditions in closed form requires specifying the trip distribution F .

As we'll see in the next sub-section, the structure of optimal driver policies in reaction to affine pricing differs sharply in the dynamic model.

2.3.2 Dynamic model: multiplicative pricing is not incentive compatible

In the single-state model, multiplicative pricing is incentive compatible; a driver cannot benefit in the future by rejecting certain trips if all trips have the same on-trip earning rate. In contrast, we now show that the same insight does not hold for the dynamic model, as a driver can influence future trips through the decision to accept or reject certain trips.

Theorem 2.3.2. *If $w = \{w_1, w_2\}$, there exists an optimal policy $\sigma = \{\sigma_1, \sigma_2\}$ (i.e., that maximizes $R(w, \sigma)$), defined with parameters $t_1, t_2, t_3, t_4, t_5, t_6 \in [0, \infty) \cup \{\infty\}$, such that*

- *Non-surge state driver optimal policy σ_1 :*
 - *If w_1 is multiplicative or positive affine, σ_1 rejects long trips, i.e., $\sigma_1 = (0, t_1)$.*
 - *If w_1 is negative affine, σ_1 rejects short and long trips, i.e., $\sigma_1 = (t_2, t_3)$.*
- *Surge state driver optimal policy σ_2 :*
 - *If w_2 is multiplicative or negative affine, σ_2 rejects short trips, i.e., $\sigma_2 = (t_4, \infty)$.*
 - *If w_2 is positive affine, σ_2 rejects medium length trips, i.e., $\sigma_2 = (0, t_5) \cup (t_6, \infty)$.*

Furthermore, there exist settings where t_i 's take positive finite values, and in which multiplicative pricing is not incentive compatible in either state.

We discuss the intuition in the next section. In the appendix, we prove the result for each case as follows: fixing σ_j for $j \neq i$, we start with an arbitrary open set $\sigma_i = \cup_k^\infty (\ell_k, u_k)$, recalling that open sets can be written as a countable union of such disjoint intervals. Then, we find $\frac{\partial}{\partial u_k} R(w, \sigma)$, the derivative of the set function $R(w, \sigma)$ with respect to one of the interval upper end-points of σ_i , i.e., u_k . This derivative is the infinitesimal change in the overall reward if σ_i is expanded by increasing u_k , and it has useful properties. In the surge state with multiplicative pricing, for example, $\frac{\partial}{\partial u} R(w, \sigma)$ has the same sign as a function that is *increasing* in u , for each fixed σ . With affine pricing, it has the same sign as a *quasi-convex* (positive affine in the surge state) or *quasi-concave* (negative

affine in the non-surge state) function in u , for a fixed σ . Such properties enable constructing a sequence of changes to σ_i that each do not decrease the reward $R(w, \sigma)$, with the limit being a policy of the appropriate form. In particular, we can show that any policy that is not of the appropriate form above has $\frac{\partial}{\partial u_k} R(w, \sigma) \geq 0$ for some u_k , allowing local improvements until adjacent intervals $(\ell_k, u_k), (\ell_{k+1}, u_{k+1})$ can be combined or expanded to infinity. The numerics in Section 2.5 provide examples in which multiplicative pricing is not incentive compatible, i.e., where policies of the form above with positive finite constants strictly increase driver earnings over the driver policy that accepts all trip requests.

The results of rejecting long trips in non-surge (and short trips in surge) extend to arbitrary functions where $\frac{w_1(\tau)}{\tau}$ is non-increasing (respectively, $\frac{w_2(\tau)}{\tau}$ is non-decreasing). The other two results do not hold with such generality, as the behavior of the derivative may be arbitrarily complex.

2.3.3 Why is multiplicative surge pricing not incentive compatible?

“I thoroughly dislike short trips ESPECIALLY when I’m picking up in a waning surge zone”

ANONYMOUS DRIVER

What explains the difference between multiplicative pricing being incentive compatible in the single-state model but not in the dynamic model? In the latter, a driver’s policy affects not just their earnings while they are busy, but also the fraction of time during which they are busy during the lucrative surge state. In particular, it turns out, accepting short trips during surge may *reduce* the amount of time that a driver is on a surge trip! Appendix Figure A.1 shows in an example how the fraction of time in the surge state $\mu_2(\sigma)$ changes as a function of how many short trips the driver rejects.

The anonymous driver we quote above identifies the key effect: when surge is short-lived, a driver may only have the chance to complete one surge trip before it ends. Thus, the driver may be better off waiting to receive a longer trip request, as with multiplicative surge they are paid a higher rate for the full duration of the longer trip. (Of course, there is a trade-off as rejecting too many trip requests risks not receiving any acceptable request before surge ends). In the surge state, then, multiplicative pricing does not compensate drivers enough to accept short trips that may reduce their future surge earnings. In the non-surge state, analogously, multiplicative pricing under-values long trips that may prevent taking advantage of a future surge.

Affine pricing is a first, reasonable attempt at fixing these issues. In the surge state, the additive value makes the previously under-valued short trips comparatively more valuable, as the earnings per unit time $\frac{w_2(\tau)}{\tau} = m_2 + \frac{a_2}{\tau}$ (with $a_2 > 0$) are now higher for short trips. Unfortunately, with such pricing the structure for the surge optimal policy becomes $\sigma_2 = (0, t_5) \cup (t_6, \infty)$ – if the values m_2, a_2 are not balanced correctly, the additive value is enough to make accepting extremely short trips $(0, t_5)$ profitable; for medium-length trips $\tau \in (t_5, t_6)$, however, the additive value is not large

enough to make up for the fact that accepting the trip prevents accepting another surged trip before surge ends. Similarly, negative affine pricing in the non-surge state, $w_1(\tau) = m_1\tau + a_1$, (with $a_1 < 0$) is now too harsh on very short trips but potentially not enticing enough for long trips.

Next, we fix these issues and construct incentive compatible pricing schemes for our dynamic model. Then, in Section 2.5 we leverage structural results derived here to numerically compare the incentive compatibility of additive and multiplicative surge.

2.4 Incentive Compatible Surge Pricing

We now present our main result, regarding the structure of incentive compatible pricing in the dynamic model. To this aim, in Section 2.4.1, we characterize $\mu_i(\sigma)$, how much time the driver spends in each state. In Section 2.4.2, we present incentive compatible prices, under a condition on the ratio of per-state earning rate constraints, $\frac{R_1}{R_2}$. Section 2.4.3 discusses an intuition of the IC pricing structure in terms of the driver's opportunity cost.

2.4.1 Transition probabilities and expected time spent in each state

The expected fraction of time spent in each state, $\mu_i(\sigma)$, depends both on the evolution of the world state and the trips a driver accepts. To quantify the effects previewed in Section 2.3.3, we first analyze the evolution of the world state CTMC.

Lemma 2.4.1. *Suppose the world is in state i at time t . Let $q_{i \rightarrow j}(s)$ denote the probability that the world will be in state $j \neq i$ at time $t + s$. Then,*

$$q_{i \rightarrow j}(s) = \frac{\lambda_{i \rightarrow j}}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \left[1 - e^{-(\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i})s} \right]$$

Note that $q_{i \rightarrow j}(s)$ is not just the probability that the world state transitions once during time $(t, t + s)$, but the probability that it transitions an odd number of times. This formulation emerges through a standard analysis of two-state CTMCs, in which this probability can be found through the inverse of the Laplace transform of the inverse of the resolvent of the Q-matrix for the system. Incorporating this value in closed form is the main hurdle in extending our results to general systems with more than two states. Using this formulation, the following lemma shows $\mu_i(\sigma)$.

Lemma 2.4.2. *Let $T_i(\sigma_i)$ be as defined in Lemma 2.2.1. The fraction of time a driver following strategy $\sigma = \{\sigma_1, \sigma_2\}$ spends either open in state i or on a trip started in state i is*

$$\mu_i(\sigma) = \frac{\lambda_i F_i(\sigma_i) T_i(\sigma_i) Q_j(\sigma_j)}{\lambda_j F_j(\sigma_j) T_j(\sigma_j) Q_i(\sigma_i) + \lambda_i F_i(\sigma_i) T_i(\sigma_i) Q_j(\sigma_j)}$$

where $Q_i(\sigma_i) = \lambda_{i \rightarrow j} + \lambda_i \int_{\tau \in \sigma_i} q_{i \rightarrow j}(\tau) dF_i(\tau)$

We prove this lemma by finding the expected number of sub-cycles in each state i , i.e., within a larger renewal reward cycle as defined, the expected number of sub-cycles that start with the driver being open in state i . This expectation is the mean of a geometric random variable parameterized by the probability that the driver will next be open in state j , given the driver is currently open in state i . $Q_i(\sigma_i)$ is proportional to this probability. (As with $T_i(\sigma_i)$, there is a normalizing constant $\frac{1}{\lambda_i F_i(\sigma_i) + \lambda_{i \rightarrow j}}$); the larger it is, the fewer sub-cycles that are spent in state i . It has two components: the first is the probability that the state changes before the driver accepts a trip request; the second is the probability that the world state is j when the driver completes a trip. Thus, the numerator in $\mu_i(\sigma)$ is proportional to the length of a sub-cycle in state i , times the fraction of sub-cycles that are started in state i . The larger $Q_j(\sigma_j)$ or $T_i(\sigma_i)$, the more time the driver spends in state i .

2.4.2 Incentive Compatible pricing in the dynamic model

How can the platform create incentive compatible pricing given the previously described effects? Our main result establishes when such IC prices exist, and reveals their form.

Theorem 2.4.1. *Let $R_1 < R_2$ be target earning rates during non-surged and surge states, respectively. There exist prices $w = \{w_1, w_2\}$ of the form*

$$w_i(\tau) = m_i \tau + z_i q_{i \rightarrow j}(\tau),$$

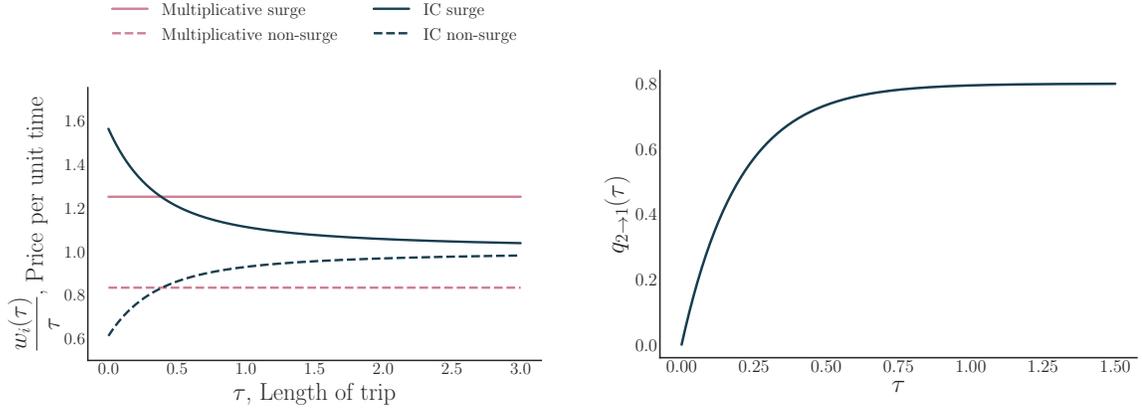
where $m_1, m_2, z_2 \geq 0$ (but z_1 may be either positive or negative), such that the optimal driver policy is to accept every trip in the surge state and all trips up to a certain length in the non-surge state. Furthermore, for $\frac{R_1}{R_2} \in [C, 1]$, there exist fully incentive compatible prices of this form, where

$$C = 1 - \frac{1}{T_1} \frac{Q_2(\lambda_{12}T_1 - Q_1) + Q_1(T_2\lambda_{1 \rightarrow 2} + Q_2)}{Q_2(\lambda_{1 \rightarrow 2}T_1 - Q_1) + \lambda_{1 \rightarrow 2}(T_2\lambda_{1 \rightarrow 2} + Q_2)} \in [0, 1),$$

and $T_i = \lambda_i F_i(\sigma_i) T_i((0, \infty))$, and $Q_i = Q_i((0, \infty))$.

Section 2.4.4 contains a proof sketch. To convey intuition, Figure 2.2a shows pricing functions in each state, plotting $\frac{w_i(\tau)}{\tau}$ against τ . Compared to multiplicative pricing with constant $\frac{w_i(\tau)}{\tau}$, IC surge pricing pays more for short trips and less for long trips. Inversely, IC non-surge pricing pays more for long trips than it does for short trips. Further, as τ increases, $w_1(\tau)$ approaches $w_2(\tau)$, reflecting the fact that the opportunity cost for long trips does not depend as strongly on the state in which it started (as discussed in Section 2.4.3). Next, observe that IC surge pricing $w_2(\tau) = m_2 \tau + z_2 q_{2 \rightarrow 1}(\tau)$ is approximately affine, as $q_{2 \rightarrow 1}(\tau)$ (plotted in Figure 2.2b) is upper bounded by $\frac{\lambda_{2 \rightarrow 1}}{\lambda_{1 \rightarrow 2} + \lambda_{2 \rightarrow 1}}$. The two components of pricing, m_i and z_i , thus balance the comparative benefit of long and short trips.

Rather surprisingly and contrary to platform design focus, the *non-surge* state is difficult to make incentive compatible. Our result establishes that there always exist payments, for any target driver earning rates $R_1 < R_2$, such that accepting every trip in the surge state is driver optimal; the same



(a) Price per unit time $\frac{w_i(\tau)}{\tau}$ for trips of different lengths τ in the each state for Incentive Compatible and multiplicative pricing when $R_2 = 1$ and $R_1 = \frac{2}{3}$.

(b) $q_{2 \rightarrow 1}(\tau)$ when $\lambda_{1 \rightarrow 2} = 1, \lambda_{2 \rightarrow 1} = 4$. IC surge pricing is well-approximated by an affine function: $z_2 q_{2 \rightarrow 1}(\tau)$ is approximately constant for longer trips.

Figure 2.2: The primitives are as follows: $\lambda_1 = \lambda_2 = 12, \lambda_{1 \rightarrow 2} = 1, \lambda_{2 \rightarrow 1} = 4$; in both states, trip lengths are distributed according to a Weibull distribution with shape 2 and mean $\frac{1}{3}$. These parameters reflect realistic average trip to wait time values, and that surge tends to be short-lived compared to non-surge times.

is not true for the non-surge state. Figure 2.3 shows how C changes with the primitives. We further give intuition for the form of payment scheme w_i and the range $[C, 1]$ in Section 2.4.3, showing how they emerge from the driver’s opportunity cost.

Finally, for a given feasible R_1, R_2 , there is a range of m_i, z_i that form an incentive compatible pricing scheme. Why? A driver who rejects a trip request waits to receive another request, during which time they do not earn money. This wait time tilts the driver toward accepting any trip request to maximize earnings. Thus, there is flexibility in the balance between short and long trip earnings. The same insight drives Proposition 2.3.1; even in the single-state model, trips do not have to have the same earnings per unit time, $\frac{w(\tau)}{\tau}$, as long as they meet some minimum threshold, $\frac{w(\tau)}{\tau} \geq c_w$.

2.4.3 Opportunity cost intuition for incentive compatible pricing

We now present some intuition to understand Theorem 2.4.1 and our incentive compatible pricing scheme. The payment $w_i(\tau)$ must account for the driver’s opportunity cost (in a VCG-like manner), i.e., how much the driver can expect to earn if they instead reject the trip request. Of course, this opportunity cost itself depends on the pricing scheme w . We now break down parts of this opportunity cost.

On-trip opportunity cost. While the driver is on-trip, the world state continues to evolve: surge might end or start, affecting the opportunity cost.

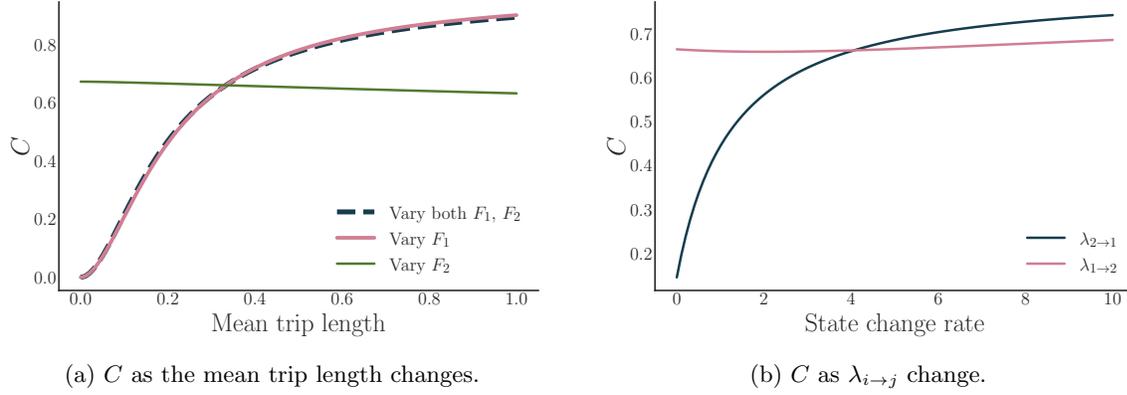


Figure 2.3: How C , the ratio R_1/R_2 at which IC pricing is feasible from Theorem 2.4.1, changes (1) with respect to the mean trip length, and (2) with respect to $\lambda_{i \rightarrow j}$. Except for those that are varied in each plot, the primitives are fixed to those used in Figure 2.2: $\lambda_1 = \lambda_2 = 12$, $\lambda_{1 \rightarrow 2} = 1$, $\lambda_{2 \rightarrow 1} = 4$ and, in both states, trip lengths are distributed according to a Weibull distribution with shape 2 and mean $\frac{1}{3}$.

Let $\phi_i^k(\tau)$ be the expected amount of time that the world is in state k during time $(t, t + \tau)$, given that it is in state i at time t . Then, by integrating $q_{i \rightarrow j}(s)$ from 0 to τ :

$$\begin{aligned}\phi_i^i(\tau) &= \left[\frac{\lambda_{j \rightarrow i}}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \right] \tau + \left[\frac{1}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \right] q_{i \rightarrow j}(\tau) \\ \phi_i^j(\tau) &= \left[\frac{\lambda_{i \rightarrow j}}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \right] \tau - \left[\frac{1}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \right] q_{i \rightarrow j}(\tau) = \tau - \phi_i^i(\tau)\end{aligned}$$

Now, let \tilde{R}_i be the driver's earnings rate while the world state is i (whether the driver is open, or on a trip that started in *either* state). \tilde{R}_i is close to but not exactly R_i , which instead is the earnings rate counting open time and trips that *start* in state i . Then, the driver's opportunity cost during time $(t, t + \tau)$, starting in state i is

$$\tilde{R}_i \phi_i^i(\tau) + \tilde{R}_j \phi_i^j(\tau) = \left[\frac{\lambda_{j \rightarrow i} \tilde{R}_i + \lambda_{i \rightarrow j} \tilde{R}_j}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \right] \tau + \left[\frac{\tilde{R}_i - \tilde{R}_j}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \right] q_{i \rightarrow j}(\tau)$$

Though \tilde{R}_i is not a simple expression in terms of R_i , several insights emerge:

One. The “network minutes” on-trip opportunity cost has the form, $m'_i \tau + z'_i q_{i \rightarrow j}(\tau)$, matching our IC scheme (which further incorporates complications ignored here).

Two. As trip length $\tau \rightarrow \infty$, the first component $\left[\frac{\lambda_{j \rightarrow i} \tilde{R}_i + \lambda_{i \rightarrow j} \tilde{R}_j}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \right] \tau$ dominates the opportunity cost. This component does not depend on starting state i ; the stationary distribution of a positive recurrent CTMC does not depend on the starting state. This fact implies that we cannot always construct incentive compatible prices, for any R_1, R_2 : as $\tau \rightarrow \infty$, the trip's opportunity cost does

not depend on the starting state i , and so the trip’s payments must be similar, $w_1(\tau) \approx w_2(\tau)$. When all trips in the non-surge state are long, i.e., F_1 is concentrated around large values, the earnings rate in each state must be similar, $R_1 \approx R_2$.

C encodes such constraints, as shown in Figure 2.3. As the mean of $\tau \sim F_1$ goes to 0, then $\lambda_{12}T_1 - Q_1 \rightarrow 0$ and so $C \rightarrow 0$, and so the range of feasible $\frac{R_1}{R_2}$ expands. Similarly, $\lambda_{2 \rightarrow 1}$ also plays an important role. When small, the surge state is long. Thus, a driver will receive many trips during surge regardless of how long their last non-surge trip is—and so long trips during non-surge are no longer constrained to be highly paid compared to short trips.

Continuation value opportunity cost It is not sufficient to consider just the opportunity cost for the duration of the trip: the driver’s counter-factual earnings by rejecting the trip depends on future trips accepted. Such counter-factual trips both (1) pay the driver according to their starting state even after a world state transition, i.e., the difference between R_i and \tilde{R}_i above; and (2) potentially are still in progress past time $t + \tau$, when the current trip ends. This second complication is illustrated in Figure A.1, where a driver can extend the time spent on trips starting in the surge state by rejecting short surge trips. The effect depends on the lengths of future potential trips, i.e., $T_i(\sigma_i)$, and state transitions during those trips, $Q_i(\sigma_i)$, and is incorporated in both C and the pricing scheme.

2.4.4 Proof sketch of Theorem 2.4.1

The result is shown in the appendix by manipulating the derivative of the reward function with respect to the policy σ . In particular, when the pricing function is of the given form with the appropriate constants m_i, z_i , then *any* policy $\sigma = \{\sigma_1, \sigma_2\}$ can be locally improved by adding more trips to it, i.e., the overall reward is non-decreasing as the driver accepts more trips: $R(w, \sigma') \geq R(w, \sigma), \forall \sigma \subseteq \sigma'$. This result follows from $\frac{\partial}{\partial u} R(w, \sigma) \geq 0$, for all u, σ , given the constraints, where u is an upper endpoint of the policy in a state, $\sigma_i = \cup_k (\ell_k, u_k)$.

The key step is finding sufficient constraints for this derivative to be positive with a pricing function of the given form, given any σ_i , as opposed to just $\sigma_i = (0, \infty)$. This difficulty emerges because incentive compatibility is a global condition on the set function $R(w, \sigma)$. In particular, we need to express these constraints simply—e.g., as a function of just $T_i((0, \infty)), Q_i((0, \infty))$, instead of the values $T_i(\sigma_i), Q_i(\sigma_i), \forall \sigma_i \subseteq \mathbb{R}_+$. The C presented in the theorem statement results from such a set of constraints on m_i, z_i .

2.5 Numerics: Incentive Compatibility with Additive Surge

We now analyze surge policies that reflect practice at ride-hailing platforms today. Non-surge pricing is typically approximately *multiplicative*, i.e., $w_1(\tau) = m_1\tau$, where m_1 is the base time (and distance)

rate for a ride. We consider two types of affine *surge* pricing w_2 , which differ in their relationship to w_1 through a single parameter:

$$\begin{array}{lll} \textbf{Multiplicative surge:} & w_2(\tau) = m_2\tau & m_2 \geq m_1 \\ \textbf{Additive surge:} & w_2(\tau) = m_1\tau + a_2 & a_2 \geq 0 \end{array}$$

Multiplicative surge uses a multiplier m_2 larger than the base fare m_1 , and $\frac{m_2}{m_1}$ is reported on the heat-map as in Figure 2.1a; additive surge uses the same base fare multiplier m_1 but adds a factor a_2 that is reported on the heat-map as in Figure 2.1b. These functions are trivial to calculate, given fixed primitives and target earnings rate R_2 in the surge state.

Appendix Figure A.2 shows these types of pricing, compared to the incentive compatible pricing function. Multiplicative surge has constant $\frac{w_2(\tau)}{\tau}$ and so under-pays short trips and over-pays long-trips compared to IC pricing. Additive surge asymptotically (for large τ) pays the same as multiplicative non-surge pricing, i.e. $\lim_{\tau \rightarrow \infty} \frac{w_2(\tau)}{\tau} = \lim_{\tau \rightarrow \infty} \frac{w_1(\tau)}{\tau} = m_1$. As a result, it over-pays short trips and under-pays long trips compared to IC surge pricing.

Uber has recently started a transition from multiplicative to additive surge. In this section, we argue that the additive component is more important than the multiplicative component for incentive compatibility in parameter regimes of interest.

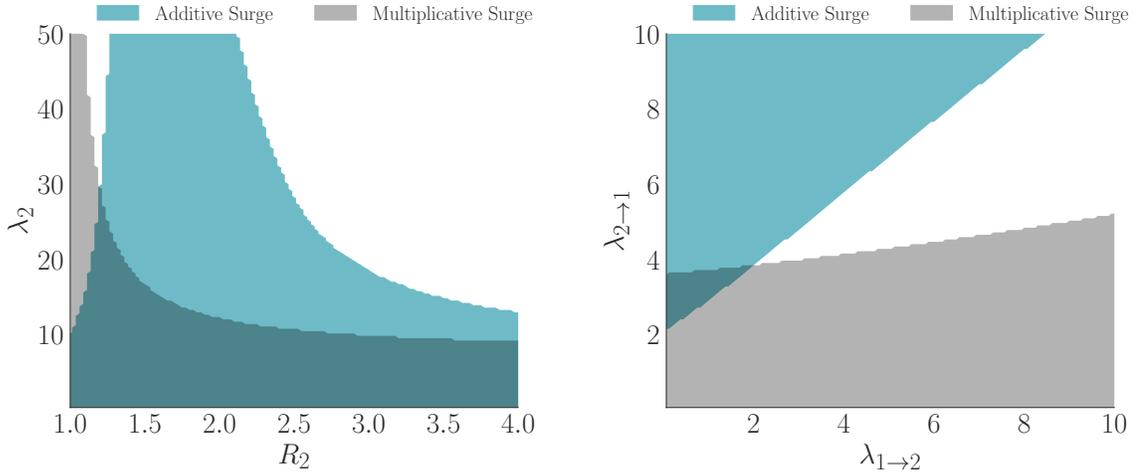
2.5.1 Computing optimal driver policies

Theorem 2.3.2 establishes that multiplicative pricing (and, more generally, affine pricing) may not be incentive compatible in general. However, we still wish to compare the various types of surge pricing, and to analyze the regimes under which each is incentive compatible.

However, to do this comparison, one needs to calculate optimal driver policies with respect to a pricing function. Recall that the optimal driver policy in each state σ_i is some subset of \mathbb{R}_+ . Finding such optimal subsets for general pricing functions w is intractable, and so Theorem 2.3.2 is particularly important for computational reasons. It establishes that, for any affine pricing structure in the surge state, there exists a driver optimal policy of the form $(0, t_1) \cup (t_2, \infty)$, for some t_1, t_2 . We only need to find the values for these parameters that maximize the driver reward among sets of this form, and the resulting policy is optimal; this search is tractable with grid search and numeric integration. Note that the proposition does not establish uniqueness of the driver optimal policy; we thus choose the policy that maximizes the fraction of trips accepted in our computations.

2.5.2 Results

We now study the regimes in which each surge mechanism is incentive compatible. The shaded regions in Figure 2.4 correspond to areas where the surge pricing function is fully incentive compatible



(a) With R_2 , surge state earnings rate, and λ_2 , surge state job arrival rate. $R_2 \in [1.1, 3]$ is common in practice. (b) With $\lambda_{2 \rightarrow 1}, \lambda_{1 \rightarrow 2}$, rates for world state changing. $\lambda_{2 \rightarrow 1} \gg \lambda_{1 \rightarrow 2}$ is common in practice.

Figure 2.4: Incentive compatibility for each type of surge. The shaded regions are where the respective scheme is incentive compatible in the surge state ($\sigma_2 = (0, \infty)$ is optimal). When not varied, $\lambda_1 = \lambda_2 = 10, \lambda_{1 \rightarrow 2} = 1, \lambda_{2 \rightarrow 1} = 4, R_2 = 3.33, R_1 = 1$, and trip lengths in both states are distributed according to a Weibull distribution with shape 2 and mean 0.3. We assume every trip is accepted in the non-surge state.

in the surge state ($\sigma_2 = (0, \infty)$ is optimal). For example, when $R_2 = 2, \lambda_2 = 30$, additive surge is incentive compatible, but multiplicative surge is not.

As illustrated in Appendix A.2 with data from the RideAustin marketplace, ride-hailing platforms most often operate in the following parameter regimes: (1) surge is between 1.1 and 3 times more valuable than non-surge; (2) surge is short-lived compared to non-surge periods ($\lambda_{2 \rightarrow 1} \gg \lambda_{1 \rightarrow 2}$); (3) and in a typical surge the driver receives several trip requests ($\frac{\lambda_2}{\lambda_{2 \rightarrow 1}} > 1$, but small) but only completes one or two such trips ($\frac{1}{\lambda_{2 \rightarrow 1}} \approx$ mean trip length). Additive surge is incentive compatible in much more of this regime than is multiplicative surge, supporting Uber’s recent shift from multiplicative to additive surge.

We can also draw qualitative insights in terms of sensitivity to the primitives, similar in spirit to effects in the form of C in Theorem 2.4.1. Figure 2.4a shows the sensitivity with respect to λ_2 and R_2 . As the arrival rate of jobs in the surge state, λ_2 , increases, it becomes optimal for the driver to reject some trips: “cherry-picking” becomes easier, as the driver is likely to receive many more trip requests before surge ends. Similarly, as surge becomes increasingly more valuable compared to non-surge (R_2 increases), the incentive to reject non-valuable trips in the surge state increases.

For additive surge, there is an interesting non-monotonicity with R_2 : when $R_2 \gg R_1$, the effect above dominates, and long trips are rejected. When the surge state is moderately more valuable

than non-surge, additive surge effectively balances the payments for different trip lengths and so is incentive compatible. When the two states are nearly equally valuable, again the optimal driver policy rejects long trips: our single-state model approximates the system, and so additive surge may not be incentive compatible, cf. Theorem 2.3.1.

Figure 2.4b shows the effects of the relative lengths of surge and non-surge. Here, the two types of surge are incentive compatible in opposing regimes. When $\frac{\lambda_{2 \rightarrow 1}}{\lambda_{1 \rightarrow 2}}$ is large, surge is comparatively rare and short, and so short trips are naturally under-valued – accepting them decreases the time spent in the surge state – and additive surge is incentive compatible. With long-lasting surge (small $\frac{\lambda_{2 \rightarrow 1}}{\lambda_{1 \rightarrow 2}}$), on the other hand, the world almost seems unchanging during surge, and so multiplicative surge becomes incentive compatible. In modern ride-hailing platforms, the scenario with short, infrequent surge is more common (as illustrated with RideAustin data in Section A.2), and so additive surge is preferable.

2.6 Empirical Comparison of Surge Mechanisms

We now study how the various surge mechanisms affect driver earnings in practice using publicly available trips data from RideAustin, a nonprofit ride-hailing company based and operating in Austin, Texas. We show that additive surge effectively balances the relative value of short and long surged trips, in contrast to the multiplicative surge pricing scheme used in practice by the platform, which comparatively undervalues short surged trips.

After reverse-engineering the functional form of the actual driver payments, we calculate both status quo (with multiplicative surge) and simulated (with additive surge) driver earnings. For each payment scheme, we estimate the driver’s value in receiving and accepting a given trip request, as a function of the trip—where “value” is the increase (or decrease) in the driver’s earnings over the next 90 minutes as a result of accepting the given request.

We note that this data is not the result of an experiment with additive surge, and thus our analysis describes what changes would occur in driver earnings with the new pricing function *if driver behavior does not change*.¹⁰ Nevertheless, the exercise provides useful evidence for what would happen with such pricing functions in a real-world setting: such as when surge has more than two levels and may not evolve in a Markovian manner, the driver is not paid for the time it takes to drive to the rider, and where location plays a role. Furthermore, as the data observed is at the *completed* trip level (i.e., requests which the driver accepted), results showing that the driver would be better off accepting the same trip in the counter-factual world should directionally hold even as driver behavior changes.

The rest of this section is organized as follows: in Section 2.6.1, we describe the data and the context, and Section 2.6.2 contains our analysis and results. Appendix Section A.2 contains

¹⁰We are not concerned with rider behavior changing, as with decoupled pricing the rider pricing can remain the same even as the driver payments change.

supporting details, and both the data and our full replication code is available online.¹¹

2.6.1 Data setting and analysis description

This analysis is enabled by the rich dataset, spanning from June 2016 to April 2017, during which RideAustin experienced tremendous growth and was one of the largest ride-hailing marketplaces serving the area. The data is at the completed trip level. Komanduri et al. (2018) study the same dataset and provide useful statistics about driver earnings, platform growth, and the service’s relationship to public transportation.

We focus our analysis on approximately the last two months of this period, February 16, 2017 to April 10, 2017, as (1) we can reliably reverse engineer the payment function used by the platform during this period, and (2), the underlying marketplace was fairly stable during this period, except for one week of high, atypical demand and surge, corresponding to the SXSW Music Festival held in Austin. (Figure A.4a in the Appendix, e.g., shows the trips per day during this period). We discard trips longer than 1 hour or shorter than 30 seconds and other trips with data errors; 6440 such trips were discarded. In total, we analyze 503,383 completed trips by 3811 drivers. (For the analyses which aggregate multiple trips, such as driver earnings in a given time period, we discard aggregations that include a discarded trip). The full pre-processing sequence is described in the appendix.

Several dataset features make it attractive for our analysis when compared to other publicly available ride-hailing datasets. Most importantly, there are consistent *driver IDs* attached to each trip. Second, for each trip, there is a value for the *total fare* paid by the rider, along with terms that contribute to this calculated fare: *trip duration (in time and distance)*, *payment rate (in time and distance)*, *surge factor*, *standard additive fare (Pickup)*, and *trip class* (Regular vs Luxury vs SUV).¹² These features allow us to track a driver’s trajectory and earnings over a day and the entire year, reverse engineer how RideAustin calculates payments, and simulate additive surge payments.

Constructing payment functions

To simulate driver earnings with additive surge, we must first reverse engineer how the platform’s actual *total fare* was calculated, a non-trivial task as the calculation changes over time in the dataset and is not documented. (The last two months were chosen for analysis partially because the calculation remains constant during this period, and we are able to reliably reverse engineer it.) We find

¹¹Data: <https://data.world/ride-austin>. Code link removed for anonymous peer review.

¹²Our results include trips from all trip classes, as a given driver may be cross-dispatched across trip classes.

that this **status quo fare** is approximately:¹³

$$\max(B + \textit{Pickup}, \textit{MinFareForClass}) \times \textit{SurgeFactor}.$$

$B \triangleq (\textit{DistanceRate} \times \textit{Distance}) + (\textit{TimeRate} \times \textit{Time})$ is the trip time and distance fare, only counting when the rider is in the car (recall that current practice deviates from the theory in that driving to the rider is typically unpaid). $\textit{MinFareForClass}$ is \$4 for Regular trips and \$10 otherwise. $\textit{SurgeFactor}$ of 1 indicates no surge, comprising 70% of trips. It increments in multiples of 0.25, and 97% of surged trips have a factor of at most 3.

Then, we construct the following payment for each trip, to simulate how the driver would be paid with additive surge, i.e., **Additive surge with base fare**:

$$\max(B + \textit{Pickup}, \textit{MinFareForClass}) + [(\textit{SurgeFactor} - 1) \times A_{\textit{SurgeFactor}}]$$

$A_{\textit{SurgeFactor}}$ are surge factor dependent constants that are set such that this alternative payment function spends the same amount of money overall for each surge factor as does the status quo fare. In other words, the alternative payment does not change the mean trip payment conditional on the surge factor, but does change how money is allocated to various trips within that surge. This choice reflects our theory in assuming an exogenous R_i , and removes any degrees of freedom in setting $A_{\textit{SurgeFactor}}$. If instead we used a single constant across surge factors, Additive surge with base fare may pay different amounts on average for the same surge factor than does the status quo fare.

Matching open drivers

We are interested in the value of a trip request to a driver; to calculate this value, as described below, we first need to *match* the driver of each given completed trip to a nearby driver who is also open to receive a trip request at the time of the request. This matched driver’s earnings then serve as a counter-factual for the given driver’s earnings had the driver rejected the request.

In the dataset, we observe trip start and end times and locations but not driver locations when they are not on a trip or even whether they still have their app open. We also observe the time at which a driver received a given trip request but not their location at this time, due to what seems like a data export bug. This data does not allow us to simply query for other open drivers nearby who could have (but did not) receive a given trip request.

Instead, we estimate matches as follows, leveraging recent, nearby completed trips. First, we define a “matching distance” between pairs of $(\textit{date-time}, \textit{location})$ tuples. Events with small matching distances occur nearby and at similar times. The exact function with how time and geographic distance are weighted is specified in the appendix. Then, for each given trip we find a driver who

¹³The payment includes a multiplier of 1.01 and an additive value of 2.02. From publicly available information, we assume that the platform takes a fixed commission independent of trip length, and so the driver receives everything but the \$2.02 (RideAustin, 2019). On average, this reversed engineered fare differs from *total fare* by less than 1 cent.

recently completed a trip nearby and has yet to start another trip. For the given trip, we use its *start* location (where the rider was), and the *dispatch* time (when the rider’s request was accepted). Next, we calculate the matching distance between this tuple, and each recent completed trips’ *destination* time and location. Finally, we choose the closest match, filtering out drivers who are the same as the given trip’s driver, who have started another trip before the given trip’s start time, or who ended their session (did not start any trip in the next hour).

In the appendix, we provide results from a different but complementary matching method, as well as additional information about the matches and their quality.

Calculating the value of a trip to a driver

We now measure how valuable a trip is to a driver, through a notion we call *trip indifference*: given a specific trip request length τ , in expectation the driver is at least as well off accepting the request as rejecting it, assuming some future behavior. Given a trip and a matched driver as described in Section 2.6.1, we estimate this measure as follows: we compare the two drivers’ future earnings over the 90 minutes after the accepted trip begins—the higher the given driver’s earnings over that of the matched driver, the more valuable the given trip request. If there is no difference, i.e., the matched driver in expectation earns the same amount, then the given driver should be “indifferent” between accepting or rejecting the request.¹⁴

Suppose trips are mis-priced and do not fully incorporate the drivers’ temporal externalities. Then, trips of different lengths τ would vary in the value delivered to drivers. We would expect to see the average earnings differential, conditional on trip length, to vary as a function of the trip length; i.e., receiving a long trip during surge may be more valuable to a driver than is receiving a short trip.

We note that bias in the matching process may shift the expected earnings difference, but should not differentially affect the distributions of earnings differences induced by each payment function, as the same matches are used for each payment function. As robustness checks, in the appendix we vary both the matching function and the length of time over which we calculate the two drivers’ earnings.

2.6.2 Analysis and results: value of short and long trips

Figure 2.5 shows the difference in value between short (below the median trip length) and long (above the median) trips, as it changes with surge. As expected, it is more beneficial for drivers to receive trips with higher surge factors. However, with the platform’s existing multiplicative surge payment function, only long trips become more valuable as the surge factor increases; even at high

¹⁴Trip indifference is related to our theoretical notion of incentive compatibility as follows. Suppose the given driver accepts all future requests over the next 90 minutes. Then, if a payment scheme is incentive compatible, the earnings difference between the given driver who accepts trip τ and the matched driver will be at least 0 for all τ .

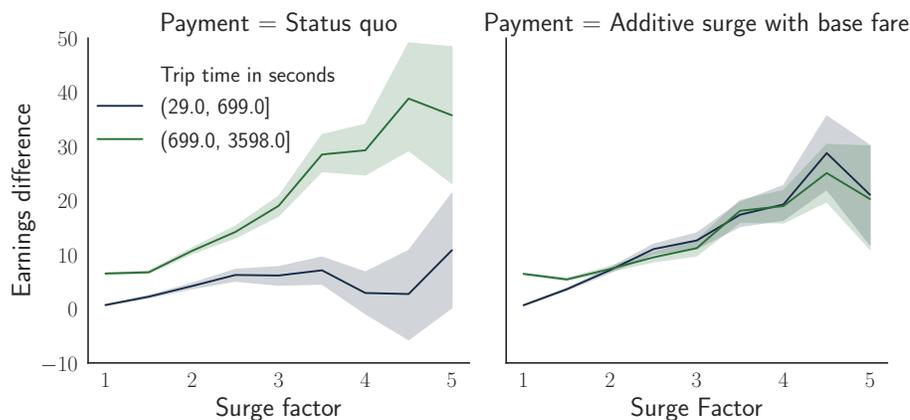


Figure 2.5: Difference in earnings over the next 90 minutes for the driver of a given accepted trip request, and a matched driver who also was open nearby at the time of the request, conditional on surge factor (rounded to nearest 0.5) and length of trip. Error bars are 95% bootstrapped confidence intervals.

surge factors, drivers would have often had higher earnings had they rejected short trip requests. With additive surge, in contrast, trips of all lengths become more beneficial on average as surge increases. During high surge times, additive surge increases the value of short trips by about \$15 per hour.

In the appendix, we further simulate a world with the RideAustin data, but with surge being common and extremely valuable (we “flip” the surge factor). This analysis illustrates that our other insights also extend to practice, with there being settings where non-surge periods cannot be made incentive compatible, and where neither multiplicative nor additive surge correctly balance the value of short and long trips. We also show how hourly driver earnings during a single “shift” change with additive and multiplicative surge, and how the former leads to more stable earnings. Overall, this analysis suggests the substantial difference that changing the structure of payments can make, and the comparative benefits of additive surge in practice under common regimes in ride-hailing.

2.7 Conclusion

In this work, we studied the problem of designing incentive compatible mechanisms for ride-hailing marketplaces. We presented a dynamic model to capture essential features of these environments. Even-though our model is simple and stylized, it highlights how driver incentives and subsequently dynamic pricing strategies would change in the presence of stochasticity. Our numeric and empirical analysis suggests the importance of such components in practice. We hope our work inspires other researchers in this area to incorporate such uncertainty in their models, as it is one of the biggest challenges faced in practice.

An important direction for extending our work is studying matching and pricing policies jointly, i.e., how to best match open drivers to riders in the presence of such effects, cf. (Ashlagi et al., 2018; Banerjee et al., 2017a,b, 2018; Feng et al., 2017; Hu and Zhou, 2018; Kanoria and Qian, 2019; Korolko et al., 2018; Özkan, 2018; Özkan and Ward, 2016; Zhang et al., 2017). In this work, we look at incentive compatible pricing. The platform, in addition to pricing, can use matching policies to align incentives.

Part II

Designing Rating Systems in Online Marketplaces

Chapter 3

Designing Informative Rating Systems: Evidence from an Online Labor Market

3.1 Introduction

Rating systems are an integral part of modern online markets. Marketplaces for products (Amazon and eBay), ridesharing (Lyft and Uber), housing (Airbnb), and freelancing all employ rating systems to vet platform participants. Buyers rely on ratings to choose which products to buy and how much to pay, and platforms use ratings to identify both poor and great performers, and in ranking search results. Ratings are consequential: a high score typically directly translates to more visibility and sales. Indeed, without effective mechanisms to collect feedback after matches, online markets would be “flying blind” in reducing search frictions between buyers and sellers.

Despite their central importance, extensive prior work suggests the standard rating systems of many online platforms are *not sufficiently informative*, i.e., ratings do not sufficiently discriminate between high and low quality sellers. A major causal factor in this lack of informativeness is *rating*

This chapter is joint with Ramesh Johari. Published as an extended abstract at the ACM Conference on Economics and Computation (EC), in 2020 (Garg and Johari, 2020). A journal version is in submission (Garg and Johari, 2019b). This work benefited from substantial implementation and experimental efforts led by Cary Luu. We also thank Michael Bernstein, Hayden Brown, Ashish Goel, John Horton, Shane Kinder, and participants of the Market Design workshop at EC’18. This work was funded in part by the Stanford Cyber Initiative, the National Science Foundation Graduate Research Fellowship under grant DGE-114747, and the National Science Foundation under grants 1544548 and 1839229.

inflation, where most participants predominantly receive high ratings. Heavily skewed rating distributions lead to systems in which noise dominates, and as a result buyers are challenged to extract meaningful signal from available rating scores.

Several empirical studies have established the prevalence of rating inflation. On eBay, more than 90% of sellers studied between 2011 and 2014 had a rating of at least 98% positive, and more transactions result in a dispute than in negative feedback (Nosko and Tadelis, 2015). On the online freelancing platform oDesk, average ratings rose by one star over seven years (Filippas et al., 2019). On Uber, an average rating of 4.6 out of 5 stars puts a driver at risk of deactivation (Cook, 2015). On Airbnb, almost 95% of hosts have an average rating of 4.5-5 out of 5 stars (Zervas et al., 2015). On Amazon, ratings tend to be bimodal with a big peak near the most positive score and then a (much) smaller one near the most negative one (Hu et al., 2009). Numerous other works report similar findings; Tadelis (2016) provides a thorough review of the literature.

The empirical literature concludes that inflated ratings are less informative about quality differences among participants. For example, Filippas et al. (2019) notes that the increase in average ratings at oDesk could not be explained solely by higher seller performance, indicating that rating informativeness dropped over time as ratings inflated. As a consequence of inflation, negative ratings carry outsized influence, because they are so rare; for example, Cabral and Hortacsu (2010) find that on eBay a seller's *first* negative feedback reduces her weekly sales growth rate from 5% down to -8%.

In this paper we investigate whether platforms can improve the quality of information obtained by changing the design of the *rating scale* that they employ. In particular, we ask: by carefully choosing both the meaning and importance of different answer choices in a rating scale, can platforms elicit higher quality information from their raters, i.e., such that the platform recovers the true relative qualities of sellers with fewer ratings? Our main contributions are as follows.

Reducing rating inflation via positive-skewed verbal scales. First, we establish evidence that a careful choice of the rating scale can in fact strongly reduce rating inflation. In particular, we analyze a test in the live rating system of a large online labor market. In this test, the platform asks buyers to choose from a list of phrases (e.g., *Best Freelancer I've Hired*) or adjectives (e.g., *Fantastic!*). Our results show that platforms can effectively combat rating inflation by using positive-skewed verbal scales: the rating distribution obtained from such scales is substantially more dispersed than under the “standard” star rating scale. Most starkly, in our experiment, 80.6% of freelancers received the best possible numeric (i.e., star) rating, but less than 35.8% were rated with the highest-ranked verbal phrase across non-numeric treatment cells. We further provide evidence that inflation *over time* can be countered: ratings on our additional question did not inflate over the test time period, in contrast to an inflation of about 0.3 points (on a five star scale) over a similar time period after the introduction of a new numeric rating system on the same platform, cf. Filippas et al. (2019). Our findings suggest that in platforms today, the *norm* is that any *acceptable* experience is given the top

numeric rating, with the rest of the scale reserved for various degrees of *unacceptable* experiences.

Positive-skewed verbal scales yields more informative ratings. Second, we establish evidence that the verbal scales we tested yield *more informative ratings*. In particular, we show ratings given with the positive-skewed verbal adjective scales are more predictive of whether a freelancer will be re-hired by the client in the near-future: clients are up to 30.8% more likely to rehire the freelancer during the test period after giving them a top rating from a positive-skewed scale than after giving them the top numeric score. In addition, for each freelancer we estimate their quality through the experimental data itself (carefully handling endogeneity concerns) and then produce an estimated joint distribution of freelancer quality and the ratings they receive with a given ratings scale. The distributions qualitatively reveal that positive-skewed verbal scales are much more informative about freelancer quality than are numeric scales.

A principled approach to comparing rating system designs. Third, we provide a principled approach to comparison of different rating system designs. In particular, we develop a metric on the joint distribution of seller quality and resulting ratings that directly reflects the typical goal of a rating system: to learn about sellers *as quickly as possible*. We develop a mathematical framework where the performance of a rating system is measured through the *large deviations* rate of convergence of the seller ranking via observed score to the true underlying seller quality ranking. This rate is the exponent in the exponential decay of the Kendall’s τ distance between the estimated and true seller rankings over time.

We develop a stylized model for rating system design within which we calculate these convergence rates. We define a fictitious “marketplace” in which sellers accumulate ratings over time, with match rates proportional to their quality. Buyers rate sellers using a multi-level *rating scale*, i.e., buyers are asked to answer a multiple choice question (e.g., 1-5 stars, or a set of adjectives describing the interaction) when rating the seller. The platform can choose amongst several rating scale options that differ in their levels (e.g., adjectives); these scale options induce different buyer rating behavior. The platform can also set the scores to assign to these adjectives (e.g., the seller might receive a “5” if the buyer selects the best adjective, and a “3.7” if they select the second best adjective). Within this marketplace, different design choices (scale choices and scores) differ in the rates of convergence to the true quality ranking they induce, with higher rates reflecting better designs.

We show that given behavioral data of how buyers have rated sellers under various rating system designs in our test, this framework can be effectively employed to compare and select among the designs. In particular, we apply this framework to the data from our online labor market test. This process reveals the quantitative gains in convergence rate obtained by verbal rating scales over the naive numeric rating system. Interestingly, our framework also reveals that the first order effect on the rate of convergence comes from the choice of verbal descriptions on the scale; optimizing the choice of scores yields a lower order improvement in performance.

Taken together, our results suggest that platforms have much to gain by optimizing the *meaning*

of the levels in their rating systems, and in particular using positive-skewed verbal rating scales instead of numeric scales. Our managerial insight is that ratings on online platforms are not doomed to be highly inflated; rating behavior is responsive to how the system is designed, and *good* rating behavior can be both quantified and obtained through a structured design methodology. Our entire approach of experimenting with various rating scales and then choosing amongst them in a principled manner also provides a framework for doing the same in other ratings contexts, including where other behavioral challenges (such as bias or deflation) may be present. For example, in Section B.2 in the Appendix, we repeat our experiment, design approach, and analysis in a synthetic rating setting on Amazon Mechanical Turk where we have access to expert ratings on item quality.

The remainder of the chapter is organized as follows. Section 3.2 contains related work. In Section 3.3, we describe the labor market test, with results presented in Section 3.4. In Section 3.5 we describe a model and approach to evaluating and designing a multi-level rating scale. Finally, in Section 3.5.3, we apply our design approach to the data from our labor market experiment. The Appendix contains additional information and robustness analyses for our labor market test, a second application of our design approach via a synthetic experiment on Mechanical Turk, and proofs.

3.2 Related literature

Challenges in designing effective online rating systems are well-documented. To help explain the empirical inflation findings discussed above, one branch of the literature focuses on how ratings are given after bad experiences, and in particular conditions under which buyers either don't leave a review at all or leave a *positive* review. On Airbnb, for example, Fradkin et al. (2018) find that inducing more reviews resulted in more negative reviews, suggesting that those with negative experiences are less likely to normally submit a review. Though historically this inflation has been thought of as a *strategic* response to potential retaliation, recent evidence indicates that *social* pressure also plays a role. For example, sellers incentivize reviews (of any kind) by offering discounts, potentially creating an implicit social obligation for reviewers to reciprocate with a positive review (Cabral and Li, 2015; Li and Xiao, 2014). Such effects, along with outright fraud and sellers asking for higher ratings, contribute to rating inflation.

3.2.1 Platform measures to counter or encourage inflation

Platforms are aware of the inflation problem and have invested in fixing it. Most existing solutions try to decrease retaliatory pressure from sellers or to encourage more buyers to submit reviews. In 2007, eBay implemented one-sided feedback (i.e., only buyers rating sellers), with anonymous ratings presented only in aggregate; the platform later eliminated negative buyer ratings altogether (Bolton et al., 2013). Through a test with private feedback, oDesk reports that such feedback predicts both

future private *and* public feedback better than does public feedback, and there is evidence that buyers utilize private ratings more than they do public ratings (Filippas et al., 2019).¹ Other work has attempted to align buyer incentives with providing informative reviews (Gaikwad et al., 2016), but the approach has not yet been widely adopted. Despite such fixes, the problem of inflation largely remains on online platforms, consistent with the hypothesis that norms have shifted so that even average experiences are given the top numeric value.

This literature suggests that many initially effective ideas may not have a first order effect in increasing the informativeness of ratings, especially in the long-term: rating behavior on online platforms is not static. Filippas et al. (2019) show that inflation happens over time: on the same online labor market as in our test, average public ratings over a span of nine years went from below 4 stars to about 4.8 stars. This view is consistent with the “disequilibrium” view of rating system design described by Nosko and Tadelis (2015).

3.2.2 Survey design and rating inflation in other contexts

Rating inflation and the question of rating system design are also prevalent in other contexts. For example, grade inflation in education is an oft-recognized issue (Johnson, 2006). Proposed solutions include forcing educators to deflate grades (either by assigning quotas to each grade or by eliciting rankings) or standardizing grades after the fact (Blum, 2017; Lackey and Lackey, 2006). Similar methods are used to evaluate employees (Shaout and Yousif, 2014) and athletes. In baseball, for example, scouts rate athletes on a numeric scale that spans from 20 to 80 (Gines, 2017); however, scouts differ in how they evaluate talent or otherwise have heterogeneous biases, and teams may use sophisticated systems to calibrate the information provided by each scout (Reiter, 2018). On online platforms, in contrast, it may not be desirable to impose ratings quotas on buyers or feasible to assess the rating ability of individual buyers (though these are interesting avenues for future work).

An alternate approach to counter grade inflation is adding and labeling rating levels (e.g., plus-minus grading, or providing suggested mappings from relative ranking to grade) in order to *behaviorally* induce more dispersed grade distributions from educators (Blum, 2017; Lackey and Lackey, 2006). This solution is similar to the well-studied idea of using labels for scales in survey responses, in which the specific design of rating scales – including the specific words, number of words, and their positive-negative balance – is known to affect responses (Hicks et al., 2000; Klockars and Yamagishi, 1988; Krosnick, 1999; Parasuraman et al., 2006). In such solutions, the raters are not forced or even explicitly asked to answer in a certain manner; rather, the question and answer choices are presented in a way such that raters naturally behave as the survey designer wishes them to.

Our behavioral results are consistent with this latter literature, *despite* the presence of incentive issues as discussed above: scale design can have a first order effect on the quality of responses in real

¹ “Public ratings” are shown publicly, non-anonymized, e.g., “A rated B 5 stars.” “Private ratings” are either shown as a summary statistic, e.g., “B averages 4.6 stars”, or not shown at all and used only internally by the platform.

rating systems. Although this finding aligns with the survey design literature, as discussed above our study is preceded by a long line of rating systems literature in which substantive changes (making ratings private, trying to prevent retaliation, or UI changes) do not in practice lead to sufficiently informative rating systems. Given the potential costs for giving negative ratings posited by previous work, it is not clear *a priori* that any change will induce raters to do so; our work provides one path forward.

Beyond this behavioral insight, we provide a theoretical framework that a survey designer in any context can use to pinpoint the most informative design for their setting in a principled manner. For example, in the Appendix we apply our approach to a setting more similar to standard survey design and crowd-sourcing, and it yields a non-trivial rating system design that outperforms others.

3.2.3 Theoretical analyses of ratings

Recent literature has attempted to explain rating behavior, and inflation in particular, through a variety of models (Cabral and Hortacsu, 2010; Filippas et al., 2019; Fradkin et al., 2018; Immerlica et al., 2010). Much of this work seeks to understand how buyer incentives may result in an equilibrium in which they provide with dishonest ratings, or how sellers may be incentivized to accumulate high ratings and then give low effort. For example, Filippas et al. (2019) posit that high ratings are unavoidable when sellers are affected by negative ratings, as buyers are incentivized to incorrectly give positive ratings even after negative experiences.

Several recent works also study the speed of learning in rating systems and other similar contexts (Acemoglu et al., 2017; Che and Horner, 2015; Ifrach et al., 2017; Johari et al., 2017; Papanastasiou et al., 2017). In these works, the platform influences *which matches occur* through its design, and this affects the learning rates. In contrast, we take the matches as given and show how the platform can meaningfully design *what it learns from each match*. Finally, in Chapter 4, we consider the optimal design of *binary* rating systems, for which far more theoretical structure exists.

3.3 Online labor market experiment description

Our work focuses on whether we can improve the design of the feedback systems used in online platforms. As we have noted, the literature suggests that despite substantial effort across a variety of platforms, rating behavior has not changed for the better over time: average ratings on platforms tend to be extremely high or “inflated” (see discussion in Section 3.2). A significant consequence of this inflation is that current ratings systems and their resulting distribution of ratings do not provide information that can effectively and efficiently differentiate high quality participants from low quality participants.

In this section, we propose a simple but under-explored innovation in the design of a rating system: using *positive-skewed* verbal phrases in the rating scale. We study the effect of such a

change through the results of a randomized controlled trial on the rating system of a large online labor market. In this test, new ratings questions were introduced in a feedback form clients submit upon finishing a job with a freelancer.

The section is structured as follows. In Section 3.3.1, we further discuss our motivation and hypotheses. In Section 3.3.2, we briefly describe the online labor market. Section 3.3.3 contains our method and the treatment conditions. We discuss the results in Section 3.4; as we show there, our results demonstrate that our proposed design changes successfully curb rating inflation and lead to substantially more informative ratings.

3.3.1 Motivation and hypothesis

We aim to design rating scales for online platforms that lead to more informative ratings. Motivated in part by the emergence of the rating norms discussed in the introduction, where 5 stars is routinely considered “average,” we are interested in evaluating the effectiveness of changes that can counter this norm: in particular, we consider rating scales where the answer choices are *positive-skewed*, with *specific* descriptions attached to each rating.

Our hypothesis is that *such positive-skewed scales lead to less “inflated” ratings than standard, numeric rating scales, and as a result, produce more informative ratings.* (By “inflated”, we mean ratings where a large majority of the rating distribution is on the highest rating score).

This hypothesis is motivated by the idea that raters feel a cost if they are dishonest in their ratings, and that *this cost is an increasing function in how dishonest she perceives herself to be.* Crucially, this quantity would vary both with experience quality and the rating system design. With standard numeric rating systems and today’s norms, a rater arguably does not consider herself dishonest for rating mediocre experiences 5/5, because that is what 5 stars has come to mean. On the other hand, suppose a platform provides explicit guidance on what ratings mean (e.g., “5 stars means best experience you’ve had”); raters would thus face a higher cost of dishonesty for giving low quality sellers a high ratings. This hypothesis is consistent with the *self-concept maintenance* literature, where people are understood to be more likely to be dishonest when they can convince themselves that they are acting honestly (Mazar et al., 2008). Models with such costs have been considered in, e.g., Filippas et al. (2019) and Fradkin et al. (2018).

Finally, note that while our test design allows us to measure the effects of other design changes, we did not hypothesize any other specific effects (direction or magnitude) *a priori*, except for the benefits of positive-skewed rating scales. These alternate design changes let us compare the relative benefits of possible solutions to the rating inflation problem.

3.3.2 Empirical context

The test ran on a large, online labor market. In this market, clients seek the services of freelancers across a variety of categories (e.g., software development, graphic design, and translation). Clients

may choose to contract with a freelancer for a job based on work history, prior ratings, the freelancer’s proposal, and potentially an initial conversation. A client-freelancer pair may work on multiple jobs together during their time on the platform.

At the end of each job, the client is asked to fill out a feedback form in which they rate the freelancer’s work through a series of multiple choice and free-form questions. This labor market has both private and public ratings, and private ratings are aggregated and made available to potential future clients as part of a freelancer’s public score. Both private and public ratings are high on the platform: even the average private feedback score is over 8.5/10. See Filippas et al. (2019), which analyzes ratings over time on the same labor market, for an in-depth description of the status-quo rating system and its performance.

3.3.3 Method

We now describe our test method. The authors were involved in test design and analysis of anonymized data, but not implementation or deployment.

The test added a question to the feedback form given to clients after they close a job. This question appeared with the current private rating questions and was marked optional. All clients were still asked the existing private rating questions, including rating the freelancer on a numeric 0 – 10 scale. The answer choices were displayed vertically after the question.

The test ran over a 90 day period in Summer 2018, with a pilot in January 2018 over 5 days. We report the set-up and results of the long test; pilot results are nearly identical.

Treatment conditions

There were six treatment conditions that included an additional question on the feedback form. The question phrasing and answer choices differed between the treatment conditions. See Table 3.1 for a detailed list of the treatment conditions. There were four different types of answer choices: (1) comparing against a client’s expectations (*Expectation*); (2) descriptive adjectives (*Adjectives*); (3) comparing against the average freelancer the client has hired, as well as two variants (*Average; Average, not affect score; Average, randomized*); and (4) a numeric scale with no descriptions attached to the ratings (*Numeric*).

The non-numeric treatments describe possible ways to design multiple choice rating systems that add more specificity to the rating scale. The choices themselves are skewed toward the positive end: each scale has two “negative” choices, one “neutral” choice, and 3 “positive” choices, in increasing levels of effusiveness. This imbalance was chosen so that (a) clients could give “positive” feedback to most freelancers while still allowing the platform to disambiguate the very best from others, and (b) to emphasize that the best ratings should be reserved for the very best freelancers.

The *Numeric* treatment, giving freelancers the option of giving 0 – 5 stars, helps disambiguate between novelty effects of introducing new questions and the idiosyncratic effects of the question

Treatment	Additional Question	Answer choices
<i>Expectation</i>	How did this freelancer compare to your expectations?	Much worse than I expected, Worse than I expected, About what I expected, Better than I expected, Far better than I expected, Beyond what I could have expected
<i>Adjectives</i>	How would you rate this freelancer overall?	Terrible, Mediocre, Good, Great, Phenomenal, Best possible freelancer!
<i>Average</i>	How does this freelancer compare to others you have hired?	Worst Freelancer I've Hired, Below Average, Average, Above Average, Well Above Average, Best Freelancer I've Hired
<i>Average, not affect score</i>	How does this freelancer compare to others you have hired? (This will not impact the freelancer's score)	Same as Average group
<i>Average, randomized</i>	How does this freelancer compare to others you have hired?	Same as Average group, but in random order
<i>Numeric</i>	How would you rate this freelancer overall?	0, 1, 2, 3, 4, 5

Table 3.1: Treatments groups for labor market test

itself. As in the other treatments, this question is asked *in addition to* the the existing rating questions on the site, which include a 0 – 10 overall rating question. Furthermore, the question phrasing is identical in the *Adjectives* and *Numeric* treatments; only the answer choices differ. This design thus teases out the different effects of the type of question itself and the answer choices.

We include two additional variants as follows: (a) a variant with additional text emphasizing that the answer will not impact the freelancer's publicly displayed rating (*Average, not affect score*), and (b) a variant where we randomize the order of the answer choices (*Average, randomized*). The first variant tests the additional informational gain from clients knowing for certain that a low rating will not affect the freelancer. The second variant helps assess the propensity of clients to not read all the answer choices before responding.

In addition to the six treatments, a *Control* condition was included, in which no additional question is asked (replicating the status quo feedback form).

Allocation to treatment groups

Allocation was done at the client level when they first closed a job and landed on the feedback form after the start of the test. Clients who had closed less than two jobs in the past were excluded, as several of the treatment conditions ask clients to compare the freelancer to past experiences. Each treatment condition was allocated 15% of the clients, and the remaining 10% of clients were allocated to *Control*. After being allocated to a treatment group, a given client was assigned the same treatment for the duration of the test and was thus shown the same additional question for any further jobs she may have closed. (During the pilot in January, 2018, 40% of clients were allocated

Condition	Assigned	Submissions		Analyzed		Mean treatment response
	Clients	Clients	Jobs	Clients	Jobs	
Control	7576	7179	23554	6880	21850	-
Expectation	11271	10073	28880	9718	27156	3.34
Adjectives	11101	9966	28413	9616	26370	3.65
Average	11375	10135	28372	9807	26605	3.76
Average, not affect score	11500	10295	28882	9944	27536	3.78
Average, randomized	11466	10258	28663	9895	26978	3.46
Numeric	11303	10120	32153	9802	27677	4.59

Table 3.2: Number of clients and jobs in each cell, and mean treatment response

to *Control* and 10% to each treatment condition.)

Due to a bug in the allocation code during the test, 1,086 out of the 66,755 clients who submitted feedback were assigned to different treatment conditions on different closed jobs. We disregard all such clients in our analysis to eliminate the possibility of contamination between treatment cells. To confirm experimental validity, we show in the Appendix that otherwise the randomization was effective: the distribution of clients in different cells are essentially identical on all observed covariates. This bug does bias the client population in our data in one way, however: clients who closed more jobs in the test period were more likely to experience the bug, and thus to be incorrectly assigned to multiple treatment cells. As a consequence, the client population on which we carry out our analysis skews slightly away from the highest volume clients on the platform.

Number of responses and data preprocessing

75,592 unique clients landed on the feedback page, and 66,755 clients submitted feedback for at least one job. We remove the clients mistakenly assigned to multiple treatment cells (the bug described above), as well as seven clients who were correctly assigned but who closed more than 200 jobs during the test period. Table 3.2 contains, for each treatment cell, the numbers of clients assigned, clients who submitted a job, and clients and jobs in our dataset after the pre-processing.

3.4 Labor market test results

In this section we provide results that demonstrate that the positive-skewed verbal scales reduced inflation and produced more informative ratings. In Section 3.4.1 we show the verbal rating scales result in *deflated* ratings compared to the numeric scale; we report simple marginal distributions of the rating choices made by clients in different treatment cells, both overall and across time throughout the experiment. Then in Section 3.4.2 we show that such verbal scales are more *informative* than the numeric scale: the rating choices made by clients in the verbal treatment cells better correspond to exogenous signals of a given freelancers quality, in two ways. First, we show that the verbal scales are more predictive of whether clients tend to rehire a freelancer. Second, we show that the verbal

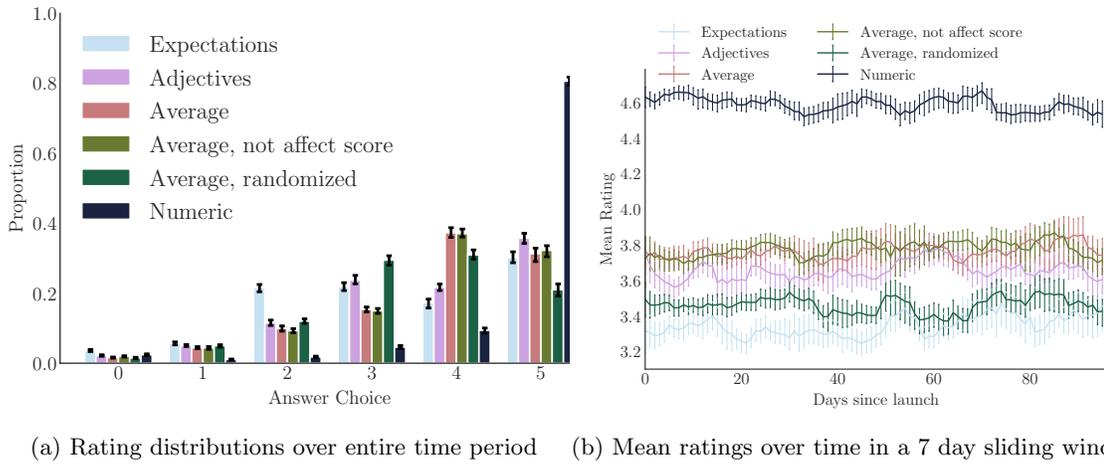


Figure 3.1: Labor market test marginal rating distributions. Error bars are 95% boot-strapped confidence intervals, where the bootstrapped sampling is done at the client level.

scales yield ratings that are more strongly correlated with the average rating a freelancer receives by distinct clients across treatment cells; this latter quantity serves as an empirical proxy for freelancer “quality”. The resulting approximate joint distributions of freelancer quality and ratings received motivate our model in the next section, where we develop a measure to compare rating scales by the joint distributions they induce.

3.4.1 Verbal rating scales counter inflation

We start our analysis of the results by looking at the marginal rating distributions in each treatment, i.e., how many freelancers received each possible rating in each treatment cell. These distributions provide evidence that the non-numeric scales provide more dispersed and deflated ratings. Furthermore, we find that the verbal rating scales are resistant to inflation throughout the course of the experiment; this surprising finding stands in contrast to prior work on rating inflation over time.

Snapshot analysis of ratings

Figure 3.1a shows the marginal rating distributions for each treatment group, and Table 3.2 contains the mean treatment response in each group, for the entire experiment period. There is a large and significant difference between the rating distribution from the numeric scale and each of the other treatment groups. Each treatment cell is different from each of the others at $p < 10^{-100}$ using the Kolmogorov-Smirnov two-sample test, except for the *Average* and *Average, not affect score* cells, where $p > 0.1$. While the *Numeric* treatment ratings follow the J-curve pattern usually seen in ratings, the other treatments are far more evenly distributed as desired. Most starkly, 80.6% of ratings on the *Numeric* scale are 5/5, while at most 35.8% of responses on any other scale received

the highest possible rating.

The substantial effect size of the difference between the *Numeric* condition and the other treatments confirms our hypothesis that specific and positive-skewed scales are an effective way to counter inflation: the answer choices presented to the rater are a first-order determinant of rating behavior. The other changes (emphasizing that the freelancer would not be affected, and randomizing the choices) have comparatively small effects.

Additional analyses are in the Appendix. In particular, our results there demonstrate that the findings reported in this section remain essentially identical even if we use other approaches to the analysis; for example, if we sample only one job per client, if we include all valid clients (i.e., including those with more than 200 jobs submitted), or if we even include the invalid incorrectly allocated clients.

Temporal analysis of ratings

The above analysis provides a *snapshot* view of what happens when a new question is added to the rating form. Some of the rating dispersion may be a novelty effect that decreases over time. As Filippas et al. (2019) emphasize, a substantial component of rating inflation in online platforms happens over time, on the order of months or even years. Here, we analyze whether ratings on the new questions inflated in the time period of the test.

We find that the rating scales do not inflate substantially. Figure 3.1b shows the average rating per treatment group over the 90 days after the launch of the test, in a sliding window of 7 days. There is no discernible inflation over time. It is instructive to compare the (lack of) inflationary trend to the inflation after the launch of a new numeric scale on the same platform in 2007, as reported by Filippas et al. (2019): average ratings inflated from about 3.8 stars to about 4.1 stars in the first three months after the system launched. (Note that introducing a new *Numeric* question in 2018 yields immediately inflated responses, suggesting that current platform users have been conditioned to the norm of inflated ratings.)

One concern with drawing conclusions from the preceding analysis over time is that there may not be enough clients who actually submit multiple jobs during the test period, and so novelty effects may still predominate when looking at overall averages. To study this concern, we analyze the ratings given by the clients who submitted at least 10 ratings each. We then run a regression for treatment response, with a covariate indicating how many previous jobs the client had submitted during the test period. Appendix Section B.1.4 has the associated table and discussion. For such high-volume clients, inflation exists but is slow: ratings may be inflated by a full point after a client has given 100 ratings.

Positively, this finding suggests that as long as new clients continue to enter the platform, ratings should remain deflated over a long time horizon. Indeed, given that existing norms are strongly biased towards inflationary ratings (as evidenced by clients' responses to the *Numeric* question), it

is quite valuable to see no evidence of inflation in the verbal treatment groups within a three month period. Of course, in principle it remains possible that over a timescale much longer than that of this test, norms would shift again towards inflated ratings. A longer-term longitudinal analysis of this type of inflationary behavior remains an important direction for future work in this area, though of course data collection over such a long time horizon is a significant obstacle.

3.4.2 Verbal rating scales yield more informative ratings

The analysis above establishes that buyers behave substantially *differently* with non-numeric rating scales than they do with the numeric scale, and in particular that such scales produce deflated ratings. In this section, we establish that this change is *beneficial* to the platform in terms of learning about freelancers: that higher “quality” freelancers indeed receive better ratings on average with the verbal scales, where “quality” is exogenously defined based on signals other than ratings on the given rating scale of interest.

To do this analysis, however, one needs such an exogenous signal on latent freelancer quality; such a strong signal is precisely what is missing on many online platforms with inflated rating systems. In fact, this lack of a signal, especially for new participants on the platform, is the primary motivation for our work. We provide two approaches to overcome this gap and show that indeed the verbal rating scales substantially provide more information to the platform. Our second approach, in particular, provides estimates for the joint distribution of freelancer quality and the ratings they receive in each scale.

Predicting freelancer rehires

First, we observe that on this labor market, clients often rehire the same freelancers for jobs in the future. Consistent with the literature, we assume that a client with a more positive experience with a freelancer is more likely to return to the platform and rehire the freelancer (Nosko and Tadelis, 2015). We thus analyze whether the verbal rating scores provide more predictive power on whether a freelancer will be rehired. (This measure is not perfect, as there are others reasons that a rehire may not occur, including that the freelancer does not wish to work with the client. However, the ratings in our test are private, and so could not have directly exerted this influence).

For each client-freelancer pair that completed a job during the experiment period, we consider the rating given by the client to the freelancer on the *first* such completed job. Across condition cells (besides *Control*), there are 125,386 such first jobs, with 58,787 unique clients and 110,798 unique freelancers. We then observe whether the client-freelancer pair completed another contract during the test time period.

The results imply that *the verbal rating scales are substantially more informative than the numeric scale* – even a *single* rating provides more predictive power to the platform. Figure 3.2 shows, for each condition, the likelihood that a freelancer given a certain rating is to be eventually rehired by

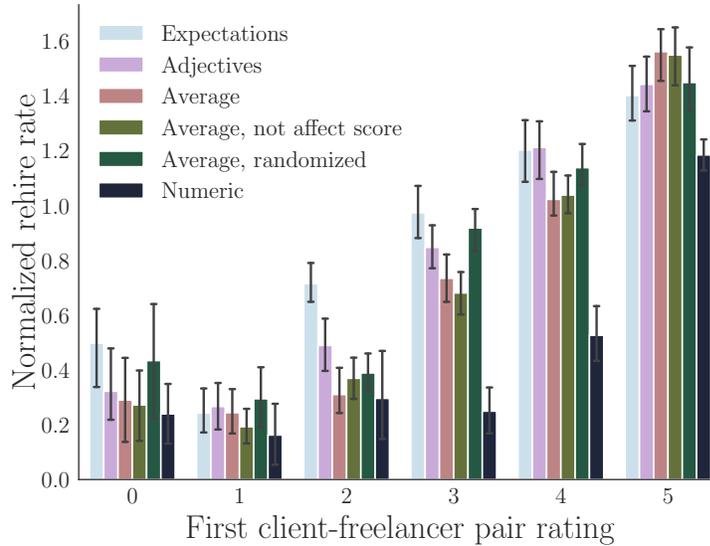


Figure 3.2: Likelihood that a client will rehire a freelancer during the time period of the test, given just the first rating the client gives that freelancer during the test period. Values are normalized by the overall mean rehire rate. Confidence intervals are 95% intervals with bootstrapped sampling done at the client level.

the same client, normalized by the overall mean rehire rate. Clients who gave a freelancer anything but a “5” on the numeric scale almost never rehired the freelancer. With the positive-skewed verbal scales, by contrast, there is a smoother decline of rehire rate, giving the platform finer-grained insight on whether a freelancer is likely to be rehired. Furthermore, top verbal rating scores better identify truly exceptional freelancers: for example, freelancers are 31.8% more likely to be rehired after receiving the top rating in the *Average* treatment than they are after receiving the top numeric score (1.56x and 1.18x higher than the average rehire rate, respectively). Clients are providing more information when asked to rate freelancers on the verbal scale.

Note, however, that rehiring data during the test period does not provide enough information to construct reliable quality estimates for individual freelancers: a given freelancer typically only matches with a few unique clients and the rehire decision is itself noisy. In the next sub-section, we construct such freelancer-level quality estimates by looking at freelancer ratings across cells.

Correlation with estimated freelancer quality

In this section, we estimate each freelancer’s quality and use this estimate to construct a joint distribution of estimated freelancer quality and ratings under a given scale. We then use this joint distribution to compare different designs.

Recall that in our experiment design, a given *client* is only in a single treatment cell throughout

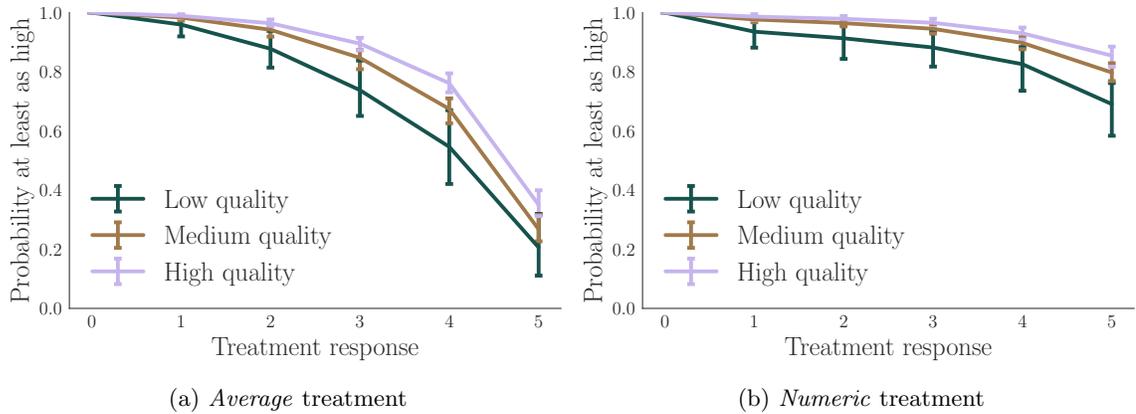


Figure 3.3: Joint distributions of freelancer quality vs. ratings in the *Average* and *Numeric* treatment cells, respectively. Low, Medium, and High quality sellers refer to those with other cell average ratings in $[0, 2)$, $[2.5, 3.5)$ and $[4.5, 5]$, respectively. The Y axis is the probability that a freelancer of a given quality receives a rating at least as high as the X axis. Confidence intervals are 95% intervals with bootstrapped sampling done at the client level.

the test period. On the other hand, a given *freelancer* may complete jobs with and receive ratings from clients across treatment cells. Thus the ratings any individual freelancer receives in different treatment cells are independent.

We can leverage this independence to construct approximate joint distributions of freelancer quality and ratings in each treatment cell as follows. For a given treatment cell, we consider all freelancers who received at least three ratings in the *other* treatment cells, and we estimate a freelancer’s quality via a simple average of these ratings. For each given treatment cell, these estimates of quality are exogenous with respect to the ratings received in that cell. For each treatment cell, we then construct a joint distribution over freelancers of the rating received in that cell, and the estimated quality of that freelancer.

We note that given the amount of data we had available, our estimates of these joint distributions are noisy. The freelancer quality estimates are only from about three ratings across the various treatment cells, and responses in the cells themselves differ in meaning. In practice, a platform with access to historical performance data across a longer time-period, especially for long-lived sellers, may be able to construct more reliable estimates.

Figure 3.3 includes two such joint distributions, for the *Average* and *Numeric* treatments, respectively. The Appendix contains the same joint distribution for the other treatment cells; we also show another way to group freelancers by their average ratings, and similar patterns emerge. In all treatment cells, higher quality freelancers receive better ratings, though to varying degrees.

In the *Numeric* cell, most freelancers receive high ratings independent of quality, and it may be difficult to distinguish high and medium quality freelancers. In contrast, in the *Average* cell there

is a larger gap between freelancers of different quality, and qualitatively one expects that this gap is beneficial in terms of learning freelancer quality. In this sense, the *Average* cell is providing ratings that are more informative than the *Numeric* cell.

3.4.3 Discussion

These results suggest that there are countervailing forces to ratings inflation that can induce ratings to be more dispersed than in existing systems, by shifting how buyers interpret the scale: a platform can find large improvements over standard rating systems by explicitly defining what each rating means and positive-skewing such descriptions. In particular, though ratings still tend positive in absolute terms in our verbal scales (over 80% of freelancers receive *Above Average* or better), clients seem hesitant to give most freelancers the best possible score when such a score is interpreted as truly exceptional. This deflationary effect has positive information implications for the platforms. For example, freelancers who receive such a rating are more likely to be rehired by the platform.

Furthermore, this large effect is first order and dwarfs other sources of rating variation on the labor market. For example, in the Appendix we show that rating heterogeneity across market segments is small, on the order of 0.1 differences in means. Similarly, the treatment with randomized answer choices reveals that clients tend to pick the first choice presented more than others, but again the effect is second order.

We conclude by noting that our qualitative assessment of the joint distribution of estimated quality and ratings in Section 3.4.2 is somewhat *ad hoc*. Motivated by this work, in the next section we develop a *quantitative* approach to capture the performance gain of verbal rating scales, based on the joint distribution of estimated quality and observed ratings. In particular, we compare rating system designs in terms of the *speed* at which they allow the platform to correctly rank the freelancers.

3.5 A framework to compare rating scales

The preceding section establishes, through a variety of metrics, that a platform can improve the information obtained through the rating system through careful choice of the descriptions for each level of a multi-level rating scale. This finding naturally prompts the question: is there a principled way to compare rating scale designs to find the one that is “best” for the platform? We now develop a framework to do so.

In particular, we take the perspective that the platform’s objective is to ensure that the ranking of sellers based on their aggregate rating score converges to the true ranking at the fastest rate possible in the number of ratings received. We develop a stylized model to formalize this notion and use it to develop an approach to compare and optimize rating systems. The stylized model we consider has the following key elements. We assume that *buyers* enter per time period and match with long-lived

sellers, potentially at varying rates according to the seller’s quality. After the match, the buyer rates the seller; the rating behavior depends on the rating scale (answer choices, e.g., the adjectives or other answer phrasings in Table 3.1). The platform’s design levers are the answer choices making up the rating scale, and the *scores* it attaches to those adjectives. We leverage this stylized model to propose an approach to maximize the rate of convergence (in a large deviations sense) of the estimated ranking based on sellers’ aggregate scores, to the true underlying ranking based on sellers’ qualities. We apply this methodology to our labor market data (presented in Section 3.5.3), and to a synthetic dataset collected through Amazon Mechanical Turk (presented in the Appendix).

3.5.1 Model

Our model is constructed to emphasize the platform’s learning rate of participants through its rating system. It is deliberately stylized so that we can derive a relatively straightforward method to compare and optimize rating scales. The key components are as follows.

Time. Time is discrete: $k = 0, 1, 2, \dots$

Sellers. The system consists of a unit mass of sellers, each associated with a quality *type* θ , which is (initially) unknown to the platform. We assume θ is drawn independently and uniformly at random from a finite and totally ordered set Θ , with $|\Theta| = M$. We use θ_i to denote the i th element of Θ within this order, for $0 \leq i < M$.

In addition, each seller has an *aggregate score*, described further below; we let $x_k(\theta)$ denote the aggregate reputation score of the seller of type θ at time k .

Rating accumulation. Sellers accumulate ratings over time by matching with buyers. At each time step, each seller matches with at most a single buyer. We make one key assumption that drives the accumulation of ratings: in particular, that sellers of higher quality are more likely to be matched. We consider an analysis that is asymptotic in the number of ratings received by sellers and so we model this visibility benefit by assuming that *sellers of higher quality accumulate ratings at a faster rate*. In particular, we assume the existence of a nondecreasing *match function* $g(\theta)$, where a seller of type θ receives $n_k(\theta) = \lfloor kg(\theta) \rfloor$ matches, and thus ratings, up to time k .

Our approach to modeling rating accumulation is stylized in at least two important ways. First, the matching function is artificial: in general, sellers are more likely to match when they have a higher *observed* aggregate score, and there may be other heterogeneity as well. Second, we suppose all sellers have the same age: at time k , all sellers have had k opportunities to match with buyers. In reality, of course, sellers have different ages on a marketplace. These choices allow us to develop a clean approach to optimizing the learning rate; we discuss the consequences further in our empirical investigation in Section 3.5.3.

Ratings. How are sellers rated? After each match, the seller receives a rating in the form of a multiple choice question answered by the buyer. The platform makes two decisions at the beginning when designing this question. First, the platform chooses a rating scale Y , composed of an ordered

set of answer choices $y \in Y$ from which the buyer will choose. Second, whenever a seller receives a rating $y \in Y$, the platform gives the seller a *score* $\phi(y) \in [0, 1]$ depending only on the rating received. The score represents the relative positivity assigned to a rating y : high scores positively affect the seller's aggregate score (as we formally describe below). Platforms often use equally spaced scores when translating rater's choices to an aggregate score (e.g., the choice "5 stars" translates to a numeric 5 when averaging, the choice "4 stars" translates to a numeric 4 when averaging, etc.), but we allow the possibility that this choice should also be optimized.

At each rating opportunity (i.e., match made), the seller receives a rating from the set Y , and we assume that this rating depends only on the true quality of the seller. In particular, we presume that, given scale Y , the probability a seller of type θ receives a rating y is $\rho(\theta, y|Y)$, with corresponding cumulative mass function $R(\theta, y|Y)$ reflecting the probability a seller of type θ receives a rating y or higher. In other words, the scale Y induces a joint distribution between the underlying seller quality and the rating choices buyers make. We make the natural assumptions that $R(\theta, y|Y)$ is strictly increasing in θ and strictly decreasing with y .

Let $y_0(\theta), y_1(\theta), y_2(\theta), \dots$ be the sequence of ratings received by the seller of type θ . The *aggregate score* up to time k of this seller is the average score from ratings received:

$$x_k(\theta) = \frac{1}{n_k(\theta)} \sum_{\ell=0}^{n_k(\theta)} \phi(y_\ell(\theta)). \quad (3.1)$$

(We presume $x_0(\theta) = 0$ for all θ .) Since $\phi(y) \in [0, 1]$ for all y , the score x_k also lies in $[0, 1]$.

This rating behavior is also a strong assumption. In particular, it does not capture heterogeneity across raters (the types of sellers a buyer matches with may correlate with the buyer's rating behavior in general). Including such heterogeneity is a direction for future work, and we discuss it further in the conclusion Section 3.6; indeed, empirical identification of such heterogeneity presents an interesting practical challenge.

System state. We represent the state of the system defined above by a joint distribution $\mu_k(\Theta, X)$, which gives the mass of sellers of type $\theta \in \Theta$ with aggregate score $x_k(\theta) \in X$ at time k . Throughout our model presentation, we describe the system model as one emerging from interactions between individual buyers and sellers. However, we assume a unit mass of sellers (and some mass of buyers), and so all such descriptions should be viewed as illuminating the evolution of a joint distribution $\mu_k(\Theta, X)$ of the types of sellers on the platform and their current scores. To formally describe the evolution of μ_k , let $E_k = \{\theta : n_k(\theta) = n_{k-1}(\theta) + 1\}$. These are the sellers who receive an additional rating at time k ; for all $\theta \in E_k^c$, $n_k(\theta) = n_{k-1}(\theta)$. Next, for each $x, x' \in [0, 1]$, define $\chi_k(x, x', \theta|Y, \phi)$ as:

$$\chi_k(x, x', \theta|Y, \phi) = \{y : n_k(\theta)x - n_{k-1}(\theta)x' = \phi(y)\}.$$

The set χ describes the rating(s) a seller of type θ at time k with aggregate score x' can receive to

transition to aggregate score x . We then have:

$$\mu_{k+1}(\Theta, X) = \int_{E_k} \int_0^1 \int_X \sum_{y \in \mathcal{X}_k(\theta, x, x' | Y, \phi)} \rho(\theta, y | Y) dx \mu_k(d\theta, dx') + \int_{E_k^c} \int_X \mu_k(d\theta, dx').$$

It is straightforward but tedious to check that the preceding dynamics are well defined, given our primitives.

Platform objective. We assume that the platform wants the ranking of sellers by observed aggregate score to reflect the underlying true quality ranking as closely as possible.

Formally, given $\theta_1 > \theta_2$, define $P_k(\theta_1, \theta_2)$ as follows:

$$P_k(\theta_1, \theta_2) = \mu_k(x_k(\theta_1) > x_k(\theta_2) | \theta_1, \theta_2) - \mu_k(x_k(\theta_1) < x_k(\theta_2) | \theta_1, \theta_2). \quad (3.2)$$

This expression captures the “errors” made by the ranking according to observed score. In particular, when $\theta_1 > \theta_2$ but $x_k(\theta_1) < x_k(\theta_2)$, the aggregate score ranking swaps the ordering of sellers θ_1 and θ_2 . Thus, a good rating system has large $P_k(\theta_1, \theta_2)$.

We consider the problem of maximizing the following objective, a scaled version of *Kendall’s τ rank correlation* between the estimated ranking of sellers and the true ranking:

$$W_k = \frac{2}{M(M-1)} \sum_{\theta_1 > \theta_2 \in \Theta} P_k(\theta_1, \theta_2) \quad (3.3)$$

The coefficient ensures that W_k remains bounded even as M increases. This objective depends on the model primitives R (rater behavior) and g (matching rates), as well as the platform’s decisions Y (levels) and ϕ (score).

We note that, in this model, the goal of the rating system is to accurately rank sellers by quality. Another approach may be to directly optimize for total platform revenue or aggregate welfare. This approach would require primarily optimizing which matches occur, a focus of many other works. We optimize information gained *per* match, for which finding the true ranking of sellers is a reasonable objective. One observation in support of this choice is that the “deliverable” for the ratings team in an online platform company is typically an accurate rating that can be an input to models used by other teams throughout the organization. Further, in a model where matching rates are exogenously determined by quality, we conjecture that optimizing other objectives (accuracy and revenue) would produce qualitatively similar results.

Learning $R(\theta, y | Y)$. We note that our approach to quantify the learning rate, which we develop in the next subsection, requires the platform to learn the rating joint distribution $R(\theta, y | Y)$ for various potential rating scales Y . In our analysis of the labor market experiment, we provide one approach to do so — using the data collected during the experiment itself to estimate seller qualities θ . In practice, a platform with access to more historical data may rely on estimates of θ for a group

of “known” sellers; e.g., these may be long-lived sellers on the platform. The platform can then test new rating scales, and use the resulting data to estimate R .

3.5.2 Quantifying design performance via convergence rate

As noted above, the platform has two design choices it makes: the set of rating levels Y , and the score function ϕ . We now consider an approximate approach to maximization of the objective W_k , by appropriate choice of Y and ϕ .

No single choice of Y and ϕ can simultaneously optimize W_k for all k : some designs may be effective in separating the best sellers from the worst quickly, but then never separate all sellers. Further, as long as $\phi(y)$ is strictly increasing, then because $R(\theta, y)$ is strictly increasing in θ , we have, for all $\theta_1 \neq \theta_2$, and all choices of Y and ϕ : $\lim_{k \rightarrow \infty} P_k(\theta_1, \theta_2) = 1$. Using the bounded convergence theorem we conclude that $\lim_{k \rightarrow \infty} W_k = 1$, independent of the design choice Y and ϕ . Thus *any* design asymptotically – with enough ratings – recovers the true ranking of sellers.

For these reasons, we focus on maximization of the *rate* at which W_k converges; we call the design (Y, ϕ) that maximizes this rate *optimal*. We use a *large deviations* approach to study the rate of convergence (Dembo and Zeitouni, 2010), following other works that adopt this approach (Garg and Johari, 2019a; Glynn and Juneja, 2004).

We have the following result.

Theorem 3.5.1.

$$r \triangleq - \lim_{k \rightarrow \infty} \frac{1}{k} \log(1 - W_k) = \min_{0 \leq i < M} \inf_{a \in \mathbb{R}} \{g(\theta_{i+1})I(a|\theta_{i+1}) + g(\theta_i)I(a|\theta_i)\} \quad (3.4)$$

where $I(a|\theta) = \sup_z \{za - \Lambda(z|\theta)\}$, and $\Lambda(z|\theta) = \log \sum_{y \in Y} \rho(\theta, y|Y) \exp(z\phi(y))$ is the log moment generating function of a single rating given to seller of type θ .

The proof follows from standard results in large deviations analysis and is in the Appendix.

The expression in (3.4) is called the *large deviations rate* for \bar{W}_k . The theorem shows that $W_k(\theta_1, \theta_2) \rightarrow 1$ exponentially fast, and provides an explicit relationship between our choice of Y and ϕ , and the corresponding exponent. In other words, $1 - W_k = \mathcal{O}(e^{-rk} \text{poly}(k))$.

Two rating systems can be compared by their respective learning rates: for each design, simply calculate their rates and then compare. The rate function can be calculated numerically given $R(\cdot|Y)$, ϕ and $g(\theta)$: in particular, observe that $\sup_z \{za - \Lambda(z|\theta)\}$ is a concave maximization problem in z , and $\inf_{a \in \mathbb{R}} \{g(\theta_{i+1})I(a|\theta_{i+1}) + g(\theta_i)I(a|\theta_i)\}$ is a convex minimization problem in a .

Our design optimization problem is thus as follows: *choose Y and ϕ to maximize the large deviations rate r in (3.4)*. We suggest the following approach, supposing the platform has a collection of candidate scales $\{Y_p\}$. First, experiment with each scale Y_p and estimate $R(\theta, y|Y_p)$ and $g(\theta)$. Then use the following brute force approach to optimization: for each Y_p , choose a random, increasing set

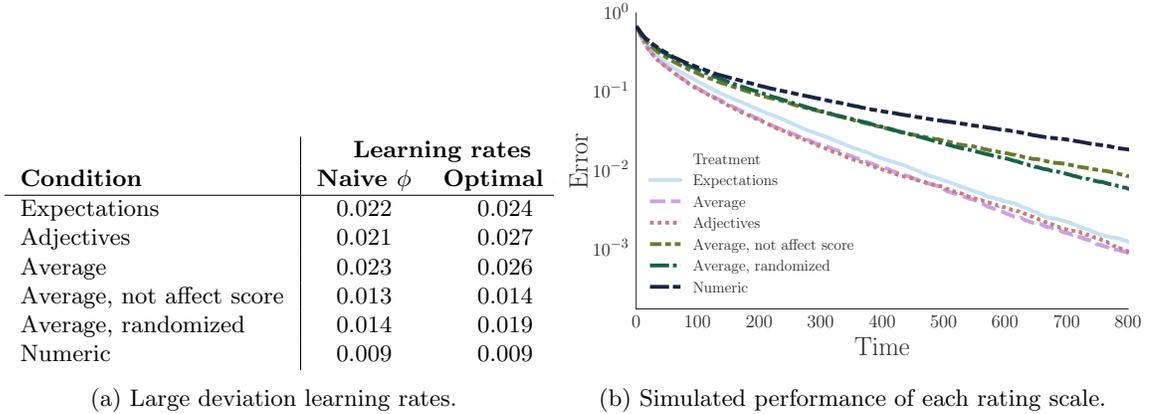


Figure 3.4: We apply and test our design approach using experimental data from our online labor market. Large deviation rates are calculated using Equation (3.4) and the joint distributions generated in Section 3.4.2. Optimal for each treatment corresponds to the highest learning rate among many random scores.

of scores $\phi(y) \in [0, 1], \forall y \in Y_p$ in each iteration, and calculate the learning rate. For each candidate scale Y_p , run a large (exponential in $|Y_p|$) number of such iterations. Finally, choose the design Y_p, ϕ with the best learning rate. While this is a brute force optimization, we envision a platform will not be changing the rating system design very frequently, and thus computation time is not critical.

Qualitatively, $\phi(y)$ should be large if higher quality freelancers are much more likely to receive a rating choice of at least y than are lower quality freelancers. If the rating joint distribution $R(\theta, y|Y)$ is such that no such choice y exists, then the scale Y will perform poorly in separating the sellers. In particular, rating scales with inflated responses are uninformative for this reason: every seller independent of quality is likely to receive the most positive rating choice, and so nothing separates low quality from high quality sellers. Similarly, a ϕ that does not reward sellers for receiving rare, positive ratings, is suboptimal.

Finally, we note that the two parts of the design – scale Y and score mapping ϕ – differ in their visibility to raters. Scale Y is presented to raters in order to induce a certain desirable rating behavior $R(\cdot|Y)$. On the other hand, a platform need not share the mapping ϕ , which is simply a technical tool that maximally leverages aggregate rating behavior to form an internal ranking of sellers. The platform may then choose to share statistics about sellers to buyers, for example whether a seller is in the top 25%. The optimal information sharing procedure is a question tackled by other work (Acemoglu et al., 2017; Ifrach et al., 2017; Papanastasiou et al., 2017).

3.5.3 Application to the online labor market

We now follow the design approach outlined above using the empirical $R(\cdot|Y_p)$ calculated in Section 3.4.2 for each scale Y_p in the test. For simplicity, we assume a uniform search rate $g(\theta) = 1$ for

all θ ; the simulation results are robust to this choice.

Large deviation learning rates for each design

First, we calculate the large deviation rates for each treatment scale, assuming equally spaced scores $\phi = \{0, 1, 2, 3, 4, 5\}$. All of the verbal treatments have larger learning rates than the *Numeric* treatment, as shown in Figure 3.4a.

Next, we optimize the scores ϕ for each of the scales. Figure 3.4a also contains the learning rate achieved by the corresponding optimal score function for each treatment. It suggests that picking the correct labels on the scale is the first order determinant of the rating system's performance, while the optimal scores are second order.

The optimal scores themselves (in Appendix Table B.3) reflect the corresponding joint distributions. For example in the *Numeric* scale, only the frequency of receiving the top rating distinguishes freelancers; thus freelancers receive a lesser score (3.45/5, versus $\approx 4/5$) for the second-highest rating in that scale versus in the verbal rating scales.

Note that perhaps because our estimation procedure on the joint distribution is noisy, the *Average* and *Average, not affect score* treatments differ in their joint distributions and learning rates, even though they have identical marginal distributions and rehire rates.

Simulated market performance of each design

Finally, we simulate a market for each of the treatment conditions as follows, in order to compare how the scales perform.

In our simulation, there are 500 sellers with i.i.d. quality in $\{Low, Medium, High\}$. There are 100 buyers, each of which matches uniformly at random to a *unique* seller per time period. In other words, matching is not independent across sellers, and each seller can only match once per time period; each seller matches approximately once every 5 time periods. The buyers rate the sellers according to the joint distributions calculated in Section 3.4.2. Ratings are converted to scores according to the optimal score function for each treatment. All sellers enter the market at time $k = 0$ and do not leave. After each time period, the sellers are ranked according to their average scores. The true ranking of sellers (i.e., $Low < Medium < High$) is also constructed. We then calculate the Kendall's τ rank distance (not counting sellers tied according to true quality) between the two lists.

Figure 3.4b shows the mean (across many simulations) ranking errors over time for each treatment system as described. The plot and corresponding learning rates for each treatment demonstrate that even though large deviations rates are an asymptotic quantity, they effectively predict the performance of each rating scale even for small horizons. The *Numeric* treatment in particular learns the ranking of sellers at a much slower rate than do the other mechanisms, both in terms of learning rate and simulated performance.

In Appendix Figure B.5, we show other simulations and analyses as robustness checks. First, we show performance over time when each seller independently leaves the market with probability .01 at the end of each time k , with a new seller with no reputation score taking her place; such entry and exit does not affect the comparative performance of each rating scale. Robustness to such entry and exist further suggests that designed scales will outperform others when only sellers’ most recent ratings are used, in order to facilitate and reflect seller improvement, cf. (Aperjis and Johari, 2010).

Next, we compare learning with equally spaced vs. optimal ϕ ; as suggested by the learning rates, calculating an optimal ϕ has a small but noticeable effect in performance.

Finally in Appendix B.2, we repeat the analysis for a synthetic setting on Mechanical Turk that demonstrates the utility of our methods for survey contexts beyond ratings on online platforms. We find that superficially similar scales may perform dramatically differently in a way that is not a priori knowable before conducting an experiment and calculating learning rates. We evaluate performance of each treatment scale on new data not used for scale optimization and find that performance improvements can transfer to a deployment.

3.6 Conclusion and discussion

In this work, we study the the design of informative rating systems. We demonstrate through a field test on a large online labor platform that there can be substantial benefit to changing answer choices and question phrasing in a rating scale. In particular, we observe that (1) it is possible to choose a design of the verbal descriptions attached to answer choices present in the rating system that lead to deflated ratings, and (2) that these ratings are much more informative than ratings obtained in standard numeric rating systems. Motivated by this finding, we develop a technical framework to compare and design the scales by properly choosing the answer choices available to raters and the mapping of these choices to scores. We show that applying this framework can lead to designs that appear to substantially outperform *ad hoc* choice of the rating scale. We believe this work provides a foundation for a much more systematic approach to the design of rating systems, and that it has direct practical guidance for platforms to build more informative systems.

3.6.1 Challenges, opportunities, and limitations

Fraud in online reviews and ratings Our results establish that verbal rating scales can effectively counter *behavioral* norms and implicit pressures to provide maximally positive ratings. However, such scales do not *constrain* rater behavior and thus are ineffective against inflation caused by ratings fraud, in which the seller may fake transactions and rate themselves. There is a large literature on the prevalence of such fraud and techniques to detect it (Akoglu et al., 2013; Hooi et al., 2016; Hu et al., 2011; Luca and Zervas, 2016; Zhang et al., 2013). Our work is complementary to such approaches and is most appropriate for markets where such fraud is not the first order determinant

of rating inflation, such as on the labor market in question (as evidenced by the informative verbal scales).

Horizontal vs. vertical differentiation In many markets, buyers have heterogeneous preferences over sellers, i.e., there is *horizontal* differentiation. Our work assumes that there is an underlying ranking of sellers, i.e., that sellers are *vertically* differentiated (at least among the buyers who match with a given seller). If the matching process segments the market, then vertical differentiation may dominate within each segment. For example, price and location may segment the market on AirBnB such that only consumers with similar preferences match with and rate a given host. Then, a single rating scale can be used if rating behavior is similar across segments (recall that we show in Section 3.4.3 that scales perform similarly across segments in our labor market). In markets with substantial horizontal differentiation (even given that a buyer and seller have matched), however, the methods in this work can be used either (1) with comparatively objective questions (e.g., rating cleanliness or timeliness), where there may be vertical differentiation; or (2) alongside techniques that detect heterogeneous preferences and create “virtual” market segments (when feasible). In particular, our work is applicable wherever rating scales are used under the assumption of some degree of vertical differentiation.

International markets One potential difficulty in implementing verbal rating scales is that they must be designed in each language, and people in different cultures may interpret the same scale differently. This difficulty is especially acute as modern online platforms often operate globally. We note that verbal scales provide an opportunity as well as a challenge. There is variation across cultures in numeric rating systems, both for response scales in general and for online platforms in particular (Chen et al., 1995; Hamamura et al., 2008; Koh et al., 2010; Wang et al., 2015). In the status quo, the platform is left without a mechanism through which it can equalize the rating distributions. On the other hand, with verbal rating scales, if comparable ratings across regions are important, the platform can choose scales for each region that provide comparable rating distributions.

Using ratings for search and matching Another potential concern is that at the moment the answers to these questions are not used on the platform for other functions, such as search or matching. As illustrated by Filippas et al. (2019), some inflation for private questions is to be expected once the answers start affecting freelancers, even if freelancers cannot directly identify the client who provided any specific rating (e.g., if freelancers start asking for higher ratings on this question). We cannot completely eliminate this concern and leave the question for future work after a treatment condition is chosen to be implemented permanently on the platform. However, note that the marginal rating distributions and relation to how often freelancers are rehired by a client are extremely similar for the *Average*, and *Average, not affect score* conditions: either the clients already

are aware that the question they are answering is a test question that will not affect freelancers, or this additional information does not substantially influence how clients rate beyond the deflating effects of the answer choices in question.

Switching to a new rating system One final practical concern with introducing a new rating system with drastically different behavior is that it may be challenging from a data integrity perspective: how can old, inflated ratings be compared to the new ratings, and how can models throughout the platform be adjusted to handle both types of ratings? In some settings, such as our large online labor market, the new system can simply co-exist with the status quo: multiple questions can be asked in the rating form until enough time has passed with the new system such that older, inflated data is no longer useful. This approach adds friction in the form of additional work for clients, but it may be a price worth temporarily paying for finer resolution information. On other platforms where typically users are only asked one question, the transition may be more challenging. However, such platforms have begun experimenting with their rating systems. Furthermore, ratings data typically grows stale, as sellers enter and exit the platform or improve over time, and platforms often only use the last few ratings given to each seller (Aperjis and Johari, 2010). Such factors mitigate the cost of switching to a new system.

3.6.2 Future work

Platform goals Rating systems should reflect the specific goals and context of a platform. On some platforms, it may be undesirable to attempt to fully recover the ranking of sellers. For example, platforms that provide a commodified experience (e.g., ridesharing or delivery services) may only care about identifying bad actors on the platform. In this setting for example, asking buyers to rate sellers against the “average” may place undesirable, excessive pressure on sellers to attempt to distinguish themselves. Rather, the platform should potentially encourage raters to give good ratings unless something truly bad happened. Platforms in practice already do this; for example, when a passenger rates a driver 4 stars out of 5, Lyft describes the choice as “OK, could have been better.”

The methods in this work are most appropriate in settings where true differentiation exists between items or sellers (whether this differentiation is under the control of sellers or not), and it is desirable to identify and encourage comparatively high performers. Future work should closely examine the practical and theoretical relationships between a platform’s informational goals and its rating system design. We take a theoretical step in this direction in our work on designing binary rating systems (Garg and Johari, 2019a).

Dynamic design and combating inflation over time Even with our non-inflated rating scales, it may be possible that over time norms shift so that again ratings become inflated. In this event,

optimization of comparison points and rating scales may need to be a *dynamic process* for a platform. An important direction for future research is to consider a dynamic equilibrium view of rating system design. In particular, online marketplaces and platforms should aim to design systems that are naturally robust to inflation yet provide an good user experience. A complete picture should consider how search, buyer rating behavior, and seller behavior may change in response to changes in the rating system. Capturing these short- and long-run equilibrium effects remain important challenges. We believe our work provides an important empirical and theoretical building block in this direction, by suggesting that the meaning raters attach to levels of a scale can substantially influence the quality of information obtained by the platform.

Chapter 4

Designing Optimal Binary Rating Systems

4.1 Introduction

As discussed in the previous chapter, rating and ranking systems are everywhere, from online marketplaces (e.g., 5 star systems where buyers and sellers rate each other) to video platforms (e.g., thumbs up/down systems on YouTube and Netflix). However, they are uninformative in practice (Nosko and Tadelis, 2015). One recurring pattern is that ratings *binarize* – most raters only use the extreme choices on the rating scale, and the vast majority of ratings receive the best possible rating. For example, 75% of reviews on Airbnb receive a perfect rating of 5 stars (Fradkin et al., 2018). Furthermore, several platforms have adopted a binary rating system, in which a user rates her experience as either positive or negative. Given the prevalence of binary feedback (either de facto or by design), in this work we investigate the optimal design of such *binary* rating systems so that the platform can learn as fast as possible about the items being rated.

The marketplace model is similar to that of Chapter 3, except that the rater answers a binary question (as opposed to a multiple choice question with more than 2 choices). This simplification allows for far more theoretical insight on optimal design of rating systems, e.g., allows a characterization of how the best possible design changes with the informational goals of a platform.

This chapter is joint with Ramesh Johari. Published at the International Conference on Artificial Intelligence and Statistics (AISTATS), in 2019 (Garg and Johari, 2019a). We thank Michael Bernstein and participants of the Market Design workshop at EC'18. This work was funded in part by the Stanford Cyber Initiative, the National Science Foundation Graduate Research Fellowship grant DGE-114747, and the Office of Naval Research grant N00014-15-1-2786.

In particular, the rating pipeline studied in this chapter is as follows: A buyer enters a platform and *matches* with an item (e.g. selects a video on Youtube, is paired with a driver on Uber, or selects a home on AirBnB). She has an experience (e.g. a view, ride, or stay). Then, the platform asks her to *rate* her experience, i.e. it asks her a question. In a binary system, she indicates whether her experience was positive or negative. She then leaves. The platform uses the ratings it has received to score the quality of items, potentially showing such scores to future buyers.

By *designing* such a system, we mean: the platform can influence how the buyer rates – how likely she is to give a positive rating, conditional on the quality of her experience. It can do so by asking her different questions, e.g. “Was this experience above average” or “Was this experience the worst you’ve ever had?”. Different questions shift the probabilities at which items of various qualities receive positive ratings.

Our first question is: *what is the structure of optimal binary feedback?* A rating system in which every buyer gives positive ratings after each match, independent of item quality, will fail to learn anything about the items. Clearly, better items should be more likely to receive positive ratings than worse ones. But how much more likely?

Informally, suppose we have a set of items that match with buyers over time (at potentially differing rates), and we wish to rank the items by their true quality $\theta_i \in [0, 1]$. The platform cannot observe θ_i , however. Rather, in our model, after each match, an item with quality θ_i receives a positive rating with probability $\beta(\theta_i)$, and negative otherwise. In other words, the platform observes, for each item i , a sequence of ratings that are each Bernoulli($\beta(\theta_i)$). Such ratings are the only knowledge the platform has about items. The platform ranks the items according to the percentage of its ratings (samples) that were positive. The function $\beta : [0, 1] \mapsto [0, 1]$ affects how quickly the platform learns the true ranking, and it prefers to maximize the learning rate. We show how to calculate an optimal β .

As an example, consider three items with qualities $\theta_a > \theta_b > \theta_c$, and β such that $\beta(\theta_a) = 0.5$ and $\beta(\theta_c) = 0.1$, i.e. item a gets positive ratings after 50% of its matches, and item c after 10% of its matches. It is unclear what $\beta(\theta_b)$ should be. Trivially, $.1 < \beta(\theta_b) < .5$. Otherwise, even with infinitely many ratings the items will be mis-ranked.

But can we be more precise? If $\beta(\theta_b) = .49$, it will take many ratings of both items a and b to learn that $\theta_a > \theta_b$, but only a few from c to learn that $\theta_c < \theta_b$. That may be good if the platform wants to identify the worst item, but not if it wants to identify the best. It may also be fine if items a and b match much more often with buyers than item c . Clearly, the optimal value for $\beta(\theta_b)$ is objective and context dependent. Of course, the problem becomes more challenging with more items i for which $\beta(\theta_i)$ must be chosen. Lastly, in this example, one might intuitively think $\beta(\theta_b) = 0.3$ is optimal by symmetry when the items matter equally and matching rates are identical. This guess is incorrect. The optimal is $\beta(\theta_b) \approx 0.28$, due to the nature of binomial variance.

In this work, we first formalize the above problem and show how to find an optimal $\beta(\theta)$, jointly

for a set of items $[0, 1]$. β changes with the platform’s objective and underlying item matching rates. Jumping ahead, Figure 4.1 shows optimal β in various settings under our model. For a platform that wants to find the worst sellers, for example, the top half of items should each get positive ratings at least 80 + % of the time; it is more important for the bottom half of items to be separated from one another, i.e. get positive ratings at differing percentages.

Once we have calculated the optimal rating function β (given context on the platform goals and matching rates), what should we do with it?

Our second question is: *How does a platform build a rating system such that buyers behave near-optimally, i.e. according to a calculated β ?* The platform cannot directly control buyer rating behavior. Rather, it has to ask *questions* such that, for each item quality θ , a fraction $\beta(\theta)$ of raters will give the item a positive rating. For example, by asking, “Is this the best experience you’ve had,” the platform would induce behavior such that $\beta(\theta)$ is small for most θ . Most platforms today ask vague questions (e.g. thumbs up/down), and items mostly get positive ratings. We show this is highly suboptimal for ranking items quickly.

Our main contributions and paper outline are:

Rating system design as information maximization. In Sections 4.3.1-4.3.2, we formulate the design of rating systems as an information maximization problem. In particular, a good rating system recovers the true ranking over items, and converge quickly in the number of ratings.

Computing an optimal rating feedback function β . In Section 4.3.3, we develop an efficient algorithm that calculates the optimal rating function β , which depends on matching rates and the platform objective. The optimal β provides quantitative insights and principled comparisons between designs.

Real-world system design. In Section 4.4, we show how a platform can use a simple experiment and existing data to empirically design a near-optimal rating system, and to audit the current system. In Section 4.5, we demonstrate the value of this approach through an experiment on Mechanical Turk.

4.2 Related work

Many empirical and model-based works document and tackle challenges in existing rating systems, many of which were discussed in Chapter 3 (Bolton et al., 2013; Cabral and Hortacsu, 2010; Cook, 2015; Filippas et al., 2019; Fradkin et al., 2018; Gaikwad et al., 2016; Hu et al., 2009; Immorlica et al., 2010; Nosko and Tadelis, 2015; Rajaraman, 2009; Tadelis, 2016; Zervas et al., 2015). To our knowledge, we are the first to formalize a rating system design problem and then show how one can use empirical data to optimize such systems.

Other works also optimize platform learning rates (Acemoglu et al., 2017; Besbes and Scarsini, 2018; Che and Horner, 2015; Ifrach et al., 2017; Johari et al., 2017; Papanastasiou et al., 2017).

When prescriptive, they modify *which matches occur*, while we view the matching process as given and modify *the rating system*. The solutions are complementary.

Many bandits works also seek to rank items from a sequence of observations (Katariya et al., 2016; Maes et al., 2011; Radlinski et al., 2008; Yue and Joachims, 2009). Our problem is the *inverse* of the bandit setting: given an arm-pulling policy, we design each arm’s feedback.¹ Our specific theoretical framework is similar to that of Glynn and Juneja (2004), who optimize a large deviations rate to derive an arm-pulling policy for best arm identification.

The “twenty questions” interpretation of Shannon entropy (Cover and Thomas, 2012; Dagan et al., 2017) seeks questions that can identify an item from its distribution. Dagan et al. (2017) show how to almost match the performance of Huffman codes with only comparison and equality questions. Our work differs in two key respects: first, we seek to rank a set of items as opposed to identifying a single item; second, we consider non-adaptive policies (i.e. the platform cannot change its rating form in response to what it knows about an item already).

4.3 Model and optimization

We now formalize our model and show how to optimize the rating function to maximize the learning rate. We focus on finding an optimal $\beta : [0, 1] \mapsto [0, 1]$, a map from item quality θ to the probability it should receive a positive rating. This section requires no data: we characterize the *optimal* system.

4.3.1 Model and problem specification

Our model is constructed to emphasize the rating system’s learning rate. Time is discrete ($k = 0, 1, 2, \dots$). Informally: there is a set of items. Each time step, buyers match with the items and leave a rating according to $\beta(\theta)$. The platform records the ratings and ranks the items. Formally:

Items. The system consists of a set $[0, 1]$ of items, where each item is associated with a unique (but unknown) quality $\theta \in [0, 1]$; i.e., the system consists of a *continuum* of a unit mass of items whose unknown qualities are uniform² in $[0, 1]$. Below, we *discretize* the continuous quality space $[0, 1]$ into M types, to calculate a stepwise increasing β . We will make clear why we introduce a continuum but then discretize.

Matching with buyers. Items accumulate ratings over time by matching with buyers. We assume the existence of a nondecreasing *match function* $g(\theta)$, where item θ receives $n_k(\theta) = \lfloor kg(\theta) \rfloor$ matches, and thus ratings, up to time k . In other words, item θ is matched approximately every $\frac{1}{g(\theta)}$ time steps. $g(\theta) \leq 1$ and bounded away from 0, i.e. $\exists c > 0: g(\theta) > c$. This accumulation captures the feature that better items may be more likely to match.

¹Note that a *rating* is not the same as a *reward*; buyers often give positive ratings after bad experiences.

²Any distribution can be handled by considering θ to be the item’s *quantile* rather than its absolute quality.

Ratings. The key quantity for our subsequent analysis is the *probability of a positive rating* for each θ , $\beta(\theta) \triangleq \Pr(\text{positive rating}|\theta)$. Let $y_\ell(\theta) \sim \text{Bernoulli}(\beta(\theta))$ be the rating an item of quality θ receives at the ℓ th time it matches.

Aggregating ratings and ranking sellers. These ratings are aggregated into a *reputation score*, $x_k(\theta)$, at each time k . The score is the fraction of positive ratings received up to time k : $x_k(\theta) \triangleq \frac{1}{n_k(\theta)} \sum_{\ell=0}^{n_k(\theta)} y_\ell(\theta)$ with $x_0(\theta) \triangleq 0$ for all θ . Thus, $x_k(\theta) \sim \frac{1}{n_k(\theta)} \text{Binomial}(\beta(\theta), n_k(\theta))$.

System state. The state of the system is given by a joint distribution $\mu_k(\Theta, X)$, which gives the *mass* of items of quality $\theta \in \Theta \subset [0, 1]$ with aggregate score $x_k(\theta) \in X \subset [0, 1]$ at time k . Because our model is a continuum, the evolution of the system state μ_k follows a deterministic dynamical system.

We have described these dynamics at the level of individual items; however, such statements should be interpreted as describing the evolution of the joint distribution μ_k . The state update for μ_k is determined by the *mass* of items that match and the *distributions* of their ratings. A formal description of the state evolution is in Appendix Section C.2.1.

Platform objective. The platform wishes to rank the items accurately. Given β and $\theta_1 > \theta_2$, define:

$$P_k(\theta_1, \theta_2|\beta) = \mu_k(x_k(\theta_1) > x_k(\theta_2)|\theta_1, \theta_2) - \mu_k(x_k(\theta_1) < x_k(\theta_2)|\theta_1, \theta_2) \quad (4.1)$$

This expression captures observed score ranking's accuracy. When $\theta_1 > \theta_2$ but $x_k(\theta_1) < x_k(\theta_2)$, the ranking mistakenly orders θ_1 below θ_2 . A good system has large $P_k(\theta_1, \theta_2|\beta)$. Integrating across items creates the following objective for each time k :

$$W_k = \int_{\theta_1 > \theta_2} w(\theta_1, \theta_2) P_k(\theta_1, \theta_2|\beta) d\theta_1 d\theta_2 \quad (4.2)$$

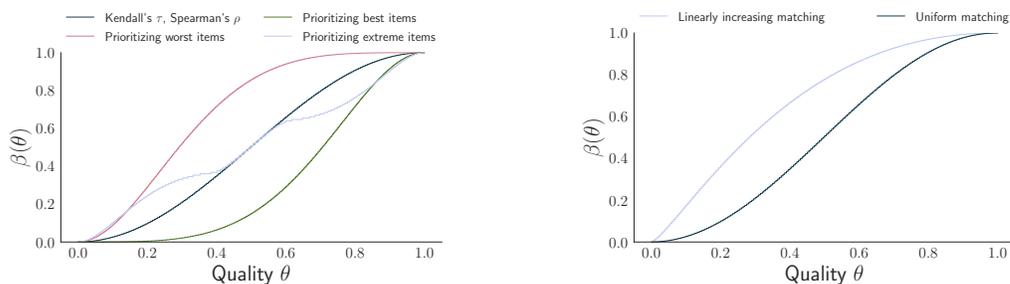
Weight function $w(\theta_1, \theta_2) > 0$ indicates how much the platform cares about not mistaking a quality θ_1 item with a quality θ_2 item. We consider scaled w such that $\int_{\theta_1 > \theta_2} w(\theta_1, \theta_2) d\theta_1 d\theta_2 = 1$.

Our first question then becomes: *What β yields the highest value of W_k ?* As discussed above, the platform influences β through the design of its rating system. The optimal choice of β sets the benchmark.

Discussion. Objective function. The specification (4.2) of the objective is quite rich. It contains scaled versions of Kendall's τ (with $w(\theta_1, \theta_2) = 1$ for all θ_1, θ_2) and Spearman's ρ (with $w(\theta_1, \theta_2) = \theta_1 - \theta_2$) rank correlations. w allows the platform to encode, for example, that it cares more about correctly ranking just the very best, very worst, or items at both extremes.³ Tarsitano (2009) and da Costa and Roque (2006) discuss other well-studied examples.

Relationship between model components. Qualitatively, β affects W_k as follows, as previewed in the introduction: when $\beta(\theta_1) \approx \beta(\theta_2)$, then $x_k(\theta_1) \approx x_k(\theta_2)$, and so $P_k(\theta_1, \theta_2|\beta)$ is small (errors are

³We use $\theta_1\theta_2(\theta_1 - \theta_2)$, $(1 - \theta_1)(1 - \theta_2)(\theta_1 - \theta_2)$, and $(\frac{1}{2} - \theta_1)^2(\frac{1}{2} - \theta_2)^2(\theta_1 - \theta_2)$ as examples.



(a) Fix $g = 1$, with various objective function weights (b) Fix $w = 1$. Vary matching, $g = 1$, and $g = \frac{1+10\theta}{11}$

Figure 4.1: Optimal β (with $M = 200$) with various objective weight functions w and matching rates g .

common). A good design thus would have large $\beta(\theta_1) - \beta(\theta_2)$ for $\theta_1 > \theta_2$ where $w(\theta_1, \theta_2)$ is large. Matching function g also affects P_k and thus W_k : when $g(\theta)$ is large, more ratings are sampled from item of quality θ , i.e. $n_k(\theta)$ is higher, and so $x_k(\theta)$ is more closely concentrated around its mean $\beta(\theta)$. Thus, $P_k(\theta, \theta' | \beta)$ increases (for all θ') with $g(\theta)$. A good design of β thus considers both w and g .

Matching. As noted above, we assume items receive a non-decreasing number of ratings based on their true quality, through matching function $g(\theta)$. This is a reasonable approximation for our analysis, where we focus on the asymptotic rate of convergence of the ranking based on to the true ranking, as the number of ratings increases. In practice, items will be more likely to match when they have a higher *observed* aggregate score. Similarly, our model makes the stylized choice that all items have the same age. In reality, items have different ages in platforms.

Non-response. In practice, many buyers choose not to rate items, which our model does not capture. One possible approach is to treat non-response as a bad experience, which yields more information in the work of Nosko and Tadelis (2015). Solutions to non-response is an important area of work.

4.3.2 Large deviations & discretization

Recall the question: *What β yields the highest value of W_k ?* We now refine objective W_k and constrain β to form a non-degenerate, feasible optimization task.

Large deviation rate function. W_k is not one objective: it has a different value per time k , and no single β simultaneously optimizes W_k for all k .⁴ Considering *asymptotic* performance is also

⁴For example, consider β such that the worst half of items never receive a positive rating and the rest always do. It would perform comparatively well for a small number of ratings k , as it quickly distinguishes the best from the worst items. However it would never distinguish items within the same half. Some β' may make more mistakes initially but perform better at larger k .

insufficient: when β is strictly increasing in θ , $\lim_{k \rightarrow \infty} P_k(\theta_1, \theta_2 | \beta) = 1 \quad \forall \theta_1, \theta_2$ by the law of large numbers. Thus, $W \triangleq \lim_{k \rightarrow \infty} W_k = 1$, and *any* such β is asymptotically optimal.

For this reason, we consider maximization of the *rate* at which W_k converges, i.e., how *fast* the estimated item ranking converges to the true item ranking. We use a large deviations approach (Dembo and Zeitouni, 2010) to quantify this convergence rate. Formally, given sequence $Y_k \leq \lim_{k \rightarrow \infty} Y_k = Y$, the *large deviations rate of convergence* is $-\lim_{k \rightarrow \infty} \frac{1}{k} \log(Y - Y_k) = c$. If c exists then Y_k approaches Y exponentially fast: $Y - Y_k = e^{-kc+o(k)}$.

Then, we wish to *choose* β to maximize W_k 's large deviations rate, $r = -\lim_{k \rightarrow \infty} \frac{1}{k} \log(W - W_k)$.

Discretizing β . Unfortunately, even this problem is degenerate if we consider continuous β : for any β that is not piecewise constant, the large deviations rate of convergence is zero, i.e., convergence of W_k to its limit is only polynomially fast, and characterizing the dependence of this convergence rate on β is intractable. Thus, the rate of convergence for W_k is not a satisfactory objective with continuous β .

We make progress by *discretizing* β ; in particular, we restrict attention to optimization over stepwise increasing β functions.⁵ Among stepwise increasing β , the large deviations rate of W_k to its limiting value W can be shown to be nondegenerate, i.e. $\exists c > 0$ s.t. $W - W_k = e^{-kc+o(k)}$. (See Lemma C.3.4 in the Appendix for further discussion.)

Notationally, we will calculate an optimal stepwise increasing β with M levels, i.e. there are M intervals $S_i \subset [0, 1]$ and levels t_i such that when $\theta_1, \theta_2 \in S_i$, then $\beta(\theta_1) = \beta(\theta_2) \triangleq t_i$. The challenge is calculating an optimal $\mathbf{S}^* = \{S_i\}$ and $\mathbf{t}^* = \{t_i\}$.

The physical interpretation is that we group the items into M subsets (types) $S_i \subset [0, 1]$. When items θ_1, θ_2 are in the same subset, then their asymptotic reputation scores are the same, $\lim_k x_k(\theta_1) = \lim_k x_k(\theta_2) \triangleq t_i$. These items cannot be distinguished from one another even asymptotically.

Though discretization allows us to define a large deviations rate for W_k , it comes at a cost: W , the limiting value of W_k , is no longer one. Different discretization choices \mathbf{S} result in different W .

Our optimization problem: *Within the class of stepwise increasing functions with M levels, find the β that is optimal, i.e. is*

- (1) *Asymptotically optimal.* It yields the highest limiting value of W_k . AND
- (2) *Rate optimal.* It yields the fastest large deviations rate r among *asymptotically optimal* β .

A remarkable result of our paper is a $\mathcal{O}(M \log^2 \frac{M}{\epsilon})$ procedure to find an optimal β with M levels.

4.3.3 Solving the optimization problem

The theorem below shows that the problem decomposes into two stages: first, find optimal discretization intervals \mathbf{S}^* ; then, find optimal \mathbf{t}^* given \mathbf{S}^* .

⁵Note that, even for purely computational reasons, calculating β requires discretization.

Theorem 4.3.1. *The β defined by the following choices of \mathbf{S}^* and \mathbf{t}^* is optimal:*

$$\begin{aligned} \mathbf{S}^* &= \arg \max \sum_{0 \leq i < j < M} \int_{\theta_2 \in S_i, \theta_1 \in S_j} w(\theta_1, \theta_2) d(\theta_1, \theta_2) \\ \mathbf{t}^* &= \arg \max r(\mathbf{t}), \text{ where }^6 g_i \triangleq \inf_{\theta \in S_i} g(\theta) \text{ and} \\ r(\mathbf{t}) &\triangleq - \lim_{k \rightarrow \infty} \frac{1}{k} \log(W - W_k) \\ &= \min_{0 \leq i \leq M-2} \inf_{a \in \mathbb{R}} \{g_{i+1} KL(a || t_{i+1}) + g_i KL(a || t_i)\} \end{aligned} \quad (4.3)$$

The proof is in the Appendix. The main hurdle is showing that the continuum of rates for $P_k(\theta_1, \theta_2)$ for each pair θ_1, θ_2 translates into a rate for W_k . This decomposition separates our two questions: \mathbf{S}^* maximizes the limiting value of W_k *given any* \mathbf{t} , and depends only on w ; Then, \mathbf{t}^* maximizes the rate at which the limiting value is reached, given g_i .

For Kendall's τ and Spearman's ρ , the optimal intervals are simply equispaced in $[0, 1]$, i.e. $S_i^* = [\frac{i}{M}, \frac{i+1}{M})$, because the entire item quality distribution is equally important. For other objective weight functions w , the difficulty of finding the optimal subsets \mathbf{S}^* depends on the properties of w . Since \mathbf{S}^* is trivial for Kendall's τ and Spearman's ρ – and w is just an analytic tool that formalizes a platform's goals – we focus on finding the optimal levels \mathbf{t}^* .

Discussion. One may naturally wonder why we introduced a continuum of quality $[0, 1]$ and then discretized into M subsets, instead of starting with M types. As established in Theorem 4.3.1, *how* we discretize (i.e. solving for \mathbf{S}^*) allows for optimization of different objective weight functions w ; it determines *which* items are most valuable to distinguish.

Suppose we started with a set of M items. Then the only remaining challenge is to equalize the rates at which each item is separated from others: the large deviations rate is unaffected by the weight function w (it does not appear in the simplification of $r(\mathbf{t})$). In other words, *given* a discrete set of M items (equiv, given \mathbf{S}^*), calculating the optimal \mathbf{t} is equivalent to solving a *maximin* problem for the rates at which each type is distinguished from each of the others. Thus, the algorithm below also solves the *inverse* bandits problem in which we wish to rank the M arms, and we can choose the structure of the (binary) observations at each arm.

We further note that the choice of M is not consequential; in the Appendix Section C.2.4 we show that in an appropriate sense, a sequence of optimal β_M for each M converges as M gets large.

Algorithm to find the optimal levels We now describe how to find \mathbf{t}^* , the maximizer of $r(\mathbf{t})$.

The following lemma describes a system of equations to find the \mathbf{t}^* that maximizes r . It states that \mathbf{t}^* equalizes the rates at which each interval i is separated from its neighbors. The proof involves manipulation of r and convexity, and is in the appendix.

⁶ $KL(a||b) = a \log \frac{b}{a} + (1-a) \log \frac{1-b}{1-a}$ is the Kullback-Leibler (KL) divergence between Bernoulli random variables with success probabilities a and b respectively.

Lemma 4.3.1. *The unique solution \mathbf{t}^* to the following system of equations maximizes $r(\mathbf{t})$:*

$$\begin{aligned} r(t_0, t_1) &= r(t_1, t_2) = \dots = r(t_{M-2}, t_{M-1}) \\ t_0 &= 0, t_{M-1} = 1 \\ r(t_{i-1}, t_i) &\triangleq -\log \left[\left((1 - t_{i-1})^{\frac{g_{i-1}}{g_{i-1}+g_i}} (1 - t_i)^{\frac{g_i}{g_{i-1}+g_i}} + t_{i-1}^{\frac{g_{i-1}}{g_{i-1}+g_i}} t_i^{\frac{g_i}{g_{i-1}+g_i}} \right)^{g_{i-1}+g_i} \right] \end{aligned} \quad (4.4)$$

We do not know of any algorithm that efficiently and provably solves such convex equality systems in general. However, we leverage some structure in our setting to develop an algorithm, *NestedBisection*, with run-time and optimality guarantees. The efficiency of our algorithm results from the property that, given a rate, t_i is uniquely determined by the value of either of the adjacent levels t_{i-1}, t_{i+1} , reducing an exponentially large search space to an almost linear one. Physically, i.e., we only need to separate each type of item from its neighboring types.

Below we include pseudo-code. Akin to branch and bound, the algorithm proceeds via bisection on the optimal value of t_{M-2} . For each candidate value of t_{M-2} , the other values can be found using a sequence of bisection subroutines. These values approximately obey all the equalities in the system (4.4) except the first. The direction of the first equality's violation reveals how to change the interval for the next outer bisection iteration.

Theorem 4.3.2. *NestedBisection finds an ϵ -optimal \mathbf{t} in $\mathcal{O}(M \log^2 \frac{M}{\epsilon})$ operations, where ϵ -optimal means that $r(\mathbf{t})$ is within additive constant ϵ of optimal.*

The proof is in the appendix. The main difficulty is finding a Lipschitz constant $\epsilon(\delta)$ for how much the rate changes with a shift δ in a level t_i . This requires lower bounding t_1 as a function of M . In practice, the algorithm runs instantaneously on a modern machine (e.g. for $M = 200$).

ALGORITHM 1: Nested Bisection

Data: Number of intervals M , match function g

Result: β levels, i.e. $\{t_0 \dots t_{M-1}\}$

```

1 Function main ( $M, \delta, g$ )
2   while Uncertainty region for  $t_{M-2}$  is bigger than error tolerance do
3     Calculate  $r(t_{M-2}, 1)$ , the rate between current guess for  $t_{M-2}$ , and  $t_{M-1} = 1$ .
4     Fixing  $t_{M-2}$ , find  $t_1 \dots t_{M-3}$  such that  $r(t_1, t_2) = r(t_2, t_3) = \dots = r(t_{M-3}, t_{M-2}) = r(t_{M-2}, 1)$ ,
       which can be done through a sequence of bisection routines.
5     Calculate  $r(0, t_1)$ , the rate between current guess for  $t_1$ , and  $t_0 = 0$ .
6     Compare  $r(t_{M-2}, 1)$  and  $r(0, t_1)$ , adjust uncertainty region for  $t_{M-2}$  accordingly.
7   return  $\{t_i\}$ 

```

4.3.4 Visualization and discussion

Figure 4.1 shows how the optimal β varies with weights w and matching rates g . Higher relative weights in a region lead to a larger range of $\beta(\theta)$ there to make it easier to distinguish those items

(e.g., prioritizing the best items induces a β shifted right). Higher relative matching rates $g(\theta)$ have the opposite effect, as frequent sampling naturally increases accuracy for the best items. We formalize this shifting in Appendix Section C.2.3.

It is interesting that even the basic case, with $w = 1$ and $g = 1$, has a non-trivial β . One would expect, with weight and matching functions that treat all items the same, that β would be linear, i.e. $\beta(\theta) = \theta$. Instead, a third factor non-trivially impacts optimal design: binomial variance is highest near $\beta(\theta) = \frac{1}{2}$. Items that receive positive ratings at such frequency have high-variance scores, and thus the optimal β has a smaller mass of items with such scores.

4.4 Designing approximately optimal, implementable rating systems

We now turn to our second question: *How does a platform build and implement a real rating system such that buyers behave near-optimally, i.e. according to a calculated β ?* In this section, we give an example design procedure for how a platform would do so, and in the next section we validate our procedure through an experiment on Mechanical Turk.

Recall that $\beta(\theta)$ gives the probability at which an item of quality θ should receive a positive rating. However, the platform cannot force buyers to rate according to this function. Rather, it must ask *questions* of buyers that will induce a proportion $\beta(\theta)$ of them to give positive ratings for an item θ .

Throughout the section, we assume that an optimal $\beta(\theta)$ has been calculated (for some M , g , and w).

Resources available to platform. We suppose the platform has a set \mathcal{Y} of possible binary questions that it could ask a buyer, e.g., “Are you satisfied with your experience” or “Is this experience your best on our platform?”. Informally, at each rating opportunity (i.e., match made), the rater can be shown a single question $y \in \mathcal{Y}$. Let $\psi(\theta, y)$ be the probability an item quality θ would receive a positive rating when the rater is asked question $y \in \mathcal{Y}$.

We further suppose the platform has a set Θ of representative items for which it has access to item quality. Θ , then, is the granularity at which the platform can collect data about historical performance, or otherwise get expert labels. (We assume $M \gg |\Theta|$).

Using known set Θ , the platform can run an experiment to create an estimate $\hat{\psi}(\theta, y), \forall y \in \mathcal{Y}, \theta \in \Theta$.

Design heuristic. How can the platform build an effective rating system using these primitives and β ? We consider the following heuristic design: the platform randomly shows a question $y \in \mathcal{Y}$ to each buyer. The choice of the platform is a *distribution* $H(y)$ over $y \in \mathcal{Y}$; in other words, for the platform the design of the system amounts to choosing the frequency with which each question is shown. At each rating opportunity, y is chosen from \mathcal{Y} according to H , independently across

opportunities.

Clearly, the probability that an item θ receives a positive rating is: $\tilde{\beta}(\theta) \triangleq \sum_{y \in \mathcal{Y}} \psi(\theta, y)H(y)$.

We want a distribution H such that $\tilde{\beta}(\theta) = \beta(\theta)$ for all $\theta \in [0, 1]$, i.e., that the positive rating probability for each item is exactly the optimal value. However, there may not exist *any* set of questions \mathcal{Y} with associated ψ and choice of H such that $\tilde{\beta} = \beta$.⁷

We propose the following heuristic to address this difficulty. Choose a probability distribution H to minimize the following L_1 distance:

$$\min_{H: \|H\|_1=1} \sum_{\theta \in \Theta} |\beta(\theta) - \sum_{y \in \mathcal{Y}} \hat{\psi}(\theta, y)H(y)| \quad (4.5)$$

This heuristic uses the data available to the platform, $\hat{\psi}(\theta, y)$ for a set of items $\theta \in \Theta$, and designs H so at least these items receive ratings close to their optimal ratings $\beta(\theta_i)$. Then, as long as ψ is well-behaved, and Θ is representative of the full set, one can hope that $\tilde{\beta}(\theta) \approx \beta(\theta)$, for all $\theta \in [0, 1]$.

Discussion. *Real-world analogue \mathcal{E} constraints.* A special case of this system is already in place on many platforms, where the same question is always shown. Static systems can be designed by restricting H to only have mass at one question y . More generally, constraints can be used in optimization (4.5).

Limitations. Our model does not allow for y to be chosen adaptively based on the platform’s current knowledge of the item. In practice, this may be a reasonable restriction for implementation purposes. Our model also restricts aggregation to be binary; the platform in our model does not use information on how “hard” a question y is.

4.5 Mechanical Turk experiment

In the previous sections, we showed how to find an optimal rating function β and we how to apply such a β to design a binary rating system using empirical data. In this section, we deploy an experiment on Amazon Mechanical Turk to apply these insights in practice. First, we collect data that can be used to create a reasonable real-world example of $\psi(\theta, y)$, as a proof of concept with which we can apply our optimization approach. Then, we use this model to demonstrate some key features of optimal and heuristic designs as computed via our methodology, and show that they perform well relative to natural benchmarks. Details of experimental design, simulation methodology, and results are in Appendix C.1.

Experiment description. We have a set of 10 English paragraphs of various writing quality, with *expert scores* θ , from a TOEFL book (Educational Testing Service, 2005); there were 5 unique possible experts ratings, i.e. $\theta \in \{.1, .3, .5, .7, .9\}$. For each possible rating, we have two paragraphs who received that score from experts.

⁷There are special cases where an exact solution exists. For example, let $\mathcal{Y} = [0, 1]$, and $\psi(\theta, y) = \mathbb{I}[\theta \geq y]$.

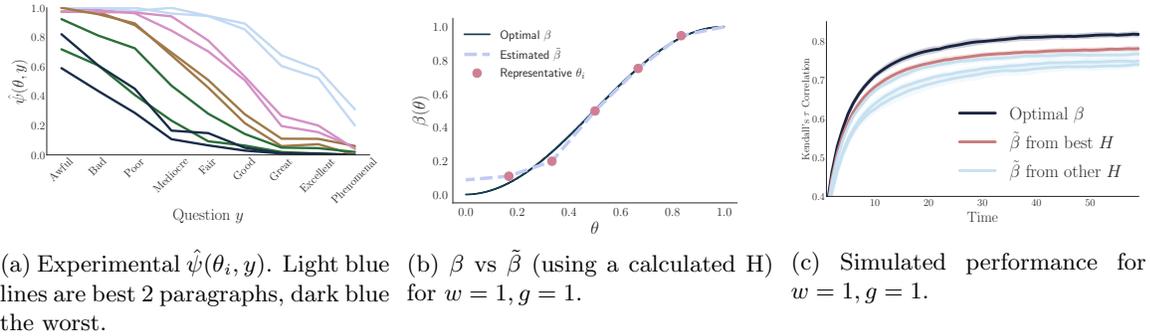


Figure 4.2: Experiment and simulation results

We asked workers on Mechanical Turk to rate the writing quality of the paragraphs from a set of adjectives, \mathcal{Y} . Using this data, we estimate $\psi(\theta, y)$, i.e., the probability of positive rating when a question based on adjective $y \in \mathcal{Y}$ is shown for paragraph with quality $\theta \in \Theta$. (e.g., “Would you consider this paragraph of quality [adjective] or better?”) Figure 4.2a shows our estimated $\hat{\psi}$ for our 10 paragraphs.

Optimization. Next, we find the optimal β for various matching and weight functions using the methods from Section 4.3. In particular, we have β for all permutations of the cases $g = 1, g = \frac{1+10\theta}{11}$, and $w = 1, w = \theta_1\theta_2(\theta_1 - \theta_2)$, and $w = (1 - \theta_1)(1 - \theta_2)(\theta_1 - \theta_2)$. Recall that this step does not use experimental data.

Then, using $\hat{\psi}$ and calculated β s, we apply the heuristic from Section 4.4 to find the distribution H with which to sample the questions (adjectives) in \mathcal{Y} . Figure 4.2b shows the optimal β (with $g = 1, w = 1$), and estimated $\tilde{\beta}$ from the procedure.

Simulation. Finally, we study the performance of these designs via simulation in various settings. We simulate a system with 500 items and 100 buyers according to the model in Section 4.3.1, except that matching is *stochastic*: at each time, a random 100 items receive ratings, based on *observed* scores $x_k(\theta)$ rather than true quality θ . Furthermore, in some simulations, we have sellers *entering and exiting* the market with some probability at each time step. We measure the performance of all the designs. For comparison, we also simulate a *naive* $H = \frac{1}{|\mathcal{Y}|}$.

Note that our experiment only provides $\hat{\psi}$ associated with qualities $\theta \in \Theta$, and for simulations we construct a full $\psi(\theta, y), \forall \theta \in [0, 1]$ from these points by averaging and interpolating (in order to model human behavior for the full system). Further, our *calibrated* simulations only provide rough evidence for the approach: although we use real-world data, the simulations assume that our model reflects reality, except for where we deviate as described above.

Results and discussion. Figure 4.2c shows the simulated performance (as measured by Kendall’s τ correlation) of the various designs over time, when $g = 1$. Further plots are in the

Appendix Figure C.4, showing performance under various weight functions w and matching functions g , and with items entering and exiting the market. We find that:

First, the optimal β for each setting outperforms other possible functions, as expected. The designs are robust to (some) assumptions in the model being broken, especially regarding matching.

Second, the H from our procedure outperforms other designs, but is worse than the optimal system β . In general, the simulated gap between an implemented system and optimal design β provides the platform quantitative insight on the system’s sub-optimality.

Third, comparing $\tilde{\beta}$ to β gives qualitative insight on *how* to improve the system. For example, in Figure 4.2b, $\tilde{\beta}$ is especially inaccurate for $\theta \in [0, .4]$. The platform must thus find better questions for items of such quality. Figure 4.2a corroborates: our questions cannot separate two low quality paragraphs rated differently by experts (in dark blue and green).

A wide range of recent empirical work has documented that real-world rating systems experience substantial inflation; almost all items receive positive ratings almost every match (Filippas et al., 2019; Fradkin et al., 2018; Tadelis, 2016). Our formulation helps understand how – and whether – such inflation is suboptimal, and provides guidance to platform designers. In particular, rating inflation can be interpreted as a current $\beta(\theta)$ that is very high for almost all item qualities θ . This system is well-performing if the platform objective is to separate the worst items from the rest, or if high quality items receive many more ratings than low quality ones; it is clearly sub-optimal in other cases. Our paper provides a template for how a platform might address such a situation.

Part III

Designing Voting Mechanisms on Civic Engagement Platforms

Chapter 5

Iterative Local Voting

5.1 Introduction

Methods and experiments to increase large-scale, direct citizen participation in policy-making have recently become commonplace as an attempt to revitalize democracy. Computational and crowdsourcing techniques involving human-algorithm interaction have been a key driver of this trend (Cabannes, 2004; Lee et al., 2014; McDermott, 2010; Quarfoot et al., 2017). Some of the most important collective decisions, whether in government or in business, lie in high-dimensional, continuous spaces – e.g. budgeting, taxation brackets and rates, collectively bargained wages and benefits, urban planning etc. Direct voting methods originally designed for categorical decisions are typically infeasible for collective decision-making in such spaces. Although there has been some theoretical progress on designing mechanisms for continuous decision-making (Cheng and Zhou, 2015; Moulin, 1980; Procaccia and Tennenholtz, 2009), in practice these problems are usually resolved using traditional approaches – they are either discretized before running an election, or are decided upon through negotiation by committee, such as in a standard representative democracy (Cabannes, 2004; Gilman, 2012; Goel et al., 2016; Shah, 2007; Sintomer et al., 2008).

One of the main reasons for the current gap between theory and practice in this domain is the challenge of designing practically implementable mechanisms. We desire procedures that are simple enough to explain and use in practice, and that result in justifiable solutions while being robust to the inevitable deviations from ideal models of user behavior and preferences. To address this

This chapter is joint with Vijay Kamble, Ashish Goel, David Marn, and Kamesh Munagala. Published at the International Conference on World Wide Web (WWW), in 2017 (Garg et al., 2017). Journal version published at the Journal of Artificial Intelligence Research (JAIR), in 2019 (Garg et al., 2019b). Supported by NSF grant nos. CCF-1408784, CCF-1637397, CCF-1637418, and IIS-1447554, ONR grant no. N00014-15-1-2786, ARO grant no. W911NF-14-1-0526, and the NSF Graduate Research Fellowship under grant no. DGE-114747. This work benefited from many helpful discussions with Oliver Hinder.

challenge, a social planner must first make practically reasonable assumptions on the nature and complexity of feedback that can be elicited from people and then design simple algorithms that operate effectively under these conditions. Further, while robustness to real-world model deviations may be difficult to prove in theory, it can be checked in practice through experiments.

We first tackle the question of what type of feedback voters can give. In general, for the types of problems we wish to solve, a voter cannot fully articulate her utility function. Even if voters in a voting booth had the patience to state their exact utility for a reasonably large number of points (e.g. how much they liked each candidate solution on a scale from one to five), there is no reason to believe that they could do so in any consistent manner. On the other hand, we posit that it is relatively easy for people to choose their favorite amongst a reasonably small set of options, or articulate how they would like to locally modify a candidate solution to better match their preferences. Such an assumption is common and is a central motivation in social choice, especially *implicit utilitarian voting* (Procaccia and Rosenschein, 2006).

In this paper, we study and experimentally test a type of algorithm for large-scale preference aggregation that effectively leverages the possibility of asking voters such easy questions. In this algorithm that we call ITERATIVE LOCAL VOTING (ILV), voters are sequentially sampled and are asked to modify a candidate solution to their favorite point within some local neighborhood, until a stable solution is obtained (if at all). With a continuum of voters, no one votes more than once. The algorithm designer has flexibility in deciding how these local neighborhoods are defined – in this paper we focus on neighborhoods that are balls in the \mathcal{L}^q norm, and in particular on the cases where $q = 1, 2$ or ∞ . (For $M < \infty$ dimensional vectors, the \mathcal{L}^q norm $\|x\|_q \triangleq \sqrt[q]{\sum_m |x_m|^q}$. $q = 1, 2$ and ∞ neighborhoods correspond to bounds on the sum of absolute values of the changes, the sum of the square of the changes, and the maximum change, respectively.)

More formally, consider a M -dimensional societal decision problem in $\mathcal{X} \subset \mathbb{R}^M$ and a population of voters \mathcal{V} , where each voter $v \in \mathcal{V}$ has bounded utility $f_v(x) \in \mathbb{R}, \forall x \in \mathcal{X}$. Then we consider the class of algorithms described in Algorithm 2. We study the algorithm class under two plausible models of how voters respond to query (1), which asks for the voter’s favorite point in a local region.

- **Model A:** One possibility is that voters exactly perform the maximization asked of them, responding with their favorite point in the given \mathcal{L}^q norm constraint set. In other words, they return a point $\arg \max_{x \in \{s: \|s - x_{t-1}\|_q \leq r_t\}} f_{v_t}(x)$. Note that by definition of this movement, the algorithm is *myopically incentive compatible*: if a voter is the last voter and no projections are used, then truthfully performing this movement is the dominant strategy. In general, the mechanism is not globally incentive compatible, nor incentive compatible with projections onto the feasible region. Simple examples of manipulations in both instances exist.
- **Model B:** On the other hand, voters may not actually search within the constraint set to find their favorite point inside of it. Rather, a voter v may have an idea about how to best improve the current point and then move in that direction to the boundary of the given constraint set. This

Algorithm 2: ITERATIVE LOCAL VOTING (ILV)

- 1 *Inputs:* Initial solution $x_0 \in \mathcal{X}$, tolerance $\epsilon > 0$, an integer N , initial radius $r_0 > 0$, termination time T , norm q for local neighborhood.
- 2 *Output:* Solution x .
- For $t \geq 1$, sample a voter $v_t \in \mathcal{V}$ at random from the population; set $r_t = r_0/t$ and elicit

$$x'_t = \arg \max_{x \in \{s: \|s - x_{t-1}\|_q \leq r_t\}} f_{v_t}(x), \quad (1)$$

and then compute $x_t = [x'_t]_{\mathcal{X}}$, where $[\cdot]_{\mathcal{X}}$ is a projection onto space \mathcal{X} ; i.e. ask the voter to move to her favorite point within constraints, and then project the reported point onto \mathcal{X} .

- Stop when either $t = T$, in which case return x_T , or when $\max_{l, m \in \{t-N, \dots, t\}} |x_l - x_m| \leq \epsilon$, in which case return $x = x_t$.
-

model leads to a voter moving the current solution in the direction of the gradient of her utility function, returning a point $x_{t-1} + r_t \frac{g_t}{\|g_t\|_q}$, for some $g_t \in \partial f_{v_t}(x_{t-1})$. Note that $\partial f(x)$ denotes the set of subgradients of a function f at x , i.e. $g \in \partial f(x)$ if $\forall y, f(y) - f(x) \geq g^T(y - x)$.

ILV is directly inspired by the stochastic approximation approach to solve optimization problems (Robbins and Monro, 1951), especially stochastic gradient descent (SGD) and the stochastic subgradient method (SSGM). The idea is that if (a) voter preferences are drawn from some probability distribution and (b) the response of a voter to the query (1) moves the solution approximately in the direction of her utility gradient, then this procedure *almost* implements stochastic gradient descent for minimizing negative expected utility.

The caveat is that although the procedure can potentially obtain the direction of the gradient of the voter utilities, it cannot in general obtain any information about its magnitude since the movement norm is chosen by the procedure itself. However, we show that for certain plausible utility and voter response models, the algorithm does indeed converge to a unique point with desirable properties, including cases in which it converges to the societal optimum.

Note that with such feedback and without any additional assumptions on voter preferences (e.g. that voter utilities are normalized to the same scale), no algorithm has any hope of finding a desirable solution that depends on the cardinal values of voters' utilities, e.g., the social welfare maximizing solution (the solution that maximizes the sum of agent utilities). This is because an algorithm that uses only ordinal information about voter preferences is insensitive to any scaling or even monotonic transformations of those preferences.

5.1.1 Contributions

This work is a step in extending the vast literature in social choice to continuous spaces, taking into account the feedback that voters can *actually give*. Our main theoretical contributions are as follows:

- **Convergence for \mathcal{L}^p normed utilities:** We show that if the agents cost functions can be expressed as the \mathcal{L}^p distance from their ideal solution, and if agents correctly respond to query (1), then an interesting duality emerges: for $p = 1, 2$ or ∞ , using \mathcal{L}^q neighborhoods, where $q = \infty, 2$ and 1 respectively, results in the algorithm converging to the unique social welfare optimizing solution. Whether such a result holds for general (p, q) , where q is the dual norm to p (i.e. $1/p + 1/q = 1$), is an open question. However, we show that such a general result holds if, in response to query (1), the voter instead moves the current solution in the direction of the gradient of her utility function to the neighborhood boundary.
- **Convergence for other utilities:** Next, we show convergence to a unique solution in two cases: (a) when the voter cost can be expressed as a weighted sum of \mathcal{L}^2 distances over sub-spaces of the solution space, under \mathcal{L}^2 neighborhoods – in which case the solution is also Pareto efficient, and (b) when the voter utility can be additively decomposed across dimensions, under \mathcal{L}^∞ neighborhoods – in which case the algorithm converges to the median of the ideal solutions of the voters on each dimension.

We then build a platform and run the first large-scale experiment in voting in multi-dimensional continuous spaces, in a budget allocation setting. We test three variants of ILV: with \mathcal{L}^1 , \mathcal{L}^2 and \mathcal{L}^∞ neighborhoods. Our main findings are as follows:

- We observe that the algorithm with \mathcal{L}^∞ neighborhoods is the only alternative that satisfies the first-order concern for real-world deployability: consistent convergence to a unique stable solution. Both \mathcal{L}^1 and \mathcal{L}^2 neighborhoods result in convergence to multiple solutions.
- The consistent convergence under \mathcal{L}^∞ neighborhoods in experiments strongly suggests the decomposability of voter utilities for the budgeting problem. Motivated by this observation, we propose a general class of decomposable utility functions to model user behavior for the budget allocation setting.
- We make several qualitative observations about user behavior and preferences. For instance, voters have large indifference regions in their utilities, with potentially larger regions in dimensions about which they care about less. Further, we show that asking voters for their ideal budget allocations and how much they care about a given item is fraught with UI biases and should be carefully designed.

We remark that an additional attractive feature of such a constrained local update algorithm in a large population setting is that strategic behavior from the voters is less of a concern: even if a

single voter is strategic, her effect on the outcome is negligible. Further, it may be difficult for a voter, or even a coalition of voters, to strategically vote; one must reason over the possible future trajectories of the algorithm over the randomness of future voters. One coalition strategy for \mathcal{L}^2 and \mathcal{L}^∞ neighborhoods, voters trade votes on different dimensions with one another; we leave robustness to such strategies to future work.

The structure of the paper is as follows. After discussing related work in Section 5.2, we present convergence results for our algorithm under different settings in Section 5.3. In Section 5.4, we introduce the budget allocation problem and describe our experimental platform. In Section 5.5, we analyze the experiment results, and then we conclude the paper in Section 5.6. The proofs of our results are in the appendix.

5.2 Related Work

Our work relates to various strands of literature. Furthermore, the term “iterative voting” is also used in other works to denote unrelated methods (Airiau and Endriss, 2009; Meir et al., 2010).

5.2.1 Stochastic Subgradient Method

As discussed in the introduction, we draw motivation from the stochastic subgradient method (SSGM), and our main proof technique is mapping our algorithm to SSGM. Beginning with the original stochastic approximation algorithm by Robbins and Monro (1951), a rich literature surrounds SSGM, cf. Boyd and Mutapcic; Jiang and Walrand (2010); Nemirovski et al. (2009); Shor (1998).

5.2.2 Iterative Local Voting

A version of our algorithm, with \mathcal{L}^2 norm neighborhoods, has been proposed independently several times (Benjamin et al., 2013; Chung and Duggan, 2018; Hylland and Zeckhauser, 1980) and is referred to as Normalized Gradient Ascent (NGA). Instead of directly asking voters to perform query (1), the movement $\frac{\nabla f_v(x_{t-1})}{\|\nabla f_v(x_{t-1})\|_2}$ would be estimated through population surveys to try to compute the fixed point where $\mathbb{E}_v \left[\frac{\nabla f_v(x)}{\|\nabla f_v(x)\|_2} \right] = 0$. (Note that we work with distributions of voters and for strictly concave utility functions, the movement for each voter is well-defined for all but a measure 0 set. Then, given a bounded density function of voters, the expectation is well-defined).

This fixed point has been called Directional Equilibrium (DE) in the recent literature (Chung and Duggan, 2018). The movement is equivalent to the movement in this work in the case voters respond according to Model B and with \mathcal{L}^2 neighborhoods, and we show in Section 5.3.3 that, in such cases, the algorithm converges to a Directional Equilibrium when it converges. We further

conjecture that even under voter Model A, if Algorithm 4 converges, the fixed point is a Directional Equilibrium.

Several properties of the fixed point have been studied, starting from the work of Hylland and Zeckhauser (1980) to more recently, the work of Chung and Duggan (2018) and Benjamin et al. (2013): it exists under light assumptions, is Pareto efficient, and has important connections to the Majority Core literature in economics. Showing that an iterative algorithm akin to ours converges to such a point has been challenging; indeed, except for special cases such as quadratic utilities $f_v(x) = -(x - x^v)^T \Omega(x - x^v)$, with society-wide Ω that encodes the relative importance and relationships between issues (Benjamin et al., 2013), convergence is an open question.

Our algorithm differs from NGA in a few crucial directions, even in the case that the movement is equivalent: by relating our algorithm to SSGM, we are able to characterize the step-size behavior necessary for convergence and show convergence even when each step is made by a single voter, rather than after an estimate of the societal normalized gradient. One can also characterize the convergence rate of the algorithm (Nemirovski et al., 2009). Furthermore, the literature has referred to the \mathcal{L}^2 norm (or “quadratic budget”) constraint as “central to their strategic properties” (Benjamin et al., 2017). In this work, this limitation is relaxed – the same strategic property, myopic incentive compatibility, holds for the other norm constraints for their respective cases.

Finally, because we are primarily interested in designing implementable voting mechanisms, we focus on somewhat different concerns than the directional equilibria literature. However, we believe that the ideas in this work, especially the connections to the optimization literature, may prove useful to work on NGA. To the best of our knowledge, no work studies such an algorithm with other neighborhoods and under ordinal feedback, or implements such an algorithm.

5.2.3 Optimization without Gradients

Because we are concerned with optimization without access to voters’ utility functions or its gradients, this work seems to be in the same vein as recent literature on convex optimization without gradients – such as with comparisons or with pairs of function evaluations (Duchi et al., 2012, 2015; Flaxman et al., 2005; Jamieson et al., 2012). However, in the social choice or human optimization setting, we cannot estimate each voter’s utility functions or gradients exactly rather than up to a scaling term, and yet we would like to find some point with good societal properties. This limitation prevents the use of strategies from such works.

Jamieson et al. (2012), for example, present an optimal coordinate-descent based algorithm to find the optimum of a function for the case in which noisy comparisons are available on that function; in our setting, such an algorithm could be used to find the optimal value for *each voter*, but not the societal optimum because each voter can independently scale her utility function. Duchi et al. (2012) present a distributed optimization algorithm where each node (voter) has access to its own subgradients and a few of its neighbors, but in our case each voter can arbitrarily scale her utility

function and thus her subgradients. Similar problems emerge in applying results from the work of Duchi et al. (2015). In our work, such scaling does not affect the point to which the algorithm converges.

5.2.4 Participatory Budgeting

The experimental setting for this work, and a driving motivation, is participatory budgeting, in which voters are asked to help create a government budget. Participatory budgeting has been among the most successful programs of Crowdsourced Democracy, with deployments throughout the world allocating hundreds of millions of dollars annually, and studies have shown its civic engagement benefits (Cabannes, 2004; Gilman, 2012; Goel et al., 2016; Lee et al., 2014; McDermott, 2010; Shah, 2007; Sintomer et al., 2008).

In a typical election, community members propose projects, which are then refined and voted on by either their representatives or the entire community, through some discrete aggregation scheme. In no such real-world election, to our knowledge, can the amount of money to allocate to a project be determined in a continuous space within the voting process, except through negotiation by representatives.

Goel et al. (2016) propose a “Knapsack Voting” mechanism in which each voter is asked to create a valid budget under the budget constraint; the votes are then aggregated using K-approval aggregation on each dollar in the budget, allowing for fully continuous allocation in the space. This mechanism is strategy-proof under some voter utility models. In comparison, our mechanism works in more general spaces and is potentially easier for voters to do.

We note that our approach may also be adapted for other multi-dimensional voting contexts, such as voting on multiple issues simultaneously, cf. Garg et al. (2018a).

5.2.5 Implicit Utilitarian Voting

With a finite number of candidates, the problem of optimizing some societal utility function (based on the cardinality of voter utilities) given only ordinal feedback is well-studied, with the same motivation as in this work: ordinal feedback such as rankings and subset selections are relatively easy for voters to provide. The focus in such work, referred to as *implicit utilitarian voting*, is to minimize the distortion of the output selected by a given voting rule, over all possible utility functions consistent with the votes, i.e. minimize the worst case error achieved by the algorithm due to an under-determination of utility functions when only using the provided inputs (Caragiannis and Procaccia, 2011; Caragiannis et al., 2017; Goel et al., 2017; Procaccia and Rosenschein, 2006). In this work, we show convergence of our algorithm under certain implicit utility function forms. However, we do not characterize the maximum distortion of the resulting fixed point (or even the convergence to any fixed point) under any utility functions consistent with the given feedback, leaving such analysis for future work.

	Model A	Model B
Spatial , $(p, q) = (2, 2), (1, \infty),$ or $(\infty, 1)$	Social Opt. (Thm 5.3.1)	
Spatial , (p, q) s.t. $1/p + 1/q = 1$?	Social Opt. (Thm 5.3.2)
Weighted Euclidean	Social Opt. (Thm 5.3.1)	
Decomposable	Medians (Thm 5.3.2)	

Table 5.1: Summary of convergence results

5.3 Convergence Analysis

In this section, we discuss the convergence properties of ILV under various utility and behavior models. For the rest of the technical analysis, we make the following assumptions on our model.

- C₁** The solution space $\mathcal{X} \subseteq \mathbb{R}^M$ is non-empty, bounded, closed, and convex.
- C₂** Each voter v has a unique ideal solution $x_v \in \mathcal{X}$.
- C₃** The ideal point x_v of each voter is drawn independently from a probability distribution with a bounded and measurable density function $h_{\mathcal{X}}$.

Under this model, for a solution $x \in \mathcal{X}$, the societal utility is given by $E_v[f_v(x)]$. and the social optimal (SO) solution is any $x^* \in \arg \max_{x \in \mathcal{X}} E_v[f_v(x)]$.

“Convergence” of ILV refers to the convergence of the sequence of random variables $\{x_t\}_{t \geq 1}$ to some $x \in \mathcal{X}$ with probability 1, assuming that the algorithm is allowed to run indefinitely (this notion of convergence also implies the termination of the algorithm with probability 1).

In the following subsections, we present several classes of utility functions for which the algorithm converges, summarized in Table 5.1. We further formalize the relationship to directional equilibria in Section 5.3.3.

5.3.1 Spatial Utilities

Here we consider *spatial* utility functions, where the utilities of each voters can be expressed in the form of some kind of spatial distance from their ideal solutions. First, we consider the following kind of utilities.

Definition 5.3.1. \mathcal{L}^p **normed utilities.** *The voter utility function is \mathcal{L}^p normed if $f_v(x) = -\|x - x_v\|_p, \forall x \in \mathcal{X}$.*

Under such utilities, for $p = 1, 2$ and ∞ , restricting voters to a ball in the dual norm leads to convergence to the societal optimum.

Theorem 5.3.1. *Suppose that conditions **C₁**, **C₂**, and **C₃** are satisfied, the voter utilities are \mathcal{L}^p normed, and voters respond to query (1) according to either **Model A** or **Model B**. Then, ILV*

with \mathcal{L}^q neighborhoods converges to the societal optimal point w.p. 1 when $(p, q) = (2, 2)$, $(1, \infty)$, or $(\infty, 1)$.

The proof is contained in the appendix. A sketch of the proof is as follows. For the given pairs (p, q) , we show that, except in certain ‘bad’ regions, the update rule $x_{t+1} = \arg \min_x [\|x - x_{v_t}\|_p : \|x - x_t\|_q \leq r_t]$ is equivalent to the stochastic subgradient method (SSGM) update rule $x_{t+1} = x_t - r_t g_t$, for some $g_t \in \partial E_v[\|x - x_{v_t}\|_p]$, and that the probability of being in a ‘bad’ region decreases fast enough as a function of r_t . We then leverage a standard SSGM convergence result to finish the proof. One natural question is whether the result extends to general dual norms p, q , where $1/p + 1/q = 1$. Unfortunately, the update rule is not equivalent to SSGM in general, and we leave the convergence to the societal optimum for general (p, q) as an open question.

Further, note that even if each voter could scale their utility function arbitrarily, the algorithm would converge to the same point.

However, the general result does hold for general dual norms (p, q) if one assumes the alternative behavior model.

Theorem 5.3.2. *Suppose that conditions \mathbf{C}_1 , \mathbf{C}_2 , and \mathbf{C}_3 are satisfied, the voter utilities are \mathcal{L}^p normed, and voters respond to query (1) according to **Model B**. Then, ILV with \mathcal{L}^q neighborhoods converges to the societal optimal point w.p. 1 for any $p > 0$ and $q > 0$ such that $1/p + 1/q = 1$.*

The proof is contained in the appendix. It uses the following property of \mathcal{L}^p normed utilities: the \mathcal{L}^q norm of the gradient of these utilities at any point other than the ideal point is constant. This fact, along with the voter behavior model, allows the algorithm to implicitly capture the magnitude of the gradient of the utilities, and thus a direct mapping to SSGM is obtained.

Note that the above result holds even if we assume that a voter moves to her ideal point x_v in case it falls within the neighborhood (since, as explained earlier, the probability of sampling such a voter decreases fast enough).

Next, we introduce another general class of utility functions, which we call *Weighted Euclidean utilities*, for which one can obtain convergence to a unique solution.

Definition 5.3.2. Weighted Euclidean utilities. *Let the solution space \mathcal{X} be decomposable into K different sub-spaces, so that $x = (x^1, \dots, x^K)$ for each $x \in \mathcal{X}$ (where $\sum_{k=1}^K \dim(x^k) = M$). Suppose that the utility function of the voter v is*

$$f_v(x) = - \sum_{k=1}^K \frac{w_v^k}{\|w_v\|_2} \|x^k - x_v^k\|_2.$$

where w_v is a voter-specific weight vector, then the function is a *Weighted Euclidean utility function*. We further assume that $w_v \in \mathcal{W} \subset \mathbb{R}_+^K$ and x_v are independently drawn for each voter v from a joint probability distribution with a bounded and measurable density function, with \mathcal{W} nonempty, bounded, closed, and convex.

This utility function can be interpreted as follows: the decision-making problem is decomposable into K sub-problems, and each voter v has an ideal point x_v^k and a weight w_v^k for each sub-problem k , so that the voter's disutility for a solution is the weighted sum of the Euclidean distances to the ideal points in each sub-problems. Such utility functions may emerge in facility location problems, for example, where voters have preferences on the locations of multiple facilities on a map. This utility form is also the one most closely related to the existing literature on Directional Equilibria and Quadratic Voting, in which preferences are linear. To recover the weighted linear preferences case, set $K = M$, with each sub-space of dimension 1. In this case, the following holds:

Suppose that conditions \mathbf{C}_1 , \mathbf{C}_2 , and \mathbf{C}_3 are satisfied, the voter utilities are Weighted Euclidean, and voters correctly respond to query (1) according to either **Model A** or **Model B**. Then, ILV with \mathcal{L}^2 neighborhoods converges with probability 1 to the societal optimal point.

The intuition for the result is as follows: as long as the neighborhood does not contain the ideal point of the sampled voter, the correct response to query (1) under weighted Euclidean preferences is to move the solution in the direction of the ideal point to the neighborhood boundary, which, as it turns out, is the same as the direction of the gradient. Thus with radius r_t , the effective movement is $\frac{\nabla f_v(x_t)}{\|\nabla f_v(x_t)\|_2}$. With (normalized) weighted Euclidean utilities, $\|\nabla f_v(x_t)\|_2 = 1$ everywhere. As before, even if the utilities were not normalized (i.e. not divided by $\|w\|_2$), the algorithm would converge to the same point, as if utility functions were normalized.

5.3.2 Decomposable Utilities

Next consider the general class of decomposable utilities, motivated by the fact that the algorithm with \mathcal{L}^∞ neighborhoods is of special interest since they are easy for humans to understand: one can change each dimension up to a certain amount, independent of the others.

Definition 5.3.3. Decomposable utilities. *A voter utility function is decomposable if there exists concave functions f_v^m for $m \in \{1 \dots M\}$ such that $f_v(x) = \sum_{m=1}^M f_v^m(x^m)$.*

If the utility functions for the voters are decomposable, then we can show that our algorithm under \mathcal{L}^∞ neighborhoods converges to the vector of medians of voters' ideal points on each dimension. Suppose that $h_{\mathcal{X}}^m$ is the marginal density function of the random variable x_v^m , and let \bar{x}^m be the set of medians of x_v^m . (By *set* of medians, we mean the set of points such that, on each dimension, the mass of voters with ideal points above and below.)

Suppose that conditions \mathbf{C}_1 , \mathbf{C}_2 , and \mathbf{C}_3 are satisfied, the voter utilities are decomposable, and voters respond to query (1) according to either **Model A** or **Model B**. Then, ILV with \mathcal{L}^∞ neighborhoods converges with probability 1 to a point in the set of medians \bar{x} .

Although simply eliciting each agent's optimal solution and computing the vector of median allocations on each dimension is a viable approach in the case of decomposable utilities, deciding an optimal allocation across multiple dimensions is a more challenging cognitive task than deciding

whether one wants to increase or decrease each dimension relative to the current solution (see Section 5.5.2 for experimental evidence). In fact, in this case, the algorithm can be run separately for each dimension, so that each voter expresses her preferences on only one dimension, drastically reducing the cognitive burden of decision-making on the voter, especially in high dimensional settings like budgeting.

5.3.3 Equivalence to Directional Equilibrium

As discussed in Section 5.2, our algorithm, with \mathcal{L}^2 -norm neighborhoods, is related to an algorithm, NGA, to find what are called Directional Equilibria in literature. Prior work mostly focuses on the properties of the fixed point, with discussion of the proposed algorithm limited to simulations. We show that with the radius decreasing as $\mathcal{O}(\frac{1}{t})$, the algorithm indeed finds directional equilibria in the following sense: if under a few conditions a trajectory of the algorithm converges to a point, then that point is a directional equilibrium.

Theorem 5.3.3. *Suppose that \mathbf{C}_1 , \mathbf{C}_2 , and \mathbf{C}_3 are satisfied, and let $G(x) \triangleq \mathbb{E}_v \left[\frac{\nabla f_v(x)}{\|\nabla f_v(x)\|_2} \right]$. Suppose, $G(x)$ is uniformly continuous, \mathcal{L}^2 movement norm constraints are used, and voters move according to **Model B**. If a trajectory $\{x\}_{t=1}^\infty$ of the algorithm converges to x^* , i.e. $x_t \rightarrow x^*$, then x^* is a directional equilibrium, i.e. $G(x^*) = 0$.*

The proof is in the appendix. It relies heavily on the continuity assumption: if a point x is not a directional equilibrium, then the algorithm with step sizes $\mathcal{O}(\frac{1}{t})$ will with probability 1 leave any small region surrounding x : the net drift of the voter movements is away from the region. We note that the necessary assumptions hold for all utility functions for which convergence holds, using the \mathcal{L}^2 norm algorithm (e.g. weighted Euclidean utilities). It is further possible to characterize other utility functions for which the equivalence holds: with appropriate conditions on the distribution of voters and how f differs among voters, the conditions on G can be met.

We further conjecture that even under voter Model A, if Algorithm 4 converges, the fixed point is a Directional Equilibrium. Note that as $r_t \rightarrow 0$, $f_v(y)$ can be linearly approximated by the first term of the Taylor series expansion around x , for $y \in \{s : \|s - x\|_2 \leq r_t\}$. Then, to maximize $f_v(y)$ in the region, if the region does not contain x_v voter v chooses y^* s.t. $y^* - x \approx r_t \frac{\nabla f(x)}{\|\nabla f(x)\|_2}$, i.e., the voter moves the solution approximately in the direction of her gradient to the neighborhood boundary.

A single step of our algorithm with \mathcal{L}^2 neighborhoods is similar to Quadratic Voting (Lalley and Weyl, 2015; Tideman and Plassmann, 2016) for the same reason. Independently of our work, Benjamin et al. (2017) formalize the relationship between the Normalized Gradient Ascent mechanism and Quadratic Voting.

5.4 Experiments with Budgets

We built a voting platform and ran a large scale experiment, along with several extensive pilots, on Amazon Mechanical Turk (<https://www.mturk.com>). Over 4,000 workers participating in total counting pilots and the final experiment, with over 2,000 workers participating in the final experiment. The design challenges we faced and voter feedback we received provide important lessons for deploying such systems in a real-world setting.

First we present a theoretical model for our setting. We consider a budget allocation problem on M items, where the items may include both expenditures and incomes. One possibility is to define \mathcal{X} as the space of feasible allocations, such as those below a spending limit, and to run the algorithm as defined, with projections. However, in such cases, it may be difficult to theorize about how voters behave; e.g. if voters knew their answers would be projected onto a budget balanced set, they may respond differently.

Rather, we consider an unconstrained budget allocation problem, one in which a voter’s utility includes a term for the budget deficit. Let $\mathcal{E} \subseteq \{1 \dots M\}$, $\mathcal{I} = \{1 \dots M\} \setminus \mathcal{E}$ be the expenditure and income items, respectively. Then the general *budget utility function* is $f_v(x) = g_v(x) - d(\sum_{e \in \mathcal{E}} x^e - \sum_{i \in \mathcal{I}} x^i)$, where d is an increasing function on the deficit.

For example, suppose a voter’s disutility was proportional to the *square* of the budget deficit (she especially dislikes large budget deficits); then, this term adds complex dependencies between the budget items. In general, nothing is known about convergence of Algorithm 2 with such utilities, as the deficit term may add complex dependencies between the dimensions. However, if the voter utility functions are decomposable across the dimensions and \mathcal{L}^∞ neighborhoods used, then the results of Section 5.3.2 can be applied. We propose the following class of decomposable utility functions for the budgeting problem, achieved by assuming that the cost for the deficit is linear, and call the class “decomposable with a linear cost for deficit,” or DLCD.

Definition 5.4.1. *Let $f_v(x)$ be DLCD if*

$$f_v(x) = \sum_{m=1}^M f_v^m(x^m) - w_v \left(\sum_{e \in \mathcal{E}} x^e - \sum_{i \in \mathcal{I}} x^i \right),$$

where f_v^m is a concave function for each m and $w_v \in \mathbb{R}_+$.

In the experiments discussed below in the budget setting, ILV consistently and robustly converges with \mathcal{L}^∞ norm neighborhoods. Further, it approximately converges to the medians of the optimal solutions (which are elicited independently), as theorized in Section 5.3.2. Such a convergence pattern suggests the validity of the DLCD model, though we do not formally analyze this claim.

5.4.1 Experimental Setup

We asked voters to vote on the U.S. Federal Budget across several of its major categories: National Defense; Healthcare; Transportation, Science, & Education; and Individual Income Tax (Note that the US Federal Government cannot just decide to set tax receipts to some value. We asked workers to assume tax rates would be increased or decreased at proportional rates in hopes of affecting receipts.)

This setting was deemed the most likely to be meaningful to the largest cross-section of workers and to yield a diversity of opinion, and we consider budgets a prime application area in general. The specific categories were chosen because they make up a substantial portion of the budget and are among the most-discussed items in American politics. We make no normative claims about running a vote in this setting in reality, and Participatory Budgeting has historically been more successful at a local level.

One major concern was that with no way to validate that a worker actually performed the task (since no or little movement is a valid response if the solution presented to the worker was near her ideal budget), we may not receive high-quality responses. This issue is especially important in our setting because a worker’s actions influence the initial solution future workers see. We thus restricted the experiment to workers with a high approval rate and who have completed over 500 tasks on Mechanical Turk (MTurk). Further, we offered a bonus to workers for justifying their movements well, and more than 80% of workers qualified, suggesting that we also received high-quality movements. The experiment was restricted to Americans to best ensure familiarity with the setting. Turkprime (<https://www.turkprime.com>) was used to manage postings and payment.

5.4.2 Experimental Parameters

Our large scale experiment included 2,000 workers and ran over a week in real-time. Participants of any of the pilots were excluded. We tested the \mathcal{L}^1 , \mathcal{L}^2 , and \mathcal{L}^∞ mechanisms, along with a “full elicitation” mechanism in which workers reported their ideal values for each item, and a “weight” in $[0, 10]$ indicating how much they cared about the actual spending in that item being close to their stated value.

To test repeatability of convergence, each of the constrained mechanisms had three copies, given to three separate groups of people. Each group consisted of two sets with different starting points, with each worker being asked to vote in each set in her assigned group. Each worker only participates as part of one group, and cannot vote multiple times.

We used a total of three different sets of starting points across the three groups, such that each group shared one set of starting points with each of the other two groups. This setup allowed testing for repeatability across different starting points and observing each worker’s behavior at two points. Workers in one group in each constrained mechanism type were also asked to do the full elicitation after submitting their movements for the constrained mechanism, and such workers were paid extra.

These copies, along with the full elicitation, resulted in 10 different mechanism instances to which workers could be allocated, each completed by about 200 workers.

To update the current point, we waited for 10 submissions and then updated the point to their average. This averaging explains the step-like structure in the convergence plots in the next section. The radius was decreased approximately every 60 submissions, $r_t \approx \frac{r_0}{\lceil t/60 \rceil}$. The averaging and slow radius decay rate were implemented in response to observing in the pilots that the initial few voters with a large radius had a disproportionately high impact, as there were not enough subsequent voters to recover from large initial movements away from an eventual fixed point (though in theory this would not be a problem given enough voters). We note that the convergence results for stochastic subgradient methods trivially extend to cover these modifications: the average movement over a batch of submissions starting at the same point is still in expectation a subgradient, and the stepped radius decrease still meets the conditions for valid step-sizes.

5.4.3 User Experience

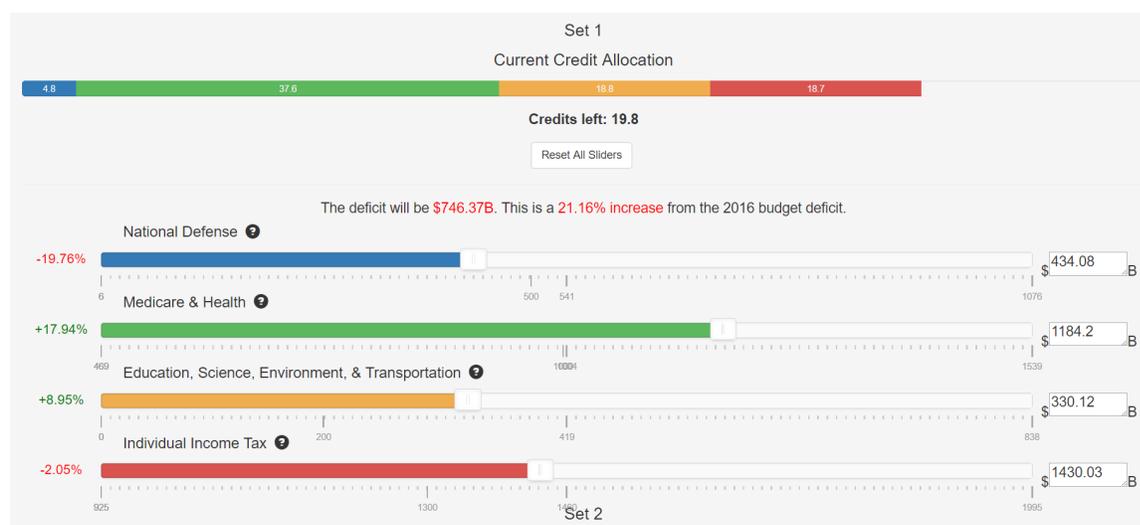


Figure 5.1: UI Screenshot for 1 set of the \mathcal{L}^2 Mechanism

As workers arrived, they were randomly assigned to a mechanism instance. They had a roughly equal probability of being assigned to each instance, with slight deviations in case an instance was “busy” (another user was currently doing the potential 10th submission before an update of the instance’s current point) and to keep the number of workers in each instance balanced. Upon starting, workers were shown mechanism instructions. We showed the instructions on a separate page so as to be able to separately measure the time it takes to read & understand a given mechanism, and the time it takes to do it, but we repeated the instructions on the actual mechanism page as well for reference.

On the mechanism page, workers were shown the current allocation for each of the two sets in their group. They could then move, through sliders, to their favorite allocation under the movement constraint. We explained the movement constraints in text and also automatically calculated for them the number of “credits” their current movements were using, and how many they had left. Next to each budget item, we displayed the percentage difference of the current value from the 2016 baseline federal budget, providing important context to workers (The 2016 budget estimate was obtained from <http://federal-budget.insidegov.com/1/119/2016-Estimate> and <http://atlas.newamerica.org/education-federal-budget>). We also provided short descriptions of what goes into each budget item as scroll-over text. The resulting budget deficit and its percent change were displayed above the sliders, assuming other budget items are held constant.

For the full elicitation mechanism, workers were asked to move the sliders to their favorite points with no constraints (the sliders went from \$0 to twice the 2016 value in that category), and then were asked for their “weights” on each budget item, including the deficit. Figure 5.1 shows part of the interface for the \mathcal{L}^2 mechanism, not including instructions, with similar interfaces for the other constrained mechanisms. The full elicitation mechanism additionally included sliders for items’ weights. On the final page, workers were asked for feedback on the experiment.

A full walk-through of the experiment with screenshots and link to an online demo is available in the Appendix. We plan on posting the data, including feedback. In general, workers seemed to like the experiment, though some complained about the constraints, and others were generally confused. Some expressed excitement about being asked their views in an innovative manner and suggested that everyone could benefit from participating as, at the least, a thought exercise. The feedback and explanations provided by workers were much longer than we anticipated, and they convince us of the procedure’s civic engagement benefits.

5.5 Results and Analysis

We now discuss the results of our experiments.

5.5.1 Convergence

One basic test of a voting mechanism is whether it produces a consistent and unique solution, given a voting population and their behaviors. If an election process can produce multiple, distinct solutions purely by chance, opponents can assail any particular solution as a fluke and call for a re-vote. The question of whether the mechanisms consistently converge to the same point thus must be answered before analyzing properties of the equilibrium point itself. In this section, we show that the \mathcal{L}^2 and \mathcal{L}^1 algorithms do not appear to converge to a unique point, while the \mathcal{L}^∞ mechanism converges to a unique point across several initial points and with distinct worker populations.

The solutions after each voter for each set of starting points, across the 3 separate groups of

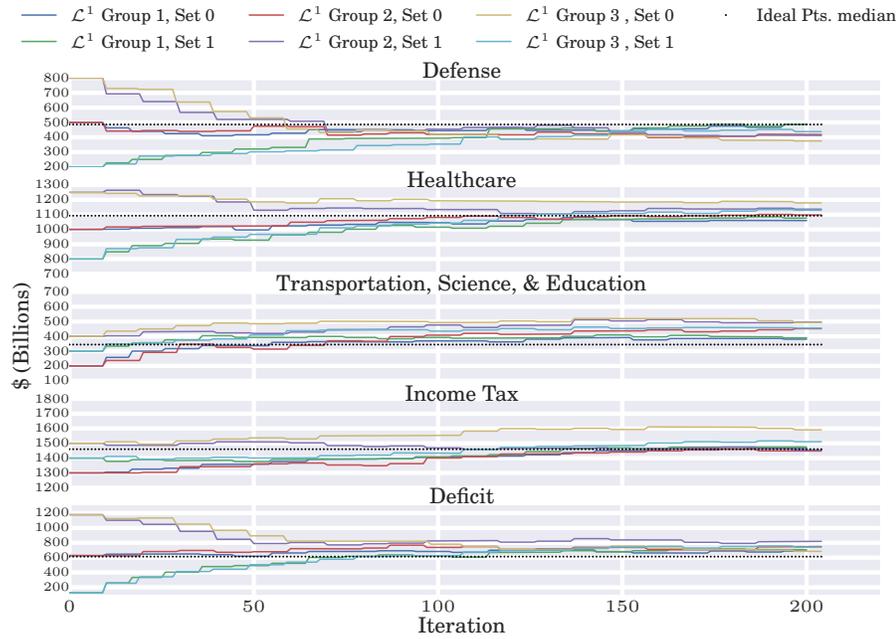
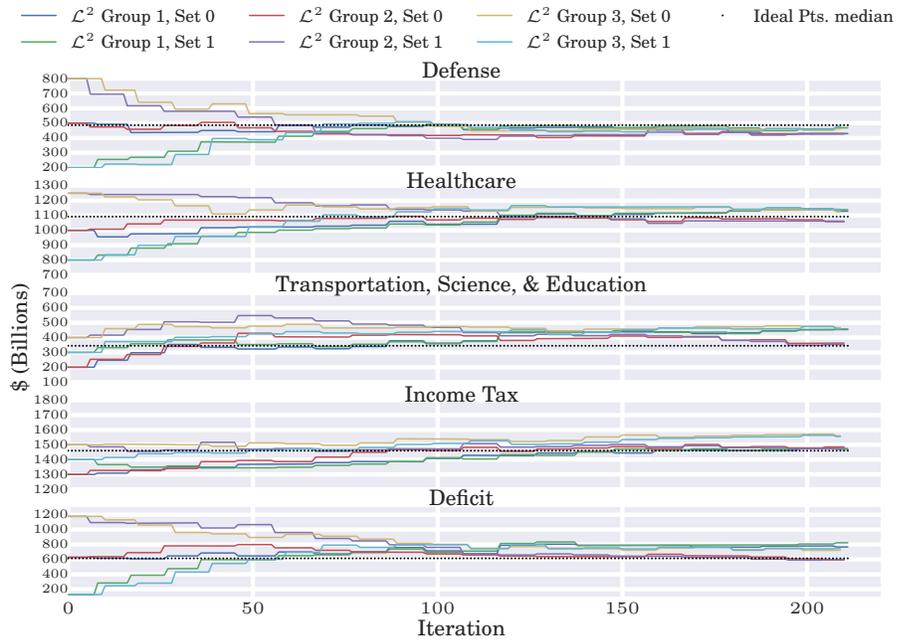
(a) \mathcal{L}^1

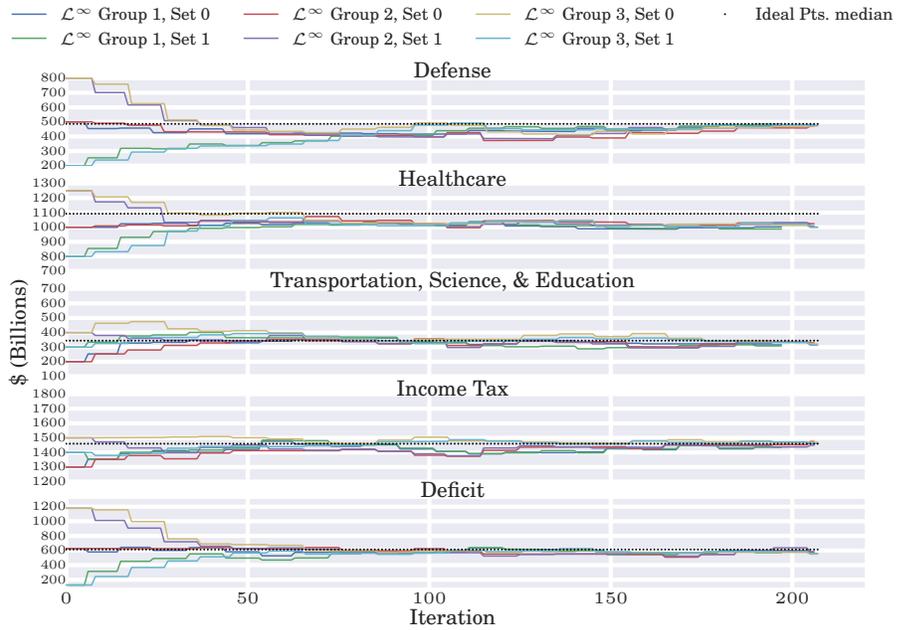
Figure 5.2: Solution over time for each mechanism type

people for each constrained mechanism are shown in Figure 5.2. Each plot shows all the trajectories with the given mechanism type, along with the median of the ideal points elicited from the separate voters who only performed the full elicitation mechanism. Observe that the three mechanisms have remarkably different convergence patterns. In the \mathcal{L}^1 mechanism, not even the sets done by the same group of voters (in the same order) converged in all cases. In some cases, they converged for some budget items but then diverged again. In the \mathcal{L}^2 mechanism, sets done by the same voters starting from separate starting points appear to converge, but the three groups of voters seem to have settled at two separate equilibria in each dimension. Under the \mathcal{L}^∞ neighborhood, on the other hand, all six trajectories, performed by three groups of people, converged to the same allocation very quickly and remained together throughout the course of the experiment. Furthermore, the final points, in all dimensions except Healthcare, correspond almost exactly to the median of values elicited from the separate set of voters who did only the full elicitation mechanism. For Healthcare, the discrepancy could result from biases in full elicitation (see Section 5.5.2), though we make no definitive claims. These patterns shed initial insight on how the use of \mathcal{L}^2 constraints may differ from theory in prior literature and offer justification for the use of DLCDC utility models and the \mathcal{L}^∞ constrained mechanism.

One natural question is whether these mechanisms really have converged, or whether if we let



(b) \mathcal{L}^2



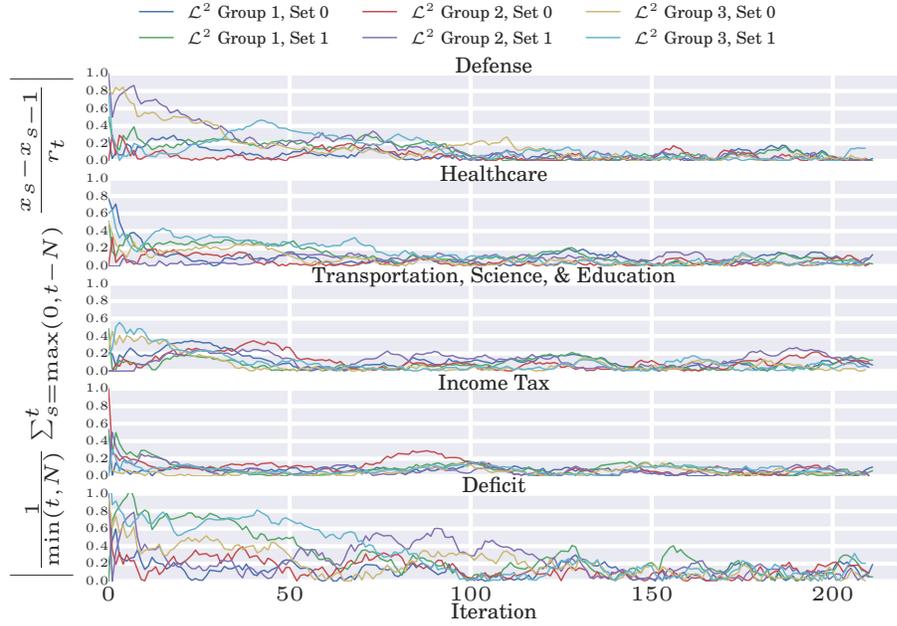
(c) \mathcal{L}^∞

Figure 5.2: (Continued) Solution over time for each mechanism type

(a) \mathcal{L}^1 Figure 5.3: Net normalized movement in window of $N = 30$

the experiment continue, the results would change. This question is especially salient for the \mathcal{L}^2 trajectories, where trajectories within a group of people converged to the same point, but trajectories between groups did not. Such a pattern could suggest that our results are a consequence of the radius decreasing too quickly over time, or that the groups had, by chance, different distributions of voters which would have been corrected with more voters. However, we argue that such does not seem to be the case, and that the mechanism truly found different equilibria.

We can test whether the final points for each trajectory are stable by checking the net movement in a window, normalized by each voter's radius, i.e. $\frac{1}{N} \sum_{s=t-N}^t \frac{x_s - x_{s-1}}{r_s}$, for some N . If voters in a window are canceling each other's movements, then this value would go to 0, and the algorithm would be stable even if the radius does not decrease. The notion is thus robust to apparent convergence just due to decreasing radii. The net movement normalized in a sliding window of 30 voters, for each dimension and mechanism, is shown in Figure 5.3. It seems to almost die down for almost all mechanisms and budget items, except for a few cases which do not change the result. We conclude it likely that the mechanisms have settled into equilibria which are unlikely to change given more voters.



(b) \mathcal{L}^2



(c) \mathcal{L}^∞

Figure 5.3: (Continued) Net normalized movement in window of $N = 30$.

5.5.2 Understanding Voter Behavior

A mechanism's practical impact depends on more than whether it consistently converges, however. We now turn our attention to understanding how voters behave under each mechanism and whether we can learn anything about their utility functions from that behavior. We find that voters understood the mechanisms but that their behaviors suggest large indifference regions, and that the full elicitation scheme is susceptible to biases that can skew the results.

Voter understanding of mechanisms

One important question is whether, given very little instruction on how to behave, voters understand the mechanisms and act approximately optimally under their (unknown to us) utility function. This section shows that the voters behavior follows what one would expect in one important respect: how much of one's movement budget the voter used on each dimension, given the constraint type.

Regardless of the exact form of the utility function, one would expect that, in the \mathcal{L}^1 constrained mechanism, a voter would use most of her movement credits in the dimension about which she cares most. In fact, in either the Weighted Euclidean preferences case (and with 'sub-space' being a single dimension) or with a small radius with \mathcal{L}^1 constraints, a voter would move only on one dimension. With \mathcal{L}^2 constraints, one would expect a voter to apportion her movement more equally because she pays an increasing marginal cost to move more in one dimension (people were explicitly informed of this consequence in the instructions). Under the Weighted Euclidean preferences model with \mathcal{L}^2 constraints, a voter would move in each dimension proportional to her weight in that dimension. Finally, with \mathcal{L}^∞ constraints, a voter would move, in all dimensions in which she is not indifferent, to her favorite point in the neighborhood for that dimension (most likely an endpoint), independently of other dimensions. One would thus expect a more equal distribution of movements.

Figure 5.4 shows the average movement (as a fraction of the voter's total movement) by each voter for the dimension she moved most, second, third, and fourth, respectively, for each constrained mechanism. We reserve discussion of the full elicitation weights for Section 5.5.2. The movement patterns indicate that voters understood the constraints and moved accordingly – with more equal movements across dimensions in \mathcal{L}^2 than in \mathcal{L}^1 , and more equal movements still in \mathcal{L}^∞ . We dig deeper into user utility functions next, but can conclude that, regardless of their exact utility functions, voters responded to the constraint sets appropriately.

Large indifference regions

Although it is difficult to extract a voter's full utility function from their movements, the separability of dimensions (except through the deficit term) under the \mathcal{L}^∞ constraint allows us to test whether voters behave according to some given utility model in that dimension, without worrying about the dependency on other dimensions.

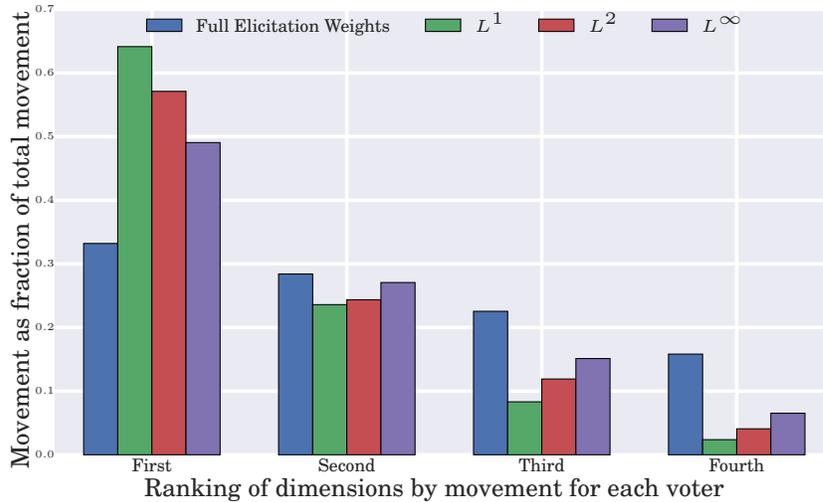


Figure 5.4: Average movement in dimension over total movement for each voter, with dimensions sorted

Figure 5.5 shows, for the \mathcal{L}^∞ mechanism, a histogram of the movement on a dimension as a fraction of the radius (we find no difference between dimensions here). Note that a large percentage of voters moved very little on a dimension, even in cases where their ideal point in that dimension was far away (defined as being unreachable under the current radius). This result cannot be explained away by workers clicking through without performing the task: almost all workers moved at least one dimension, and, given that a worker moved in a given dimension, it would not explain smaller movements being more common than larger movements. That this pattern occurs in the \mathcal{L}^∞ mechanism is key – if a voter feels *any* marginal disutility in a dimension, she can move the allocation without paying a cost of more limited movement in other dimensions. We conclude that, though voters may share a single ideal point for a dimension when asked for it, they are in fact relatively indifferent over a potentially large region – and their actions reflect so.

We further analyze this claim in Appendix Section D.2, looking at the same distribution of movement but focusing on workers who provided a text explanation longer than (and shorter than, separately) the median explanation of 197 characters. (We assume that the voters who invested time in providing a more thorough explanation than the average worker also invested time in moving the sliders to a satisfactory point, though this assumption cannot be validated.) Though there are some differences (those who provide longer explanations also tend to use more of their movement), the general pattern remains the same; only about 40% of workers who provided a long explanation and were far away from their ideal point on a dimension used the full movement budget. This pattern suggests that voters are relatively indifferent over large regions.

Furthermore, this lack of movement is correlated with a voter’s weights when she was also asked

to do the full elicitation mechanism. Conditioned on being far from her ideal point, when a voter ranked an item as one of her top two important items (not counting the deficit term), she moved an average of 74% of her allowed movement in that dimension; when she ranked an item as one of least two important items, she moved an average of 61%, and the difference is significant through a two sample t-test with $p = .013$. We find no significant difference in movement within the top two ranked items or within the bottom two ranked items. This connection suggests that one can potentially determine which dimensions a voter cares about by observing these indifference regions and movements, even in the \mathcal{L}^∞ constrained case. One caveat is that the differences in effects are not large, and so at the individual level inference of how much an individual cares about one dimension over another may be noisy. On the aggregate, however, such determination may prove useful.

Furthermore, we note that while such indifference regions conflict with the utility models under which the \mathcal{L}^2 constraint mechanism converges in theory, it fits within the DLCDC framework introduced in Section 5.4.

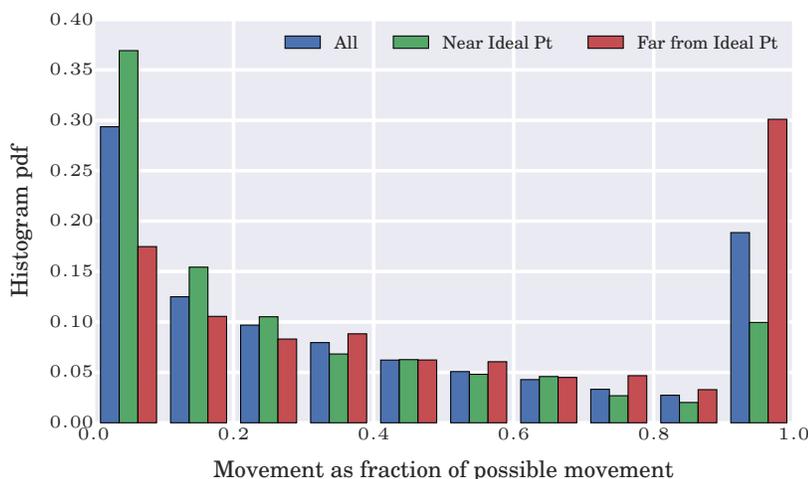


Figure 5.5: Fraction of possible movement in each dimension in \mathcal{L}^∞ , conditioned on distance to ideal pt. The ‘All’ condition contains data from all three \mathcal{L}^∞ instances, whereas the others only from the instance that also did full elicitation.

Mechanism time

In this section, we note one potential problem with schemes that explicitly elicit voter’s optimal solutions – for instance, to find the component-wise median – as compared to the constrained elicitation used in ILV: it seems to be cognitively difficult for voters. In Figure 5.6, the median time per page, aggregated across each mechanism type, is shown. The “Mechanism” time includes a single user completing both sets in each of the constrained mechanism types, but not does include the time

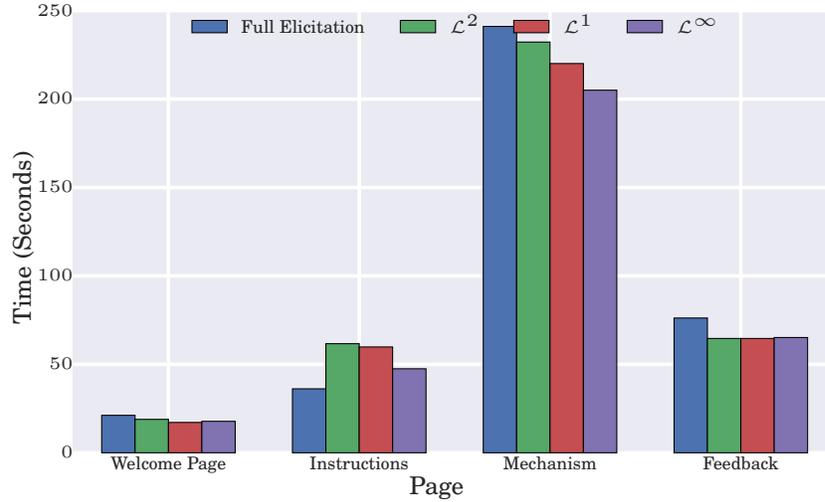


Figure 5.6: Median time per page

to also do the extra full elicitation task in cases where a voter was asked to do both a constrained mechanism and the full elicitation. The full elicitation bars include only voters who did only the full elicitation mechanism, and so the bars are completely independent. On average, it took longer to do the full elicitation mechanism than it took to do *two* sets of any of the constrained mechanisms, suggesting some level of cognitive difficulty in articulating one’s ideal points and weights on each dimension – even though understanding what the instructions are asking was simple, as demonstrated by the shorter instruction reading time for the full elicitation mechanism. The \mathcal{L}^∞ mechanism took the least time to both understand and do, while the \mathcal{L}^2 mechanism took the longest to do, among the constrained mechanisms.

This result is intuitive: it is easier to move each budget item independently when the maximum movement is bounded than it is to move the items when the sum or the sum of the changes squared is bounded (even when these values are calculated for the voter). In practice, with potentially tens of items on which constituents are voting, these relative time differences would grow even larger, potentially rendering full elicitation or \mathcal{L}^2 constraints unpalatable to voters.

One potential caveat to this finding is that the Full Elicitation mechanism potentially provides more information than do the other mechanisms. From a polling perspective, it is true that more information is provided from full elicitation – one can see the distribution of votes, the disagreement, and correlation across issues, among other things. However, from a voting perspective, in which the aggregation (winner) is the only thing reported, it is not clear that this extra information is useful. Further, much of this information that full elicitation provides can reasonably be extracted from movements of voters, especially the movements of those who are given a starting point close to the eventual equilibrium.

UI biases

We now turn our attention to the question of how workers behaved under the full elicitation mechanism and highlight some potential problems that may affect results in real deployments. Figures 5.7 and 5.8 show the histogram of values and weights, respectively, elicited from all workers who did the full elicitation mechanism. Note that in the histogram of values, in every dimension, the largest peak is at the slider’s default value (at the 2016 estimated budget), and the histograms seem to undergo a phase shift at that peak, suggesting that voters are strongly anchored at the slider’s starting value. This anchoring could systematically bias the medians of the elicited values.

A similar effect occurs in eliciting voter weights on each dimension. Observe that in Figure 5.4 the full elicitation weights appear far more balanced than the weights implied by any of the mechanisms (for the full elicitation mechanism, the plot shows the average weight over the sum of the weights for each voter). From the histogram of full elicitation weights, however, we see that this result is a consequence of voters rarely moving a dimension’s weight down from the default of 5, but rather moving others up.

One potential cause of this behavior is that voters might think that putting high weights on each dimension would mean their opinions would count more, whereas in any aggregation one would either ignore the weights (calculate the unweighted median) or normalize the weights before aggregating. In future work, one potential fix could be to add a “normalize” button for the weights, which would re-normalize the weights, or to automatically normalize the weights as voters move the sliders.

These patterns demonstrate the difficulty in eliciting utilities from voters directly; even asking voters how much they care about a particular budget item is extremely susceptible to the user interface design. Though such anchoring to the slider default undoubtedly also occurs in the \mathcal{L}^∞ constrained mechanism, it would only slow the rate of convergence, assuming the anchoring affects different voters similarly. These biases can potentially be overcome by changing the UI design, such as by providing no default value through sliders. Such design choices must be carefully thought through before deploying real systems, as they can have serious consequences.

5.6 Conclusion

We evaluate a natural class of iterative algorithms for collective decision-making in continuous spaces that makes practically reasonable assumptions on the nature of human feedback. We first introduce several cases in which the algorithm converges to the societal optimum point, and others in which the algorithm converges to other interesting solutions. We then experimentally test such algorithms in the first work to deploy such a scheme. Our findings are significant: even with theoretical backing, two variants fail the basic test of being able to give a consistent decision across multiple trials with the same set of voters. On the other hand, a variant that uses \mathcal{L}^∞ neighborhoods consistently leads to convergence to the same solution, which has attractive properties under a likely model for voter

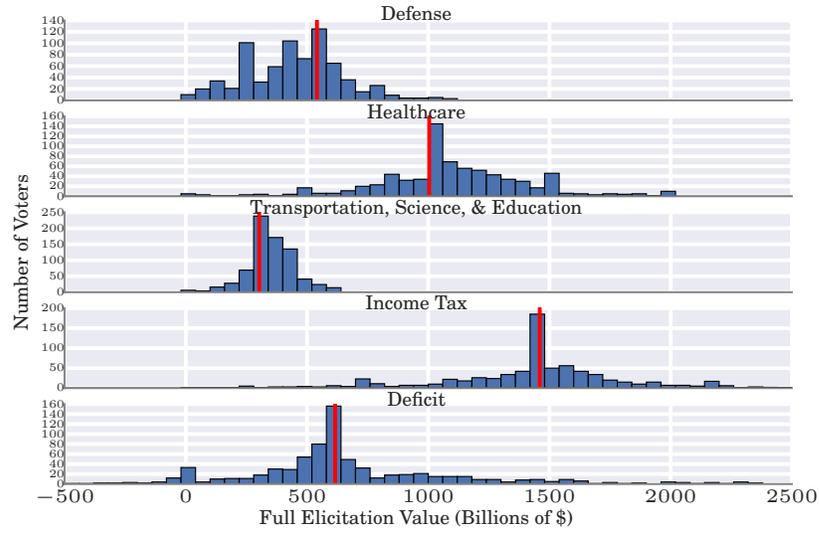


Figure 5.7: Histogram of values from all full elicitation data. The red vertical lines indicate each slider’s default value (at the 2016 estimated budget).

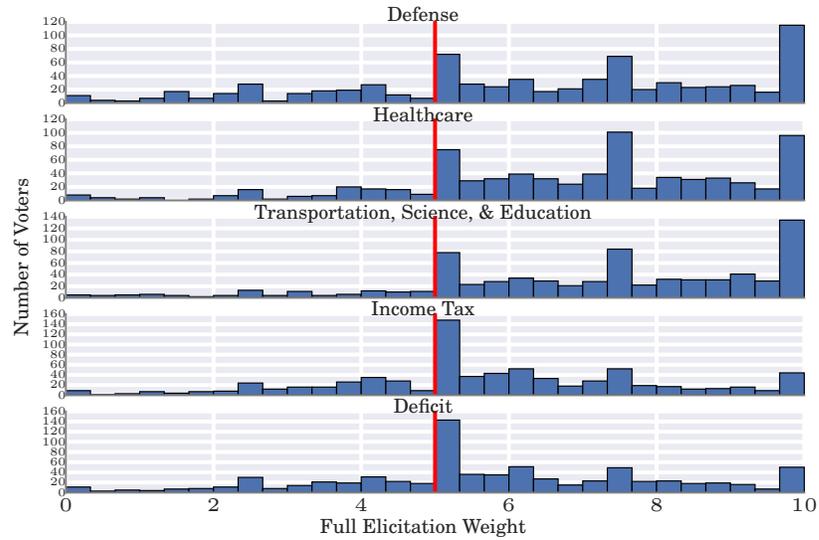


Figure 5.8: Histogram of weights from all full elicitation data. The red vertical lines indicate the sliders’ default value of 5.

preferences suggested by this convergence. We also make certain observations about other properties of user preferences – most saliently, that they have large indifferences on dimensions about which they care less.

In general, this work takes a significant step within the broad research agenda of understanding the fundamental limitations on the quality of societal outcomes posed by the constraints of human feedback, and in designing innovative mechanisms that leverage this feedback optimally to obtain the best achievable outcomes.

Chapter 6

Who is in Your Top Three? Optimizing Learning in Elections with Many Candidates

6.1 Introduction

Elections and opinion polls with many candidates and multiple winners are common. In participatory budgeting (PB), for example, people directly determine a part of the government’s budget (Alós-Ferrer and Granić, 2012; Goel et al., 2016). These elections often contain many candidate projects (up to 70, cf. Gelauff et al. (2018)) and only a few thousand voters, with potentially millions of dollars on the line (Public Agenda, 2016). Similarly, polls may compare tens of candidates and yet only sample hundreds of voters.

Unfortunately, the number of voters required to recover the asymptotic ranking or set of winners often scales, potentially exponentially, with the number of candidates (Caragiannis and Micha, 2017). Thus with many candidates, it is essential to use a voting mechanism that most efficiently elicits information from each voter.

In this work, we analyze *positional scoring rules* (de Borda, 1781; Young, 1975), mechanisms in which each position in each voter’s personal ranking maps to a score given to the candidate that

This chapter is joint with Lodewijk Gelauff, Sukolsak Sakshuwong, and Ashish Goel. Published at the AAAI Conference on Human Computation and Crowdsourcing (HCOMP), in 2019 (Garg et al., 2019a). We thank our Participatory Budgeting city partners, especially those in Boston, Durham, and Rochester. We also thank anonymous reviewers for their comments. This work was supported in part by the Stanford Cyber Initiative, the Office of Naval Research grant N00014-19-1-2268, and National Science Foundation grants 1544548 and 1637397.

occupies that position. We focus on the special cases of such rules implied by K -Approval elicitation, in which each voter is asked to select their favorite K candidates, as they are the most commonly used such mechanisms in practice. Section 6.3 formalizes our model. Then:

Section 6.4. For a given election, we show how the particular scoring rule used affects the rate at which the final outcome (asymptotic in the number of voters) is learned. These rates, based on large deviation bounds, extend and tighten the results of Caragiannis and Micha (2017), and are precise enough to determine, for example, which of 3-Approval and 4-Approval is better in a particular context. We focus on the goals of learning both a ranking over all candidates and identifying a subset of winners.

Section 6.5.1. Leveraging these rates, we study when randomization between scoring rules can improve learning, extending previous results to general positional scoring rules, the goal of selecting a set of winners, and arbitrary noise models. In particular, we find that randomizing between scoring rules can never speed up learning, for arbitrary noise models. This result differs from case when one is restricted to K -Approval mechanisms.

Section 6.5.2. For the Mallows model, we study how the optimal K in K -Approval scales with the noise parameter, the number of candidates, and the number of winners desired. We find that, in contrast to design choices made in practice, one should potentially ask voters to identify up to their favorite *half* of candidates, even if the goal is to identify a *single* winner.

Section 6.6. We apply our approach to experimental ballots attached to real participatory budgeting elections across several US cities, as well as other ranking data from a range of domains. We find that the exact mechanism used matters: in one setting, for example, asking voters to identify their favorite candidate results in only a 80% chance of identifying the best candidate after 400 voters, while asking voters for their favorite 2 candidates identifies the same best candidate 99.9% of the time. Extending our theoretical insights, we find that, historically across elections, K has been set too low for effective learning. We also identify examples in which randomization among K -Approval mechanisms would have sped up learning.

Our work bridges a gap between coarse theoretical analyses of voting rules and the fine-grained design questions a practitioner wishes to answer. Proofs are in the Appendix.

6.2 Related work

Our work is part of several strands of research on mechanisms that elicit peoples' preferences. Aggregating voter rankings has a long history (Copeland, 1951; de Borda, 1781; Kemeny, 1959; marquis de Condorcet, 1785; Young, 1988).

Learning properties of voting rules Most related are works that study the learning properties of voting rules, assuming that a “true” ranking exists. One approach is to specify a noise model under which voter preferences are drawn (e.g., Mallows, Plackett-Luce) and then derive error rates by the number of voters for maximum likelihood or other estimators under the model (Chierichetti and Kleinberg, 2014; de Weerd et al., 2016; Guiver and Snelson, 2009; Lu and Boutilier, 2011; Maystre and Grossglauser, 2015; Procaccia and Shah, 2015; Zhao et al., 2016).

Caragiannis et al. (2013) ask similar questions to us: under what voter noise models do certain voting rules asymptotically recover the true underlying ranking, and how quickly do they do so. They define a class of voting rules and voter noise models under which a “true” ranking of candidates is eventually recovered. They further show that for a subset of this class (that does not contain positional scoring rules) and under the Mallows model, only a number of voters that is logarithmic in the number of candidates is required, where each voter provides a full ranking. Lee et al. (2014) develop an algorithm that can approximate the Borda rule, given a number of comparisons by each voter that is logarithmic in the number of candidates.

Most similar is that of Caragiannis and Micha (2017). They show that under the Mallows model, K -Approval with any fixed K takes exponentially many voters (in the number of candidates) to recover the underlying ranking; on the other hand, K -approval with K chosen uniformly at random for each voter takes only a polynomial number of voters.

These works provide order estimates for the learning rate, *asymptotic in the number of candidates*; fine-grained differentiation between different rules or K -Approval mechanisms for a given election is not possible. We provide the latter and show that it matters.

Other approaches to comparing mechanisms Many works take an axiomatic and computational approach, comparing mechanisms that may produce different outcomes even given asymptotically many votes (Aziz et al., 2015, 2017; Caragiannis et al., 2017; Elkind et al., 2017; Faliszewski and Talmon, 2018; Fishburn, 1978; Fishburn and Gehrlein, 1976; Lackner and Skowron, 2018a,b; Ratliff, 2003; Staring, 1986; Tataru and Merlin, 1997; Wiseman, 2000). Caragiannis et al. (2019) for example show how to find a scoring rule that most agrees with a given partial ground truth ranking. In contrast, we compare mechanisms’ learning rates under a condition (in Section 6.3.2) in which they produce the same asymptotic outcome.

Benade et al. (2018) and Gelauff et al. (2018) experimentally compare different mechanisms across several dimensions, including ease of use and consistency with another mechanism; the latter leverages data from a participatory budgeting election at a university.

Large deviation analysis of elicitation mechanisms Theoretically, we leverage large deviation rates and Chernoff bounds to derive how quickly a given scoring rule learns its outcome; see work of Dembo and Zeitouni (2010) for an introduction to large deviations. This work is thus conceptually similar to work on elicitation design for rating systems, discussed in Part II. There, we derive large

deviation-based learning rates that depend on the questions that are asked to buyers as they review an item, where the goal is to accurately rank items; they further run an experiment on an online labor platform. In that setting, however, buyers rate a *single* item, and mechanisms are distinct based on the *behavior* they induce; in this work, voters see all the candidates and provide a partial ordering, and different designs (e.g., 3-Approval vs 4-Approval) *constrain* the types of orderings voters can provide.

6.3 Model

We now present our model and a condition under which different positional scoring rules induce the same asymptotic outcome.

6.3.1 Model primitives

We begin with the model primitives: candidates and voters, the election goal, and elicitation and aggregation.

Candidates and Voters There is a set of M candidates $C = \{1, \dots, M\}$, typically indexed by $i, j \in C$. There are N voters $V = \{1, \dots, N\}$. Each voter $v \in V$ has a strict ranking of candidates σ_v , drawn independently and identically from probability mass function over strict rankings $F(\sigma)$. Let $i \succ_\sigma j$ denote that i is preferred over j in σ , and $\sigma(i) = k$ denote that candidate i is in the k th position in σ .

A special case for F is the *Mallows* model (Mallows, 1957), in which there is a “true” societal preference σ^* from which each voter’s ranking is a noisy sample. In particular,

$$F_{\text{Mallows}}(\sigma) \propto \phi^{d(\sigma, \sigma^*)}$$

Where $d(\sigma, \sigma^*)$ is the Kendall’s τ distance between rankings σ, σ^* , and $\phi \in [0, 1]$ is the noise parameter: the smaller it is, the more concentrated F is around σ^* .

Election goal We assume that the *goal* G is to divide the candidates into T disjoint, ordered *tiers* $G = \{C_1, \dots, C_T\}$, such that $C = \cup_{t=1}^T C_t$, where candidate $i \in C_s$ is deemed societally preferable over $j \in C_t$ if $s < t$. The size of each tier is fixed before the election. For example, recovering a strict ranking over all candidates corresponds to $G = \{C_1, \dots, C_M\}$, where $|C_t| = 1$. Alternatively, identifying a set of W winners, without distinguishing amongst the winners, corresponds to $G = \{C_1, C_2\}$, with $|C_1| = W$.

In the main text and especially the empirics, we will focus on the task of selecting W winners as it is the most common task in practice. However, this general notation allows comparison of the

learning properties of different settings, and for example asking how much more expensive is it (in terms of the number of voters needed) to identify a strict ranking as opposed to just a set of winners.

Elicitation and Aggregation Voters vote using an elicitation mechanism. Their votes are then aggregated using a positional scoring rule, parameterized as $\beta : \{1, \dots, M\} \mapsto \mathbb{R}$. We consider the following mechanisms:

K -Ranking Voter v ranks her favorite K candidates, i.e., reveals $\{(i, \sigma_v(i)) : \sigma_v(i) \leq K\}$. Candidate i then receives a score $s_{iv} = \beta(\sigma_v(i))$ if ranked, 0 otherwise. For example, $\beta(k) = M - k$ for the Borda count.¹

K -Approval Voter v selects her favorite K candidates, i.e., reveals $\{i : \sigma_v(i) \leq K\}$. A candidate receives a score $s_{iv} = 1$ for being selected, 0 otherwise.

β encodes both elicitation and aggregation. For example, K -Approval is equivalent to K -ranking with score function $\beta(k) = \mathbb{I}[k \leq K]$. Furthermore, note that given K -ranking data, one can simulate K' -ranking elicitation for $K' \leq K$ with a β s.t. $\beta(k) = 0$ for $k > K'$.

The scoring rule β is a design choice made by the election organizer, and so we will refer to β as the election's *design*. We restrict ourselves to non-constant, non-increasing scoring rules, i.e., $\beta \in \mathcal{B} = \{\beta : \forall k < \ell \in 1, \dots, M, \beta(k) \geq \beta(\ell), \text{ and } \exists k < \ell, \beta(k) > \beta(\ell)\}$.

Outcome After N voters, candidate i 's cumulative score is $s_i^N = \frac{1}{N} \sum_{v=1}^N s_{iv}$. Candidates are ranked in descending order of score, to form ranking σ^N , with ties broken uniformly at random. We denote the *outcome* after N voters, corresponding to the goal G , as $O^N(M, F, \beta, G)$. For example, for the goal of selecting W winners, $O^N(M, F, \beta, G)$ is simply the top W candidates in σ^N . When (M, F, β, G) is clear from context, we will refer to the outcome as O^N .

As the number of voters $N \rightarrow \infty$, candidate scores $s_i^N \rightarrow \mathbb{E}_F[s_{iv}] \triangleq s_i$ by the law of large numbers; when such expected scores are distinct, i.e., $s_i \neq s_j$ for $i \neq j$, then $\sigma^N \rightarrow \sigma^*$ for some ranking σ^* . However, note that there may exist an asymptotic outcome $O^N \rightarrow O^*$ even without an asymptotic ranking $\sigma^N \rightarrow \sigma^*$, as long as expected scores s_i and goal G are such that candidates with identical expected scores are sorted into the same tier.

6.3.2 Asymptotic design invariance

The asymptotic outcome O^* of an election may vary with the scoring rule β . For example, there may be a different winner if voters are asked to identify their favorite two candidates than if they identify their single favorite candidate, if the winner in the latter case is a polarizing candidate. As an axiomatic comparison between outcomes is out of the scope of this paper, we restrict our

¹In Borda, candidates not ranked receive a score $(M - K - 1)/2$, consistent with assuming they are all tied in position $(K + 1)$.

attention to cases where all “reasonable” choices of different β asymptotically result in the same outcome (where “reasonable” corresponds to the set of scoring rules \mathcal{B} defined above).

Definition 6.3.1. *A setting (M, F) is asymptotically design-invariant for goal G if any reasonable β induces the same outcome asymptotically. $\exists O^* : \forall \beta \in \mathcal{B}$,*

$$\lim_{N \rightarrow \infty} O^N(M, F, \beta, G) = O^*, \text{ with probability 1}$$

Such design invariance only occurs under a fairly strong condition on the voter preference distribution: that the candidates can be separated into tiers (according to goal G) such that candidates in higher tiers are *strictly* more likely to be ranked by a voter in the top k positions, for all $k < M$, than are candidates in lower tiers.

A setting (M, F) for goal G is asymptotically design-invariant if and only if there exist candidate tiers $O^* = \{C_1^* \dots C_T^*\}$ (corresponding to G) s.t. $\forall s < t: i \in C_s^*, j \in C_t^* \implies \Pr_F(\sigma_v(i) \leq k) > \Pr_F(\sigma_v(j) \leq k), \forall k \in \{1 \dots M - 1\}$.

Note that this condition is stronger than stochastic dominance as the inequality is strict for *every* position k .

This proposition connects to Caragiannis et al. (2013) as follows: they prove that many rules (including all positional scoring rules and the Bucklin rule) asymptotically recover the base ranking σ^* of a generalization of the Mallows model in which the probability $F(\sigma)$ of a ranking σ is monotonic in the distance $d(\sigma, \sigma^*)$, where distance function d is itself in some general class that contains the Kendall’s τ distance. Their results directly imply that such noise models, including the standard Mallows model, are asymptotically design-invariant for any goal G .

However, for goals G where recovering a full ranking is unnecessary, the condition in Proposition 6.3.2 is weaker than the assumptions of Caragiannis et al. (2013); there need not even be a single base ranking σ^* . For example, when G such that we wish to select a set of W winners, F corresponding to a mixture of Mallows models – with all possible permutations of the W candidates in the top W positions in the base rankings – would still be design-invariant. Constructing a general class of ranking noise models that satisfies this property is an avenue for future work.

Assuming asymptotic design-invariance on voter preferences F may seem restrictive. However, absent axioms – that are precise enough for design purposes – to prefer one scoring rule β over another, the assumption allows us to proceed in a principled manner. We believe it is unlikely that such precise, satisfactory axioms exist generally. In the Appendix, we provide a simple example (similar to that of Starling (1986)) where 1-Approval and 2-Approval select disjoint sets of 2-Winners, and such examples can be adapted more generally to selecting W winners from either K -Approvals or K' -Approvals. In participatory budgeting with the goal of identifying 6-10 winning projects out of over twenty projects, it is unclear whether there is a principled reason to prefer 4-Approval over 8-Approval. However, such axioms would be an interesting avenue for future work.

Furthermore, in Section 6.6.2 we show that design invariance is often approximately satisfied in practice, especially for identifying a small set of winners, using data from a wide range of participatory budgeting and other elections.

6.4 Learning Rates and Optimal Design

Different elicitation and aggregation mechanisms may take different amounts of voters to learn the asymptotic outcome. For example, suppose we want to identify the worst candidate out of 100, where the voter’s rankings are drawn from a Mallows model with $\phi > 0$. Then, asking each voter to identify their single favorite candidate will eventually identify the worst candidate, but after many more voters than if we ask each voter to identify their least favorite candidate. We make such learning rates precise in this section. Our results in this section extend those of Caragiannis and Micha (2017) as discussed above, both to arbitrary positional scoring rules and by providing tighter bounds for how a scoring rule affects the convergence rate. These rates are precise enough to *design* scoring rules, for example comparing 4-Approval and 8-Approval in the above example.

6.4.1 Learning rates

We begin by deriving rates for how quickly a given positional scoring rule β learns its asymptotic outcome O^* (given it exists), as a function of the voter preference model F . In particular, we use *large deviation rates* at which a scoring rule learns (Dembo and Zeitouni, 2010).

Definition 6.4.1. Consider a sequence $\{A_N \geq 0\}_{N \in \mathbb{N}}$, where $A_N \rightarrow 0$. Value $r > 0$ is the large deviation rate for A_N if

$$r = - \lim_{N \rightarrow \infty} \frac{1}{N} \log A_N$$

When $r > 0$ exists, $A_N \rightarrow 0$ exponentially fast, with exponent r asymptotically, i.e., A_N is $e^{-rN \pm o(N)}$. These rates provide us both upper and lower bounds for the probability of an error or the number of errors in an outcome after N voters, up to polynomial factors. In particular, in the propositions below, we will calculate the large deviation rate of errors in the outcome. We will also then provide (loose) upper bounds for such errors after N voters that hold without any missing polynomial factors, for any N . These upper bounds are equivalent to Chernoff bounds.

The particular forms for these rates, derived below for general noise models F , may seem complex. However, they are useful both for theoreticians and practitioners. For example, in Section 6.5.1, we use the structure of such rates to resolve open questions regarding when randomization between mechanisms can help learn the outcome from votes drawn from an *arbitrary* noise model. In Section 6.6, we show that learning rates – even when empirically calculated – reflect the true behavior

of errors in real elections with a small number of voters; we then use empirically calculated learning rates to draw design insights across elections.

Rates for separating two candidates We now derive the large deviation learning rates for recovering the true ordering between a pair of candidates i, j , given noise model F . These rates will directly translate to the learning rate for the overall election, given some goal G .

Fix scoring rule $\beta \in \mathcal{B}$, voter distribution F , and consider candidates i, j such that $s_i > s_j$. Then, the probability of making a mistake in ranking these two candidates after N voters, $\Pr(\sigma^N(i) > \sigma^N(j))$, goes to zero with large deviation rate

$$r_{ij}(\beta) = - \inf_{z \in \mathbb{R}} \log \mathbb{E}_F [\exp(z(\beta(\sigma_v(i)) - \beta(\sigma_v(j))))]$$

Further, the following upper bound holds for any N .

$$\Pr(\sigma^N(i) > \sigma^N(j)) \leq \exp(-r_{ij}(\beta)N)$$

The proof follows directly from writing a random variable for the event of making a mistake after N voters and then applying known large deviation rates. This simplicity emerges because positional scoring rules are additive across voters.

The proposition establishes that – for a fixed number of candidates M and voter noise model F – the probability of making a mistake on any single pair of candidates i, j decreases exponentially with the number of voters, at a rate governed by the scoring rule β and the candidates' relative probabilities of appearing at each position of a voter's preference ranking. The rate $r_{ij}(\beta)$ is non-negative, and larger values correspond to faster learning of the relative ranking of i, j . Note that for notational convenience, we suppress F in the argument for the rate.

For general β , we cannot find a closed form for $r_{ij}(\beta)$. However, the structure of this rate, in particular that of the argument in the $\log(\cdot)$, will directly let us show that randomization cannot help learning outcomes among positional scoring rules, for *arbitrary* noise models F .

For K -Approval voting, further, the rate simplifies.

Consider β consistent with K -Approval voting for some fixed K , and candidates i, j such that $s_i > s_j$. Then the large deviation rate $r_{ij}(\beta)$ in Proposition 6.4.1 is

$$r_{ij}(K) = - \log \left(2\sqrt{t_{ij}^i(K)t_{ij}^j(K)} + 1 - t_{ij}^i(K) - t_{ij}^j(K) \right)$$

Where $t_{ij}^i(K) \triangleq \Pr_F(\sigma_v(i) \leq K, \sigma_v(j) > K)$, i.e., the probability that a voter approves i but not j .

The proof follows directly from the structure of β for K -Approval, $\beta(k) = \mathbb{I}[k \leq K]$; for each pair of candidates, the sufficient statistics are how often each candidate appears in a voter's top K

list but the other candidate does not.

We overload notation and use K directly in the argument for $r_{ij}(K)$. This rate function $r_{ij}(K)$ is convex in the probabilities $t_{ij}^i(K), t_{ij}^j(K)$; this fact will let us show that randomization, even among K -Approval mechanisms, cannot help learning the relationship of any pair of candidates.

Rates for learning the outcome In general, the rates at which one learns each pair of candidates immediately translate to rates for learning the entire outcome O^* .

Consider goal G and $\beta \in \mathcal{B}$ such that $O^N \rightarrow O^*$. Let Q^N be the expected number of errors in the outcome after N voters, $\sum_{i \in C_s^*, j \in C_t^*, s < t} \Pr(\sigma^N(i) > \sigma^N(j))$. Then Q^N goes to zero with large deviation rate

$$r(\beta) = \min_{i \in C_s^*, j \in C_t^*, s < t} r_{ij}(\beta)$$

Further, the following upper bound holds for any N .

$$Q^N \leq M^2 \exp(-rN)$$

The large deviation rate $r(\beta)$ thus provides a tight characterization for how many voters it takes to (with high confidence) recover the asymptotic outcome of an election. Note that the goal plays an important role: for selecting W winners, for example, it is not necessary to learn the exact relationship among candidates $\{1, \dots, W\}$, speeding up outcome learning. Design β also matters; e.g., even amongst approval voting mechanisms, $K = 1$ vs $K = 5$ will produce substantially different $t_{ij}^i(K)$. To derive learning rates for K -Approval for any given noise model or using real-world data, one simply needs to calculate these values. We do so numerically for the Mallows model and empirically with real world data in Sections 6.5.2 and 6.6, respectively.

6.4.2 Optimal design and discussion

Now that we can quantify how quickly a given scoring rule β learns its asymptotic outcome, we apply our framework to *designing* elections, i.e., choosing an optimal scoring rule β . For the rest of this work, we assume that the setting (M, F) is asymptotically design-invariant for the goal G , i.e., there exists an outcome that is asymptotically induced by every reasonable scoring rule. Then, the design of an election β only affects the *rate* at which the election converges to the asymptotic outcome O^* , as calculated above. With no other constraints, then, the design challenge is simple: find the rate optimal β .

Definition 6.4.2. *A scoring rule $\beta^* \in \mathcal{B}$ is rate optimal if it maximizes the rate in Proposition 6.4.1. K^* -Approval is Approval rate optimal if it maximizes the rate among K -Approval mechanisms.*

Rate optimal designs β learn the outcome faster than others in the number of voters, and so are preferable to other designs. What influences how quickly a design β learns? $\mathbb{E}_F[\exp(z(\beta(\sigma_v(i)) - \beta(\sigma_v(j))))]$

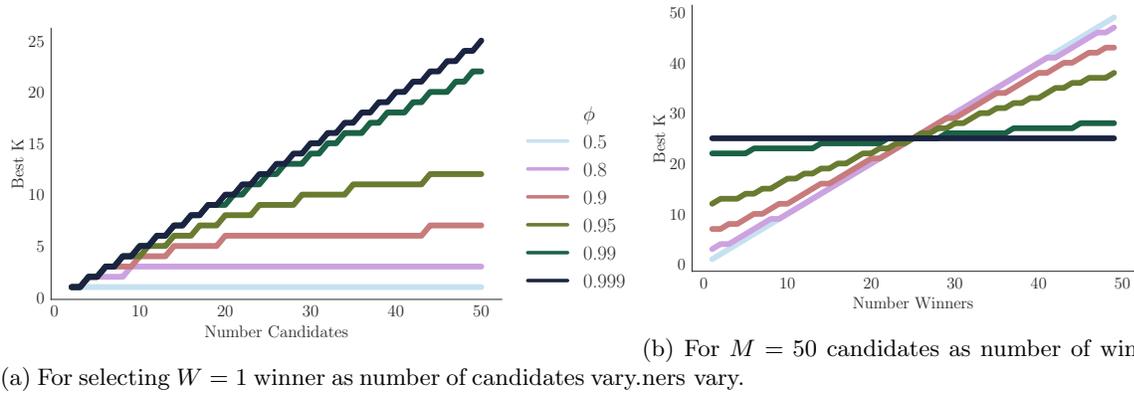


Figure 6.1: K -Approval rate optimal mechanism for the Mallows model as ϕ , number of candidates, and number of winners vary.

must be small (near zero) for negative z , and so $\beta(k) - \beta(k')$ must be large when $\Pr(\sigma_v(i) = k, \sigma_v(j) = k')$ is large. In other words, a scoring rule must reward a candidate achieving a position in a voter's ranking that is only achieved by asymptotically high-ranking candidates. For example, if it is common for worse candidates to be ranked second in a given voter's ranking but not to be ranked first, then $\beta(1) \gg \beta(2)$ would be beneficial.

Note that finding such designs requires knowledge of the voter noise model F , which in many settings may not be available before the election. However, next in Sections 6.5 and 6.6, we show that there are valuable insights that apply *across* elections, including how our approach has informed participatory budgeting deployments.

6.5 Theoretical Design Insights

The learning rates derived in the previous section provide election design insights, even before our approach is applied to real-world data. In particular, in this section, we first extend the previous literature on the (potential) benefits of randomizing between mechanisms. Then, we study the task of selecting W winners using K -Approval voting.

6.5.1 When does randomization help?

We now consider the question of whether *randomizing* between mechanisms in an election may speed up learning. By randomization, we mean: consider a set of scoring rules $B = \{\beta_1, \dots, \beta_P\} \subseteq \mathcal{B}$; elicitation and aggregation for a given voter is done according to a scoring rule picked at random from B , where β_p is selected with probability d_p .

Note that the learning rate of such randomized schemes can be calculated as before, by summing across β_p inside the $\mathbb{E}[\cdot]$ of $r_{ij}(\beta)$ or – for B consisting only of K -Approval votes – directly through

the resulting probability that the voter approves i but not j . We use $r_{ij}(B, D), r(B, D)$ to denote the candidate pairwise and overall outcome learning rates, respectively, for randomized mechanism (B, D) , where $B = \{\beta_1, \dots, \beta_P\} \subseteq \mathcal{B}$ and $D = \{d_1, \dots, d_P\}$.

It is known that in some settings randomization improves learning, asymptotically in the number of candidates. Caragiannis and Micha (2017) provide an example in which randomizing uniformly between all possible K -Approval mechanisms outperforms any static K -Approval elicitation, when the goal is to rank all the candidates. Their insight is that, under the Mallows model and under a fixed K , either the first two candidates will be hard to distinguish from each other, or the last two will, and randomizing between mechanisms balances learning each pair.

We now study randomization for the goal of selecting W winners and for arbitrary positional scoring rules and voter noise models. Our first result is that randomizing between scoring rules does not help, for *any* voter noise model, in contrast to the case when restricted to approval votes.

Theorem 6.5.1. *Randomization does not improve the outcome learning rate for any asymptotically design-invariant noise model F or goal G . For any randomized scoring rule mechanism (B, D) , where $B \subset \mathcal{B}$, for any F, G , the scoring rule $\beta^*(k) = \sum_p d_p \beta_p(k)$ satisfies $r(\beta^*) \geq r(B, D)$.*

The result follows from the fact that $\mathbb{E}_F[\exp(z(\beta(\sigma_v(i)) - \beta(\sigma_v(j))))]$ is convex in $\beta(k)$, for all i, j, z, F . Then, given a randomization over β_1, \dots, β_P , we can increase $-\inf_z \log(\cdot)$ by decreasing its argument, by instead using the static scoring rule defined by the corresponding convex combination of β_1, \dots, β_P . Note that such a negative result cannot be obtained via analysis that is asymptotic in the number of candidates; we need learning rates for a given election.

Next, we further refine the result of Caragiannis and Micha (2017), by showing that the “pivotal pair” feature of their example – where different pairs of candidates dominate the learning rate for different mechanisms – is key. In particular, our next result establishes, again for any noise model, that randomization amongst K -Approval mechanisms cannot help separate any *given* pair of candidates.

Theorem 6.5.2. *Randomization amongst K -Approval mechanisms does not improve the learning rate for separating a given pair of candidates i, j for any asymptotically design-invariant noise model F or goal G . For any randomized K -Approval mechanism (B, D) , where $\beta_p \in B$ corresponds to p -Approval, for any F, G , there exists a mechanism K_{ij}^* -Approval such that $r_{ij}(K_{ij}^*) \geq r_{ij}(B, D)$.*

The proof relies on the pairwise rate function $r_{ij}(K)$ being convex in the approval probabilities $t_{ij}^i(K), t_{ij}^j(K)$.

This theorem directly implies that, for the Mallows model, randomization among K -Approval voting cannot speed up learning when the goal is to identify a set of W winners, as opposed to when the goal is to rank.

Corollary 6.5.1. *Randomization among K -Approval mechanisms does not improve the learning rate for selecting W winners from the Mallows model. For any randomized K -Approval mechanism*

(B, D) , where $\beta_p \in B$ corresponds to p -Approval, for selecting W winners from the Mallows model, there exists an Approval rate optimal mechanism K^* -Approval such that $r(K^*) \geq r(B, D)$.

The proof simply notes that under the Mallows model with this goal, the candidate pair $W, W+1$ (when candidates are indexed according to reference distribution σ^*) is pivotal regardless of the K -Approval mechanism used. This corollary does not extend to arbitrary noise models, where randomization amongst K -approval mechanisms may improve the learning rate.

Theorem 6.5.3. *Randomization among K -Approval mechanisms may improve the learning rate for the goal of selecting W winners. There exist asymptotically design-invariant settings (M, F) for the goal of selecting W winners such that a randomized K -Approval mechanism (B, D) , where $\beta_p \in B$ corresponds to p -Approval, satisfies*

$$r(B, D) > \max_K r(K)$$

We prove the result two ways: (1) we construct an example in which candidate h is asymptotically selected, and candidates i, j are not. Which of $h \succ i$ or $h \succ j$ is the pivotal pair (determines the overall rate function) depends on the K -Approval mechanism used, and randomizing between two mechanisms improves the overall rate; (2) perhaps more interestingly, we find many examples in our real PB elections and other ranking data in which randomization would have sped up learning for the task of selecting a set of winning candidates (see Section 6.6.4).

6.5.2 K -Approval for selecting W winners

One of the most common voting settings is identifying a set of W winners using K -Approval, whether in representative democracy elections (typically $K = W = 1$), polling for such elections (where the goal often is to identify the top few candidates out of many, especially in primary races), or crowd-sourcing labels (where one wants one or a few labels for an item out of many possible ones). Here, we study how to design such elections, i.e., how to choose the best K , i.e., the one that maximizes the learning rate. For simplicity, we work with the Mallows model, extending the resulting insights to real-world data in the next section.

Recall that in a Mallows model, each voter's ranking is a noisy sample from a reference distribution σ^* . With this symmetric model, one may believe that setting $K = W$ is always optimal. For example, when noise parameter $\phi = 0$ and so each voter's ranking is exactly σ^* , $K = W$ is optimal; in fact, any other design $K \neq W$ fails to correctly identify the set of winners even asymptotically: it would not distinguish among the first K candidates in σ^* or among the last $M - K$ candidates. However, our next result establishes that the cases with $\phi > 0$ are different.

Theorem 6.5.4. *Under the Mallows model and the goal of selecting W winners, W -Approval may not be Approval rate optimal.*

We prove the theorem by example. To find this example and to generate the plots discussed next, we use an efficient dynamic program to exactly calculate the joint distributions of the locations $\sigma_v(i), \sigma_v(j)$ of pairs of candidates i, j in a voter’s ranking, given the Mallows noise parameter; we can then directly calculate $t_{ij}^i(K), t_{ij}^j(K)$ and thus the learning rate for each K -Approval mechanism. This program leverages Mallows repeated insertion probabilities (Diaconis, 1988; Lu and Boutilier, 2011) and may be of independent interest for numerical analyses of the Mallows model.

Numerical analysis We now numerically analyze, for the Mallows model, how the Approval rate optimal K -Approval mechanism varies with the Mallows noise parameter ϕ , the number of candidates M , and the number of winners W . Recall that the Mallows model is asymptotically design invariant, so different mechanisms only differ in *how quickly* they learn the asymptotic outcome.

In Figure 6.1a, the goal is to select $W = 1$ winner, and ϕ and M are varied. With low noise, $\phi \lesssim .5$, it is rate optimal to use 1-Approval, i.e., to ask each voter to select their favorite candidate, regardless of how many candidates there are. However, with higher noise ϕ , as the number of candidates in the election increases, so does the K in the optimal K -Approval mechanism. For $\phi = .999, M = 50$, for example, it is best to ask each voter to select their favorite 25 candidates, even if the task is to identify the single best candidate according to the reference distribution σ^* .

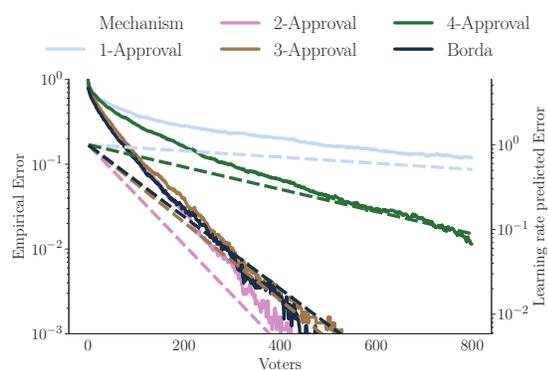
Similarly, Figure 6.1b shows how the rate optimal K -Approval mechanism changes with the number of winners desired and the noise parameter, fixing the number of candidates at $M = 50$. Again, with high noise it is best to ask voters to identify their favorite half of candidates, regardless of how many winners need to be identified. With low noise, however, W -Approval is optimal to select W winners.

Overall, the analysis suggests that with higher noise in the voter model, one should tend toward asking voters to rank their favorite half of candidates, regardless of M and W .

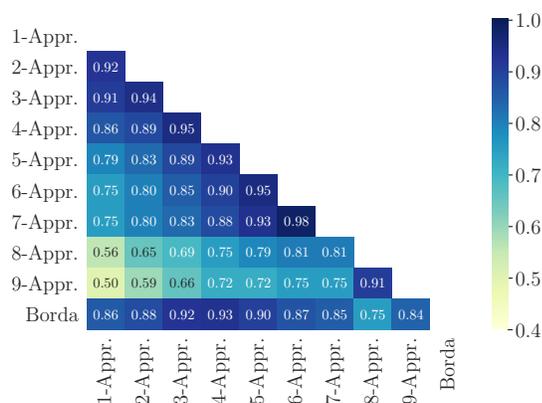
The high-noise setting may seem unrealistic; however, as we will see in the next section, which K -Approval mechanism is rate optimal in practice often scales like the high noise settings, consistent with the idea that voting distributions in practice do not look like they are drawn from a low-noise Mallows model.

6.6 Empirics and PB deployments

We now apply our insights to practice. We focus on K -Approval voting, as opposed to general scoring rules. This section is organized as follows. In Section 6.6.1, we describe our data sources. We validate our model in Section 6.6.2; first, we demonstrate that large deviation rates effectively capture how quickly various mechanisms learn; next, we show that in practice voter noise models are approximately design invariant. In Section 6.6.3, we show that the insights from Section 6.5.2 regarding optimal approval mechanisms extend to practice. Finally in Section 6.6.4 we note that



(a) Boston 2016 PB election, selecting 1 winner: Average empirical bootstrapped error – i.e., fraction of times the asymptotic winner is selected (solid lines, left axis), compared to such errors over time implied by the (empirically calculated) learning rates – i.e., e^{-rN} (dashed lines, right axis). The right axis is a vertically shifted (in log scale) version of the left axis, reflecting that the learning rate errors are asymptotically valid up to polynomial factors. All mechanisms return the same winner when all votes are counted. “Borda” is the Borda count for the 4 candidates ranked, and all others are assumed to be tied at rank 5 for each voter.



(b) Approximate design invariance across elections. For the task of selecting $W = 4$ winners, this plot shows the average overlap in the top 4 candidates identified by different mechanisms across all the elections in our dataset, if all voters with complete rankings are counted. For example, of the top 4 candidates identified by 1-Approval across elections, 92% are also identified as top 4 candidates by 2-Approval. For each K -Approval mechanism, we include all elections where there were at least $K + 1$ candidates.

Figure 6.2: Validating model: comparing learning rates to empirical error, and showing approximate design invariance.

we find many examples in practice where randomizing between K -Approval mechanisms improves learning.

6.6.1 Data description

We leverage two data sources (detailed dataset information is in Appendix Table E.1). First, we have partnered with dozens of local governments to help run participatory budgeting (PB) elections in the last five years. These elections have used a variety of methods, primarily K -Approval; our data in this work comes from 5 elections where K -Ranking was used, including 3 recent elections where $K = 10$. This data is particularly useful as PB is among the most common types of elections with many candidates and several winners, with several theoretical analyses (Freeman et al., 2019; Garg et al., 2019b; Goel et al., 2016).

Second, we use data available on PrefLib (Mattei and Walsh, 2013; O’Neill, 2013; Popov et al., 2014; Regenwetter et al., 2007, 2008), limiting ourselves to 28 elections with at least 5 candidates and 700 voters who provided full rankings. This ranking data spans many domains, from people’s sushi preferences to Glasgow City Council elections. This domain breadth supports the broad applicability of the design insights explored in this section.

We focus on ranking data to be able to simulate counter-factuals for the same election: with K -Ranking data, we can simulate what would have occurred with any K' -Approval elicitation mechanism, for $K' \leq K$ (assuming no behavioral quirks). With approval data, on the other hand, one cannot compare the mechanism to any other for that given election.

One challenge is that ranking many candidates is onerous, and so voters rank at most 14 candidates in our dataset. For the data we use from on PrefLib, full rankings (rankings up to the number of candidates) are available. In the PB elections in our partner cities, typically each voter ranks or selects her favorite $K \ll M$ candidates.

6.6.2 Model validation

Our model and design approach has two components that must be validated: (1) that learning rates can effectively be used to compare different mechanisms, and (2) that design invariance (approximately) holds in practice.

Large deviation rates as effective proxies for learning We now confirm that, for a given election, empirically calculated large deviation learning rates are effective proxies for the rate at which the error in recovering the asymptotic output decreases as the number of voters increases (even though large deviation learning rates are only asymptotically valid in the number of voters). As examples, we first identify three elections and goals for which many of the potential K -Approval mechanisms return exactly the same asymptotic outcome. Then, we bootstrap voters from the available data of voters and empirically calculate the errors made in identifying the winning set of

candidates. We further calculate the large deviation learning rates for these mechanisms, using F implied by the voting data and the formula in Proposition 6.4.1.²

Figure 6.2a shows the resulting errors over time for one such election where 4-Rankings are available. We further plot e^{-rN} for each mechanism, i.e., the error over time implied by the learning rate (up to polynomial factors). This plot, along with Appendix Figure E.1, yields several insights:

1. The mechanism matters: when selecting 1 winner from the election in Figure 6.2a after 400 votes, there is 20% chance of not picking the ultimate winner if 1-Approval is used. With 2 or 3-Approval, this number is 0.1%. The winner appears often in a voter’s top two or three positions (but not necessarily first), while the ultimate second place candidate often falls outside the top three. Scoring rules that reward top three placements thus perform well.
2. The learning rates effectively capture the behavior of the empirical error: both comparatively across mechanisms, as well as the asymptotic rate (slope of the line in log scale). This property enables use of large deviation learning rates as proxies for learning even in elections with a small number of voters.
3. Ranking K candidates rather than selecting K candidates is more onerous for voters. However, it does not always provide more information in terms of learning rates, as in the examples in Appendix Figure E.1.

Design invariance in practice Design invariance does not strictly hold in any election in our dataset (as expected as the condition is strong). However, it *approximately* holds. Similar mechanisms produce the same asymptotic outcome for many tasks. Figure 6.2b shows, for example, the average overlap across elections in the top 4 candidates identified by each mechanism. (Appendix Figure E.2 shows the same plot for the top 1 and 3 candidates, as well as the average Kendall’s τ rank correlation between the full rankings identified by different mechanisms). Furthermore, we find many elections and goals where most mechanisms return the same asymptotic answer, as in the elections we leverage for the plots showing learning rates are effective proxies. This relative consistency, especially for similar mechanisms, enables us to compare different mechanisms by their learning rates.

6.6.3 K -Approval for selecting W winners

In Section 6.5.2, we showed for the Mallows model how the rate optimal K -Approval mechanism changes with the noise parameter ϕ , the number of candidates, and the number of winners. We now show this scaling in practice.

For every election in our dataset, we find the Approval rate optimal mechanism (among K we can simulate) for every goal of selecting W winners, for $1 \leq W \leq M$. We then run a regression

²Given an empirical \hat{F} , learning rates can be numerically calculated: the $\inf_z[\cdot]$ is a convex minimization problem.

across all the elections for which K is rate optimal, versus the number of winners desired and the number of candidates; see Table E.2 in the Appendix for the regression table. While there is some variation across elections, the number of candidates and winners proves a reasonable metric across elections for the rate approval K -Approval mechanism ($R^2 \approx .27$).

The regression confirms the idea that in practice, one should regularize toward asking voters to choose their favorite half the candidates. For picking a small subset of winners $W \approx 4$ out of more than 10 candidates, for example, one should ask voters to provide their favorite $K \approx 6$ candidates, with $K > W$. This suggestion directly counters common practice. In the PB elections that we have helped run, for example, 4 or 5-Approval is most typical, even though ultimately 6-10 projects may be funded (out of ≈ 15 -20).

Then, in Figure E.3 in the Appendix, we plot the line induced from the regression coefficients with the Mallows rate optimal lines, for $M \leq 10$ candidates. Comparing to the rate optimal mechanisms for the Mallows model with various ϕ (within the candidate range for which we have empirical data), we find that empirical data behaves most closely to a Mallows model with noise parameter $\phi \in [.8, .9]$. (We are not claiming that empirical data is drawn from a Mallows model; it most certainly is not, with factors such as polarizing projects important in practice). This coarse comparison provides an approximate expected scaling behavior for elections with many candidates.

6.6.4 Randomization in practice

We find 16 examples in which randomizing between two K -Approval mechanisms leads to faster learning than using either mechanism separately, including 8 examples where such randomization beats the Approval rate optimal mechanism. Table E.3 in the Appendix contains details.

6.7 Discussion

We show that in elections with many candidates, the elicitation mechanism and corresponding scoring rule used affect how quickly the final outcome is learned. The learning speed differential between mechanisms can be the difference between identifying the ultimate winner with only a 80% probability or a 99.9% probability after 400 voters, for example. We then provide design decisions that emerge when our framework is applied to data from real elections. When using K -Approval to select a small number of W winners, for example, it is often better to ask voters to identify their favorite $K > W$ candidates. The insights from this work should be applicable in a variety of such settings, from elections to crowdsourcing labeling tasks.

There are several important, open research avenues. Most importantly, in real elections maximizing the rate at which the final outcome is identified is not the only goal, and future work should seek to balance such multiple objectives.

For example, there may be axiomatic reasons to prefer one elicitation mechanism over another,

e.g., that the final outcome corresponds to the candidate(s) that the most voters indicate is their first choice. Another objective may be to minimize the cognitive load imposed on voters. Asking voters to provide a full ranking over the candidates and then using a rate-optimal scoring rule trivially provides faster learning than any other mechanism. However, asking voters to rank 20 candidates is prohibitive in many settings. Future empirical work, in line with that of Benade et al. (2018) and Gelauff et al. (2018), should study the cognitive load various mechanisms impose on voters, to better understand the trade-off between the objectives.

Appendix A

Driver Surge Pricing

A.1 Additional discussion and information

A.1.1 Platform objective

Our focus in this work is on designing incentive compatible payment functions for drivers. Here, we establish that this task is a sub-problem of the comprehensive platform pricing problem—one that can be studied separately given the components we considered exogenous in our model description. We work with the dynamic model, and suppose that the platform’s primary objective is profit (our argument also trivially holds for revenue, trips served, welfare, or other objectives). With our assumption of a single, earnings-maximizing driver, the platform’s overall challenge is as follows.

On the rider side, we suppose that the two world state periods, $i \in \{1, 2\}$, are induced by latent demand shocks. The platform’s design lever is the pricing policy $p = \{p_1, p_2\}$, where $p_i(\tau)$ indicates the rider price for trip length τ in world state i . Rider demand depends on the prices, inducing request rates and distributions λ_i^p, F_i^p through a standard demand model for each trip: a rider with latent demand for trip τ requests a ride if the price is no more than their valuation for the trip (without substituting for trips of different lengths).

On the driver side, as detailed in our model formulation, the driver chooses a strategy σ to maximize earnings rate $R(w, \sigma, \lambda_i^p, F_i^p)$, where the additional arguments emphasize that earnings depend on rider prices through induced demand. Further, the driver has an outside option earnings rate of R , and will participate in the system only if it is possible to achieve earnings rate $R(w, \sigma, \lambda_i^p, F_i^p) \geq R$ with some strategy σ .

The set of rides *served* by the platform are those that are both *requested* by riders (as induced by pricing p) and *accepted* by the driver (denoted by driver strategy σ). Let $\text{Rev}(p, \lambda_i^p, F_i^p, \sigma) = \liminf_{t \rightarrow \infty} \frac{\text{Rev}(p, \lambda_i^p, F_i^p, \sigma, t)}{t}$ denote the resulting *revenue* rate for the platform, i.e., the rate paid by riders.

Putting things together, the platform's profit maximization problem is as follows.

$$\begin{aligned}
& \underset{p,w}{\text{maximize}} && \text{Rev}(p, \lambda_i^p, F_i^p, \sigma^*) - R(w, \sigma^*, \lambda_i^p, F_i^p) \\
& \text{subject to} && R(w, \sigma^*, \lambda_i^p, F_i^p) \geq R \\
& && \sigma^* \in \arg \max_{\sigma} R(w, \sigma, \lambda_i^p, F_i^p)
\end{aligned} \tag{A.1}$$

Where the first constraint is for driver participation, and the second for incentive compatibility (where the arg max is not unique, assume that the driver chooses the policy σ with largest measure.). With this formulation, the platform must jointly optimize prices p and payments w , as both together determine the set of trips served and the profit for each such trip. Such a tightly connected optimization would preclude the approach taken in this work, where we focus on designing the payment functions for drivers, holding prices p fixed. However, the optimization can be rewritten to make our approach tractable.

Program (A.2) yields the same optimal value as Program (A.1). For each solution, the same rides are served at the same prices as in a matching solution of (A.1).

$$\begin{aligned}
& \underset{p,w}{\text{maximize}} && \text{Rev}(p, \lambda_i^p, F_i^p, \sigma^*) - R \\
& \text{subject to} && \sigma^* = \{(0, \infty), (0, \infty)\} \\
& && R(w, \sigma^*, \lambda_i^p, F_i^p) = R \\
& && \sigma^* \in \arg \max_{\sigma} R(w, \sigma, \lambda_i^p, F_i^p)
\end{aligned} \tag{A.2}$$

The reformulation in Proposition A.1.1 follows from a simple insight: in our model with no driver private information, a driver rejecting a request is equivalent to the rider not requesting the trip – and the platform can predict such rejections perfectly. Then, for any optimal solution of Program (A.1) in which a rider requests a trip τ but the driver rejects it, the platform can equivalently raise rider prices until no rider requests such a trip, $F_i^p(\tau) = 0$, and so the driver accepts all requested trips lengths. Further, the driver earnings constraint $R(w, \sigma^*, \lambda_i^p, F_i^p) \geq R$ is of course tight: driver payments can otherwise be proportionally scaled down.

With Program (A.2), the driver payment function w and induced driver strategies σ just appear in the constraints. Given each potential choice of rider pricing function p and induced demand λ_i^p, F_i^p (i.e., which trips to service at what prices), the platform must determine how to pay drivers such that they accept every request, i.e., the platform must choose payments w_i such that the participation and IC constraints are met. In this work, we focus on this challenge, holding rider prices p and thus demand $\lambda_i \triangleq \lambda_i^p, F_i \triangleq F_i^p$, fixed.

A.1.2 Driver earnings in each state

Recall that in Lemma 2.2.1, we decompose the driver reward into reward rates for each world state, $R_i(w_i, \sigma_i)$, denoting the earnings rate while the driver is either open in i or on a trip that started in i . In our theoretical pricing results in Section 2.4, we show how to construct incentive compatible pricing given choices of average earnings in each state, i.e., setting $R_i(w_i, \sigma_i) = R_i$ for some R_1, R_2 . These rates, subject to the participation constraint that overall earnings $R(w, \sigma) \geq R$, is a design choice for the platform. Here, we provide some intuition for how to make this choice.¹

Business constraint from revenues. The platform’s revenue rate can be decomposed just like the driver earnings rate, with state i revenue rate, $\text{Rev}_i(p_i, \sigma_i, \lambda_i^p, F_i^p) = \frac{\frac{1}{F_i(\sigma_i)} \int_{\tau \in \sigma_i} p_i(\tau) dF_i(\tau)}{T_i(\sigma_i)}$. Latent demand and the choice of prices p_i together induce platform revenue rates for each world state. Then, in practice, the per-state driver earnings rates R_i are approximately set as a fixed fraction of revenue

$$R_i = \alpha \text{Rev}_i(p_i, \sigma_i, \lambda_i^p, F_i^p)$$

for some α . This choice passes on the revenue earned in each state to drivers, and so represents a *partially* decoupled setting: at the trip level, the amount paid to drivers may deviate from that paid by the rider, but prices are coupled on average at the level of a surge state. In practice this simple rule helps ensure that individual prices for a rider and driver do not differ by too much, which may be desirable for transparency and driver satisfaction reasons.

Driver positioning. However, the question of at what level to best decouple prices, and e.g., how to potentially transfer money between different surge states, is an interesting one for future work. Here, we describe one potential rationale for optimizing R_i .

Empirically, Lu et al. (2018) find that drivers respond to real-time surge prices (displayed through a heat-map) by re-positioning themselves to surge areas, an effect that is in addition to drivers choosing to drive (activating) in times and places where they expect to see surge. Thus, a higher surge earnings rate R_2 translate to more drivers during surge, as a result of both (a) short term, real-time movement toward surge due to seeing the heat-maps as in Figure 2.1, and (b) drivers logging on when and where there tends to be surge. A platform could thus choose the relative values of R_i as a lever for this type of re-positioning. See, e.g., Besbes et al. (2018b) for theoretical insight on this challenge.

Our model does not directly capture either of the above ways a platform could set and optimize R_i , as it has a single driver and geographic location, and we do not optimize rider prices and thus revenue. However, note that both effects above are mediated through the average earnings (i.e., R_i), either predicted by the driver or communicated through a heat-map, and do not depend directly on trip specific earnings, i.e., w_i . Thus, these effects can be incorporated by adding the constraints

¹The rider-side pricing problem of setting average prices and thus revenue Rev_i , given the latent demand, is potentially easier as the primary goal is a short-term allocation of the supply (drivers) to the riders who most value the service. The driver side problem, as discussed, is trickier as there are both short- and long-term effects.

$R_i(w_i, \sigma_i) = R_i$ in Program (A.2), with target earnings rate R_i optimized elsewhere.

We take this approach in this work, analyzing for what values of R_i the constraints $R_i(w_i, \sigma_i) = R_i$ are compatible with incentive compatible pricing. In our main result, Theorem 2.4.1, we cannot construct IC prices that induce all relative values of $R_1(w_1, \sigma_1) = R_1$ and $R_2(w_2, \sigma_2) = R_2$: if the platform tries to make the surge state $i = 2$ is too valuable compared to regular times $i = 1$, $R_2 \gg R_1$, then drivers will reject long trips in the non-surge state.

A.1.3 Model’s relationship to practice

Several of our theoretical model choices emerge from common ride-hailing practice; other choices – such as not considering spatial heterogeneity – differ from practice, and so we consider the generalizability of our insights to practice in Section 2.6, using real ride-hailing data. See also Section A.2.1 where we justify our choices in the numerical section with RideAustin data and provide more information on, e.g., surge evolution.

Heat-map constraint and affine pricing. When drivers are not on a trip, they see a heat-map of the current surge values, indicated as a multiplier or additive value, cf. Figure 2.1; this has important implications for practice, and for the pricing functions we consider in this work.

First, in our numerical and empirical sections we focus on multiplicative and additive surge, as opposed to other general surge payment schemes. Two rationales for this choice are that these are the schemes considered by platforms in practice, and that they naturally serve as approximations of our IC scheme. The fundamental rationale, however, is that such schemes can be directly displayed on the heat-map. With such single-parameter schemes, the driver can connect their surge payment to knowledge available to them before the trip starts. This is an important feature in practice, where platforms must be as transparent as possible regarding how they pay drivers. Consider for example, if the platform instead displayed on the heat-map some expected payment over all trips taken in that spatio-temporal spot (e.g., the equivalent of R_i); the driver would have no way to verify that the platform in fact did pay out that amount on average, unless they crowd-sourced data from other drivers.

Second, in this work we consider only pricing functions that depend on the world state when the trip starts, but do not incorporate information from what happens during the trip. Again, this is an important practical constraint: incorporating on-trip information would require the platform to perform a path-integral over surge values in the driver’s spatio-temporal path from the origin to the destination, which would be difficult to implement and for the driver to verify. More fundamentally, however, the surge payment is partially an incentive for drivers to re-locate to a surge area, cf. Lu et al. (2018), and modeled by R_i in our work. Updating surge payments based on what happens when a driver is on-trip would change such incentives.

Surge evolution. Surge is clearly non-Markovian and non-binary in practice, with strong intra-day patterns – for example, rush hours have predictably higher surge values: see Appendix

Figure A.3b.

However, evolution of surge on finer time scales, on the level of individual trips, is more volatile, and believably Markovian: see Appendix Figure A.3c, which shows the (spatially-averaged) surge factor in a small region around the Texas Capitol building every ten minutes over 3 days. Thus, from the perspective of a single driver who has decided to drive at a certain time block (for example, 5-8pm), surge is believably Markovian on the time order that they are making decisions for whether to accept certain trips.

The main theoretical difficulty with analyzing non-Markovian updates is that, then, the driver optimal policy is dependent on the time index as well as the state index; then, results will very strongly depend on the specific trip length distribution chosen, and in particular the interaction between the trip length distribution and the surge pattern structure. This interaction prevents any generalizable insights from emerging. However, as detailed above, our empirical analysis suggests that our results hold up even under more realistic surge.

Driver Activation We do not endeavor to explain *why* surge pricing might be useful in this paper: in our view, riders respond to rider prices, and drivers activate based on expected mean earnings (i.e., R_i and R , as discussed above), which we take as exogenous. These aspects are well studied in the ride-hailing literature. Rather, our paper studies the orthogonal question of how to pay a driver for trips *once they are online*, not how to induce drivers to drive when and where there is high demand.

Single driver and equilibrium effects. Our model considers a *single* driver, when in reality there are of course many drivers on the road. We do not believe that doing so affects the results, as the number of other drivers on the road affects average surge dynamics and activation, but presumably not individual trip decisions, except as mediated through future expectations of surge.

The main theoretical difficulty with analyzing multiple drivers is it would add historical state to the system not captured by just the current surge state, pertaining to the number of currently open drivers and the distribution of when currently busy drivers will next become open. This difficulty is similar to that of modeling non-Markovian surge evolution. It would also lead to a implausible driver behavior model – each earnings maximizing driver would have to keep track of the number of other open drivers (and the distribution of when currently busy drivers will next become open).

A.1.4 Supplementary Figures

Figure A.1 shows in an example $\mu_2(\sigma)$ as it changes with the surge driver policy $\sigma_2 = (t, \infty)$, for some t . Figure A.2 compares IC surge pricing to multiplicative and additive surge.

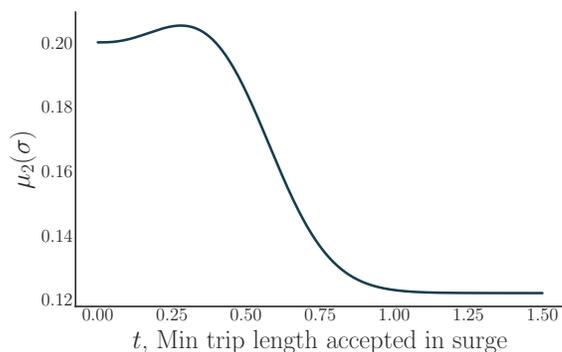


Figure A.1: Fraction of time spent in surge state, $\mu_2(\sigma)$, with driver policy $\sigma = \{\sigma_1 = (0, \infty), \sigma_2\}$, where $\sigma_2 = (t, \infty)$, i.e., t is the minimum trip length accepted in the surge state. The primitives are as follows: $\lambda_1 = \lambda_2 = 12$, $\lambda_{1 \rightarrow 2} = 1$, $\lambda_{2 \rightarrow 1} = 4$; in both states, trip lengths are distributed according to a Weibull distribution with shape 2 and mean $\frac{1}{3}$. These parameters reflect realistic average trip to wait time values, and that surge tends to be short-lived compared to non-surge times. Note that the driver can increase the time spent in the surge state by rejecting short surge trips.

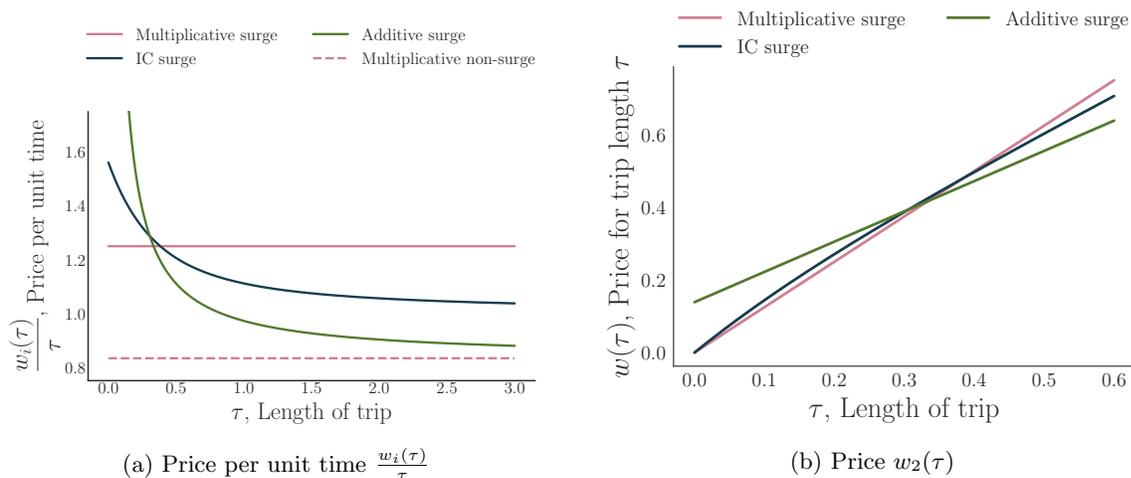


Figure A.2: Using the same model primitives as in Figure A.1: the payment function $w_i(\tau)$ for various surge mechanisms plotted two ways, when $R_2 = 1$ and $R_1 = \frac{2}{3}$ for drivers who accept every trip.

A.2 Extra empirical information

This section contains more empirical information. Section A.2.1 provides additional results related to the model validity and the variance of driver earnings with the various payment functions. Section A.2.2 contains more detail and robustness checks for the main empirical analysis.

A.2.1 Additional results and facts

Model validity

Here, we discuss how various components of the model relate to ride-hailing marketplaces in practice, using the RideAustin data from the rest of the empirics. We also justify the three claims we make in the numerics regarding the common parameter regimes for ride-hailing platforms.

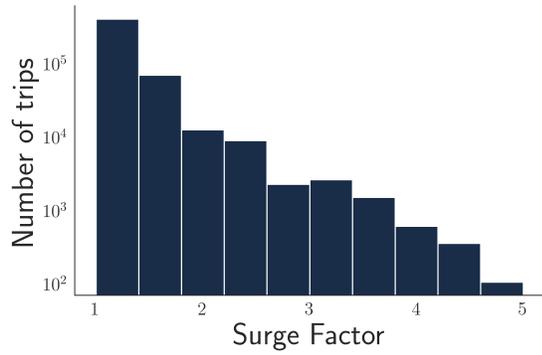
Surge is non-binary, and between 1.1 and 3 times more valuable than non-surge Figure A.3a contains a histogram of the surge factor. Surge in the RideAustin marketplace during the time period analyzed takes values divisible by 0.25, between 1 and 5. The mean surge factor is 1.19, only 30% of trips are surged, and more than 97% of surged trips have a surge factor in $(1, 3]$.

In the model, surge evolves according to a continuous time markov chain. Figure A.3b breaks down the average surge factor in each 30 minute period in a day, split up by weekdays and weekends. Surge is clearly not Markovian – there are clear, expected patterns in surge that correlate with rush hours and early morning times when there may be few drivers on the road. However, there is substantial additional volatility in addition to the non-Markovian daily patterns. Figure A.3c shows average surge in each 10 minute period, over 3 days for trips starting near the Texas Capitol building. The lengths, peak, and start/end times of each surge period differ – on a ten minute time scale, i.e., on the order of trip lengths, surge is not very predictable, and so a Markovian assumption may be reasonable on a small time scale.

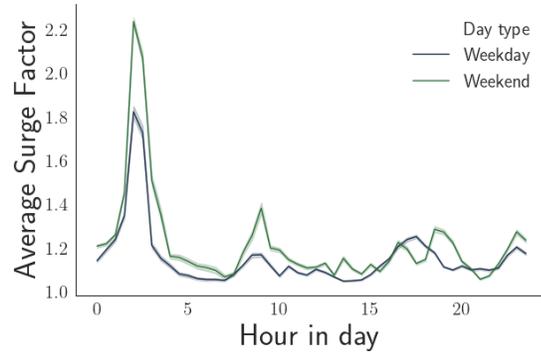
Surge is short-lived compared to non-surge periods ($\lambda_{2 \rightarrow 1} \gg \lambda_{1 \rightarrow 2}$) High-surge periods are indeed short-lasting compared to low surge periods, and peak surge tends to be short lasting. Figure A.3d shows the mean surge factor in the future, based on the current surge factor. With low surge, the average surge even an hour in the future remains close to 1. With high surge, however, the average surge in the future decays – and the higher the surge, the faster the decay.

In a typical surge a driver may only be able to complete one or two such trips. ($\frac{1}{\lambda_{2 \rightarrow 1}} \approx$ mean trip length). By jointly analyzing Figures A.3d and A.4c, we can see that drivers are indeed only be able to complete a few trips during surge before it dissipates. On trip times (with rider in the car) are on the order of 10-15 minutes, and the driver must also wait for a new request and then drive to the rider. Surge has typically decreased substantially after an hour.

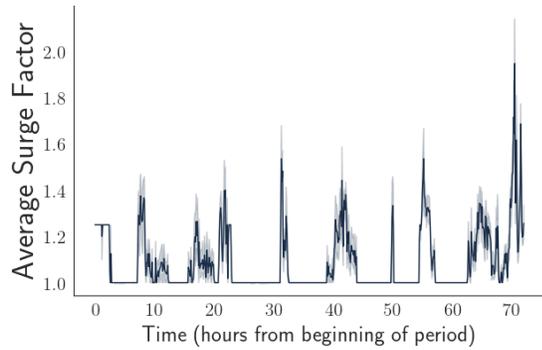
More directly, Figure A.3e shows for each driver session that has at least 5 trips, the average surge factor of each trip in the session, split by the surge factor of the first trip. Indeed, a driver is only able to complete a few trips with peak surge. We note, however, that this plot is susceptible to selection effects – a driver may choose to drive a different amount of time based on surge conditions.



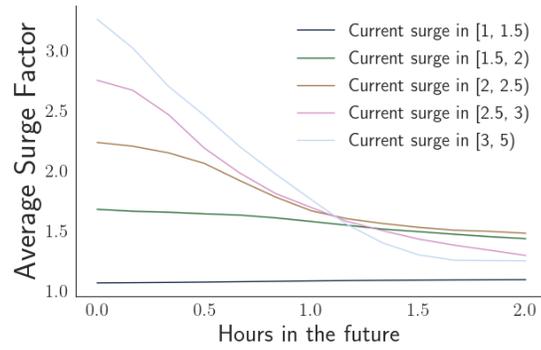
(a) Histogram of surge, in log scale.



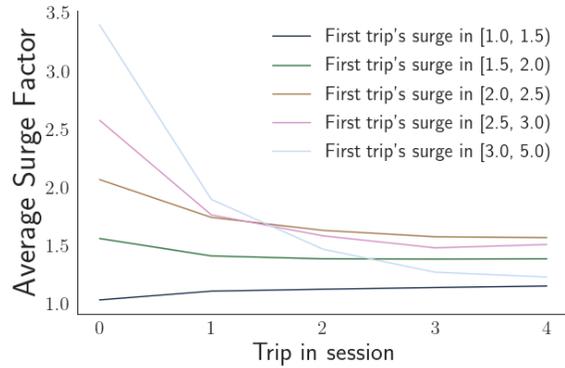
(b) Average surge factor in each 30 minute period of the day



(c) Average surge in each 10 minute period over three days on trips that start within 5 miles from Texas Capitol building.



(d) Divide the 2 months into periods of 10 minutes each. Then, this plot shows the mean surge factor x hours in the future, split by bucket of the current surge factor.



(e) For each driver session that has at least 5 trips, the average surge factor of each trip in the session, split by the surge factor of the first trip.

Figure A.3: Surge facts from RideAustin marketplace

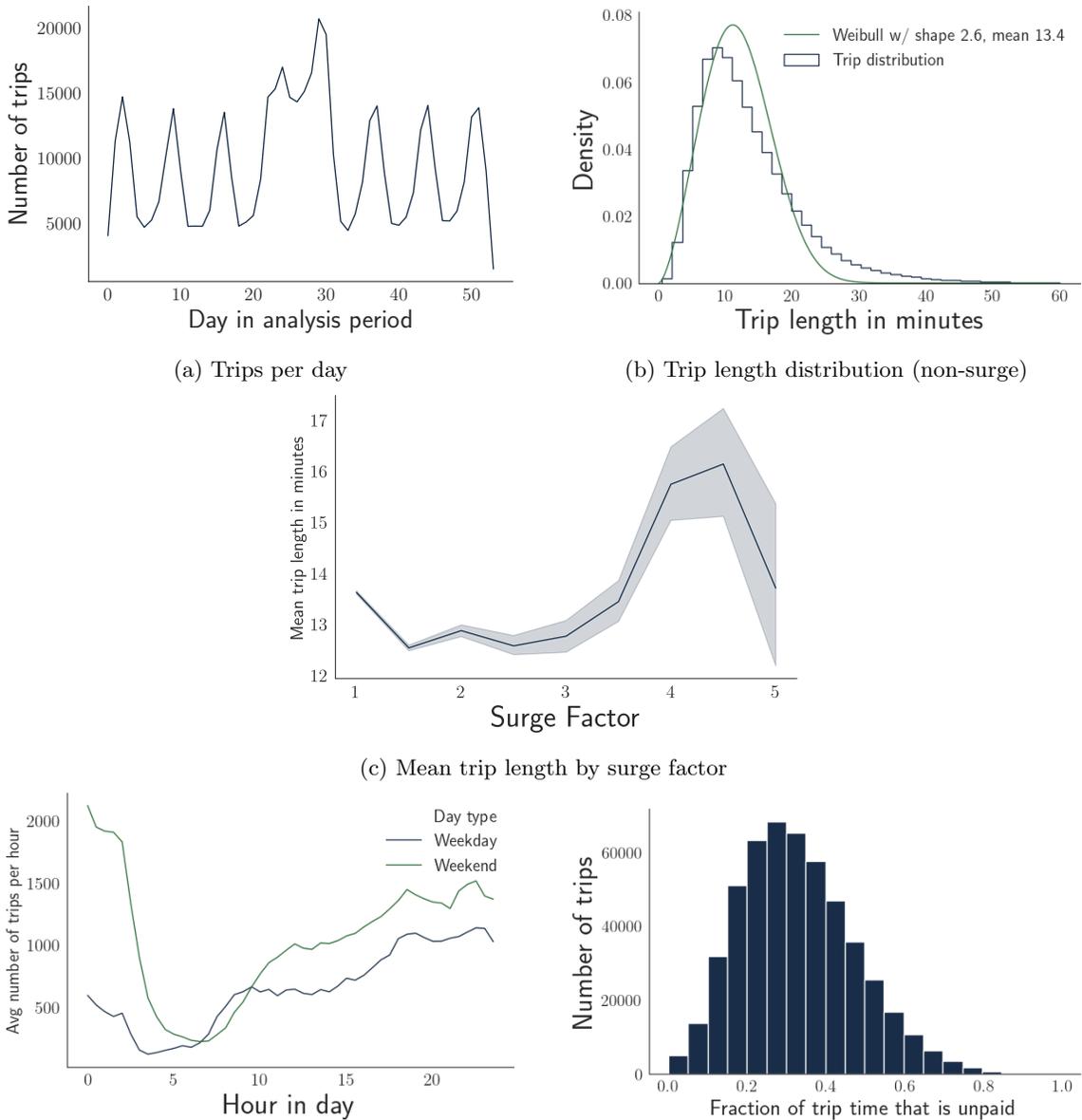


Figure A.4: Basic trip facts from RideAustin marketplace

In the model, on-trip time and time driving to the rider are combined. In practice, a job is typically split up into two components: the time it takes to drive to the rider, and the time that the ride is in the car – and only the second part is paid. Figure A.4e shows a histogram

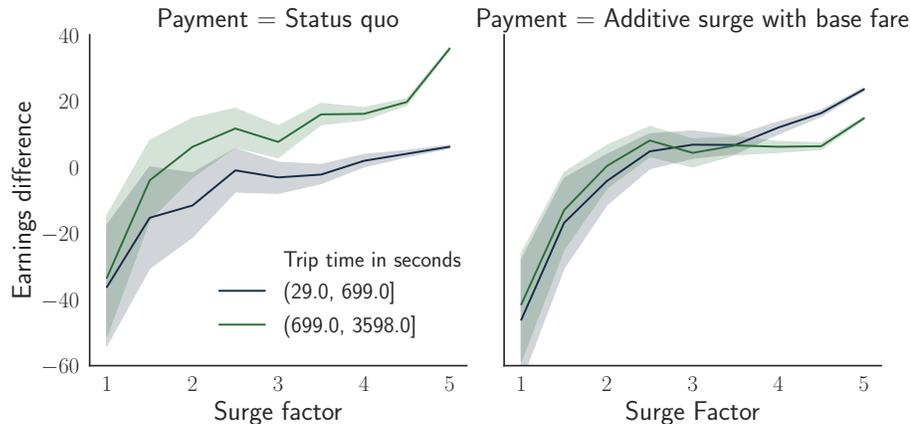


Figure A.5: Same as Figure 2.5, except with the surge factor flipped to simulate a world with frequent, valuable surge.

of the resulting fraction of the total job time that is unpaid. Note that this time is substantial in the RideAustin data, on average about 30%.

In the numerics, trip lengths are distributed as a Weibull distribution with shape 2. Figure A.4b shows the distribution of trip lengths for trips without surge. The shape approximation is reasonable, as a Weibull distribution with shape 2.6 best fits the data (with mean set to the empirical mean). Figure A.4c shows the mean length distribution by surge factor. Perhaps interestingly, this mean length is non-monotonic in the surge factor, first decreasing and then increasing with the surge factor.

We cannot directly test the claim in the numerics that in a typical surge the driver will be able to receive and reject several trip requests ($\frac{\lambda_2}{\lambda_{2 \rightarrow 1}} > 1$, but small) – we do not observe drivers being open to receive a request, unless they actually received a trip request. Unlike in the matching technique for trip indifference, we cannot use drivers who completed a trip as a proxy – the measurement would be sensitive to drivers logging off, and the end-locations of trips not being representative of all trips.

Furthermore, note that the insights regarding additive vs multiplicative surge extend to the empirics, despite the ways reality deviates from the model.

Regime with frequent, valuable surge

Recall that one of the theoretical insights from Theorem 2.4.1 is that our incentive compatible pricing scheme only works in a certain regime, if surge is not too valuable compared to regular periods on average, that $\frac{R_1}{R_2} \in [C, 1]$. This general insight extends to arbitrary pricing functions (i.e., as $\frac{R_1}{R_2} \rightarrow 0$, then no pricing function w_1 during regular periods will induce drivers to accept trips then).

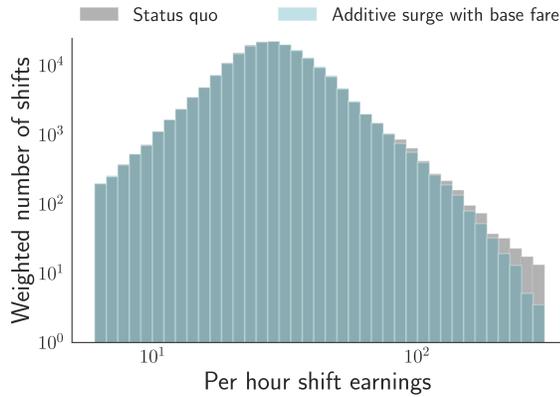


Figure A.6: Histogram of per-shift driver earnings per hour. Note that the y-axis is in log scale.

Here, we show that this insight also extends to practice, with non-binary surge. We simulate the following world: we “flip” the surge factor

$$\text{Simulated surge} = 6 - \text{Actual surge}.$$

With this flipped surge, 97% of surged trips have a surge factor in $[3, 5]$, and 30% of the trips have a surge factor of 5: surge is now the default, and extremely valuable compared to non-surge periods.

Then, we calculate the driver’s payment according to each such pricing function. Figure A.5 shows the resulting plots for earnings difference by trip length, using the status quo payment function (but with the simulated surge factor) and with an equivalent additive surge. Two insights emerge:

- With low surge (factor in $[1, 3]$), drivers are better off on average rejecting most trip requests, regardless of whether payments are additive or multiplicative.
- A more complex pricing function may be needed: multiplicative surge over-values long trips with high surge, and additive surge over-values short trips.

Driver earnings variance

We now calculate statistics regarding the average amount drivers earn during a single driving “shift,” ideally defined as the time between which drivers turn on their app and when they turn it off. To group trips together into a single driver shift, we use a data column called *active driver ID*, which is a refinement of *driver ID* and seems to correspond to a shift as defined internally by RideAustin.

The “length” of a shift is defined as the time between the first time the driver was dispatched for a trip during the shift, and the end time of the last completed trip during the shift. Note that this value is an underestimate of the true shift length, as it does not contain the time it took to receive

the first trip request or the time it takes for the driver to go home after their last trip. Thus, our estimated shift per hour earnings are biased upwards.

The driver's total earning during the shift is simply the sum of the payments from each trip, under the payment function being analyzed. Then, the earnings per hour in a single shift is the total earnings divided by the trip length.

Figure A.6 shows a weighted histogram of the per hour shift earnings, where the weights are the shift lengths in hours. Additive surge leads to a lower variance of per hour shift earnings (but the same mean, as constructed). The standard deviation of per-hour earnings are, respectively: \$16.97 (Status quo fare), and \$15.83 (Additive surge with base fare) with mean hourly earnings of about \$32.22. If we instead remove the minimum fare and pickup fare components and simulate pure additive or multiplicative surge, the standard deviations are: \$16.59 (Additive surge), \$18.35 (Multiplicative surge).

A.2.2 Empirical analysis additional information

Pre-processing

There are 509,823 rows (trips) in the time period analyzed.

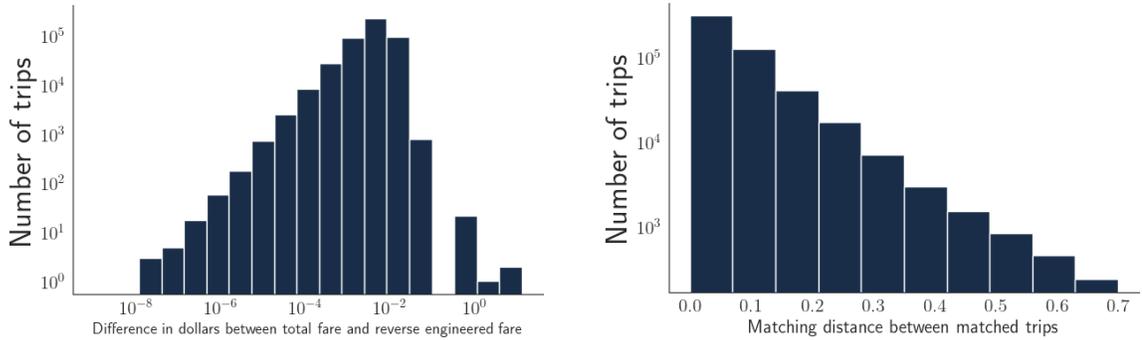
- 4626 trips were longer than 1 hour or shorter than 30 seconds and were discarded.
- 3780 were longer than 100 miles or shorter than 0.25 miles and were discarded (some overlap with those discarded for time).
- 26 trips had clearly erroneous *total fare* (null, or too high for mileage/distance by multiple orders of magnitude) and were not used to calibrate the reverse engineered fare.

We end up with 503,383 trips in our analysis.

Payment functions

Figure A.7a shows a histogram of the difference between the *total fare* available as a column, and the reverse engineered fare derived from the functional form in the main text. The fit is good, with a mean difference of \$0.005.

Figure A.8 plots the constructed Additive surge fare versus the status quo payments, at the trip level. As expected, additive surge pays more for short surged trips, and less for long surged trips.



(a) Histogram of difference between *total fare* and the reverse engineered fare.

(b) “Matching distance” between matched trips used for the counter-factual earnings.

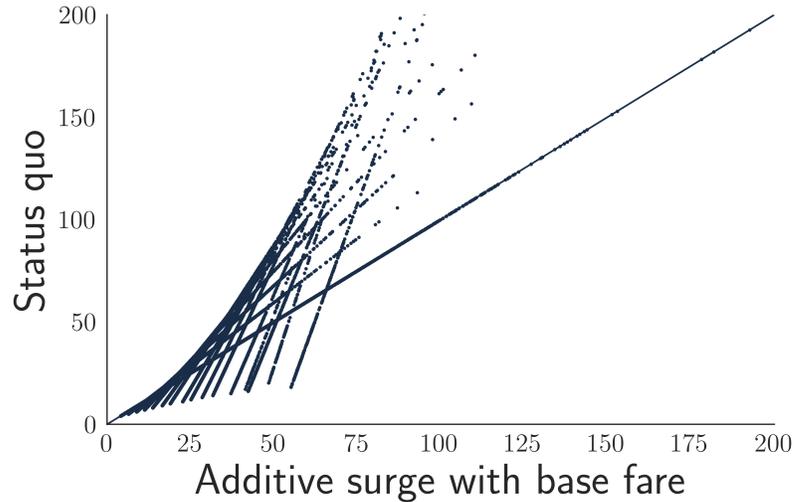


Figure A.8: Constructed payment function (Additive surge with base fare) vs the reverse engineered Status quo fare payments at the trip level. As expected, additive surge tends to pay higher for shorter trips and lower for longer trips.

Matching trips

The “matching distance” as described in the main text between pairs of $(date-time, location)$ tuples is:

$$\begin{aligned}
 \text{distance}((\text{time}_1, \text{location}_1), (\text{time}_2, \text{location}_2)) &= \text{difference in hours}(\text{time}_1, \text{time}_2) \\
 &\quad + \frac{1}{20} \text{difference in miles}(\text{location}_1, \text{location}_2)
 \end{aligned}$$

Figure A.7b shows the distribution of these distances between a given trip and the matched trip used for counter-factual earnings, for the matching technique described in the main text.

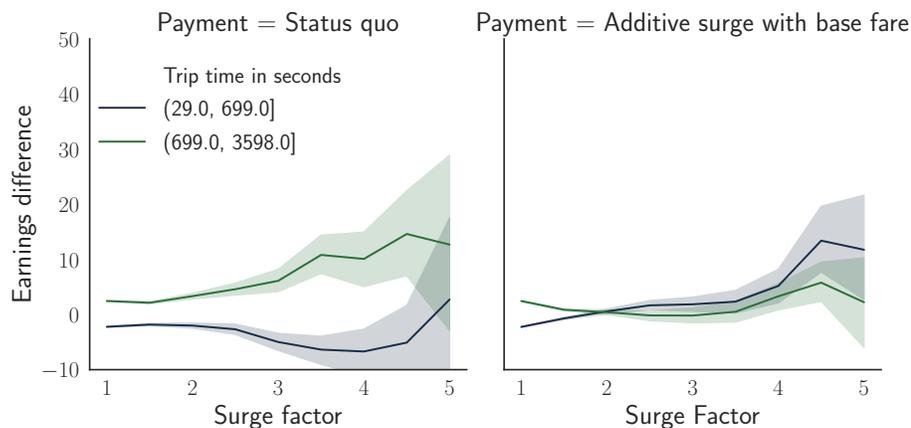


Figure A.9: Using next nearby driver with an accepted trip as the counter-factual match.

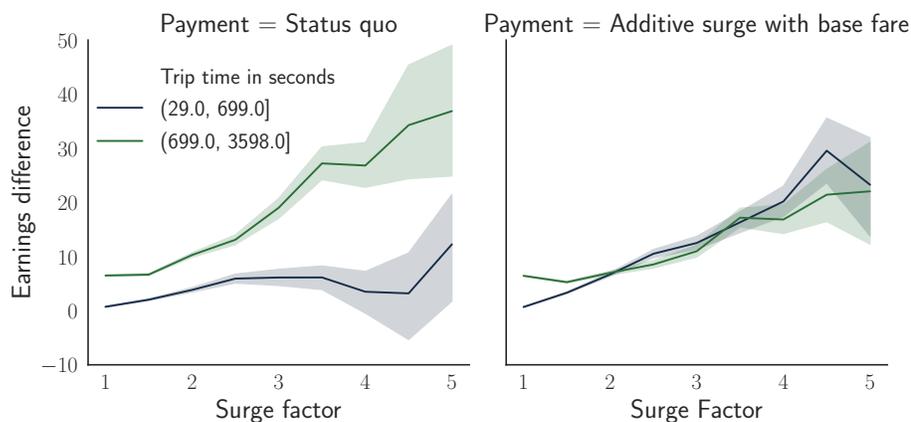


Figure A.10: Using period length of next 1 hour (instead of 1.5 hours).

For robustness, we also use an alternate way to find a match for a given trip: using the *next* driver who accepted a trip nearby. We calculate the matching distance between the given trip's *start* time and location, and each future trips' *start* time and location, and choose the driver of the closest match. As with the previous method, we filter out recent trips with drivers who are the same as the given trip's driver. Note that with this method, the expected earnings difference should be close to zero, as both drivers match at about the same time and place. However, the variances may vary with the payment function.

Trip indifference

We now carry out some robustness checks for the trip indifference results, and present supplementary results.

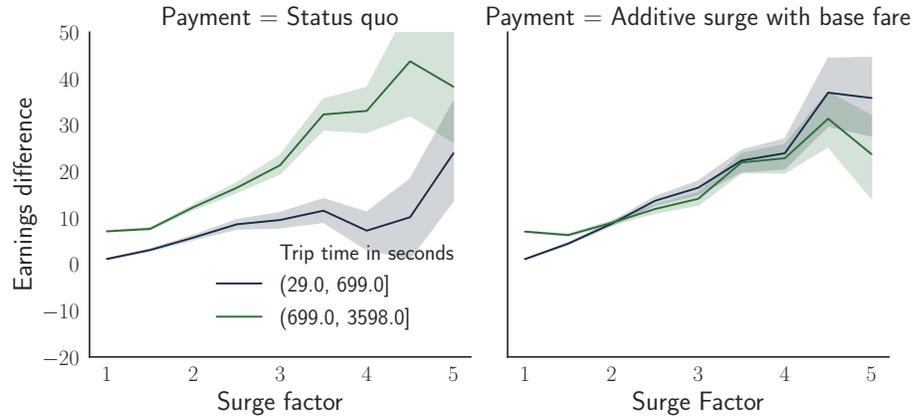


Figure A.11: Starting measurement from dispatch time instead of trip start time, i.e., taking into account the first part of the trip that is unpaid for the driver.

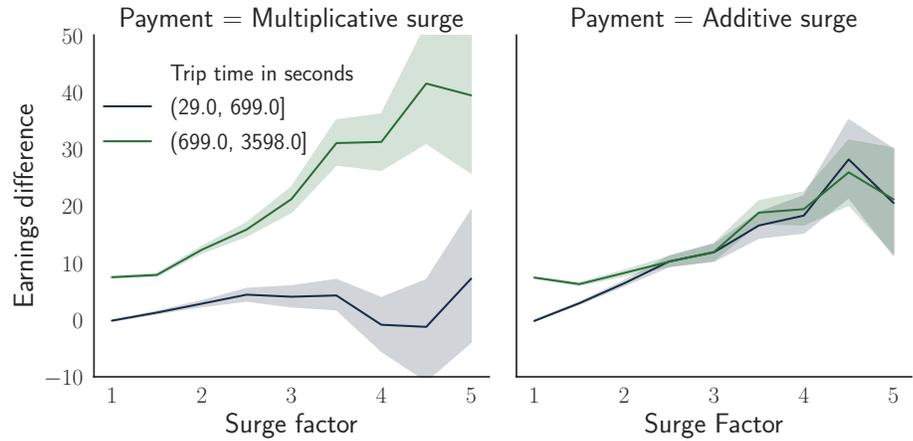


Figure A.12: With pure multiplicative and additive surge, respectively (no min fare).

Figure A.9 shows the same figure as in the main text, but instead using the next driver with an accepted trip matching function described in Section A.2.2. The means of the trip indifference (unconditional on trip length) are close to zero, as expected, but additive surge better balances the relative value of short and long trips, as before.

Figure A.10 shows the same figure as in the main text with the same matching function, but instead calculating the driver's earnings over the next 1 hour. Results are identical.

Figure A.11 starts counting the earnings of drivers starting at the given driver's *dispatch* time instead of trip *start* time; results are qualitatively identical, demonstrating that the fact that in practice there are two components to a trip – time from dispatch to the rider (unpaid typically), and time with the rider to the destination (paid) – do not substantively affect the results.

Finally, Figure A.12 shows the same figure but with how the driver would be paid under the *pure* multiplicative and additive surge functions studied in the rest of this work, defined as follows:

$$\textbf{Multiplicative surge : } [B \times M_{SurgeFactor}] \times SurgeFactor$$

$$\textbf{Additive surge : } [B \times M_{SurgeFactor}] + [(SurgeFactor - 1) \times A_{SurgeFactor}]$$

$M_{SurgeFactor}$ and $A_{SurgeFactor}$ are surge factor dependent constants that are set such that these alternative payment functions spend the same amount of money overall for each surge factor as does the status quo fare. As with the additive surge with a minimum fare, these alternative payments do not change the mean trip payment conditional on the surge factor, but do change how money is allocated to various trips within that surge. If instead we used a single constant across surge factors, this feature would not hold, and the payment functions may pay different amounts on average for the same surge factor.

A.3 Proofs of single state model results

In this section, we provide proofs of the theorems and lemmas in the main text regarding the single state model. Section A.3.1 formally states the driver reward. Section A.3.2 contains the proof of Theorem 2.3.1. Section A.3.3 contains the proof of Proposition 2.3.1. Finally, Section A.3.4 contains a partial uniqueness result regarding the optimal driver policy.

A.3.1 Driver reward

Recall that $R(w, \sigma, t)$ is the total earnings from jobs accepted up from time 0 to time t , i.e., $R(w, \sigma, t) = \mathbb{E} \left[\sum_{k=1}^{N(t)} w(\tau_i) \right]$, where τ_i is the length of the i th job the driver accepts, e_i is the wait time to that job, and $N(t) = |\{i : 0 \leq \tau_i + e_i \leq t\}|$ is the number of accepted jobs up to time t .

As mentioned using the renewal reward theorem in the main text,

$$R(w, \sigma) \triangleq \liminf_{t \rightarrow \infty} \frac{R(w, \sigma, t)}{t} = \frac{\text{Expected cycle payment given } \sigma}{\text{Expected cycle length given } \sigma} = \frac{\frac{1}{F(\sigma)} \int_{\tau \in \sigma} w(\tau) dF(\tau)}{\frac{1}{F(\sigma)\lambda} + \frac{1}{F(\sigma)} \int_{\tau \in \sigma} \tau dF(\tau)}$$

The $\frac{1}{\lambda F(\sigma)}$ term is the expected value of a exponential random variable with rate $\lambda F(\sigma)$, which is the rate at which a driver accepts ride requests when open.

A.3.2 Proof of Theorem 2.3.1

We now prove Theorem 2.3.1, regarding the form of the optimal policy in the single-state model – where the *length* of a trip does not matter, only the earnings rate. The optimal policy trades off the earnings rate while on a trip with the driver’s utilization rate. At a high level, the proof proceeds as

follows: starting from any policy that is not of the appropriate form, we replace trips in the policy with those with a higher earnings rate, while keeping the utilization rate exactly the same. Such replacements result in a policy that is *almost* of the correct form, except there may be an earnings rate c such that only a *subset* of $\{\tau : \frac{w(\tau)}{\tau} = c\}$ is in the policy. The remainder of the proof is showing that such a policy can be improved to form a policy of the appropriate form.

Theorem 2.3.1. *With a single state, for each w there exists a constant $c_w \in \mathbb{R}_+$ such that the policy $\sigma^* = \left\{ \tau : \frac{w(\tau)}{\tau} \geq c_w \right\}$ is optimal for the driver with respect to w .*

Proof. Proof. Let $\gamma(\tau) \triangleq \frac{w(\tau)}{\tau}$. Assume that $F(\{\tau : w(\tau) > 0\}) > 0$. Otherwise any policy is optimal and so the result is trivial.

Start at $\sigma \subsetneq (0, \infty)$. We first show that there exists $c \in \mathbb{R}_+$ such that $\sigma_c = \{\tau : \gamma(\tau) \geq c\}$ or $\sigma_c = \{\tau : \gamma(\tau) > c\}$, where $R(w, \sigma_c) \geq R(w, \sigma)$. Assume that $0 < F(\{\tau : w(\tau) > 0\} \cap \sigma) < 1$. (If $F(\{\tau : w(\tau) > 0\} \cap \sigma)$ is either 0 or 1, we are done, as $R((0, \infty)) \geq R(w, \sigma)$ and is of the desired form.)

1. First we construct $\tilde{\sigma}_c = \{\tau : \gamma(\tau) > c\} \cup C$, where $C \subseteq \{\tau : \gamma(\tau) = c\}$ and $R(w, \tilde{\sigma}_c) \geq R(w, \sigma)$.

For the given σ, c , let

$$\begin{aligned} A_c &= \{\tau : \tau \notin \sigma, \gamma(\tau) \geq c\} \\ B_c &= \{\tau : \tau \in \sigma, \gamma(\tau) < c\} \\ L(X) &= \int_{x \in X} \tau dF(\tau) & X \subseteq (0, \infty) \end{aligned}$$

A_c is a set of trips that pay more than c per unit time but are not in σ , and B_c is the set of the trips that pay less than c per unit time but are not in σ . $L(X)$ is the mean extra utilization that trips in X contribute in a renewal cycle. The idea is that if we find sets A, B such that $L(A) = L(B) > 0$ and $\gamma(a) > \gamma(b), \forall a \in A, b \in B$, then $\sigma' = \sigma \cup A \setminus B \implies R(w, \sigma') > R(w, \sigma)$: the denominator of the reward stays the same, and the numerator increases. A few facts that follow from assumptions:

- $L(A_0) > 0$
- $\exists c : L(B_c) > 0$
- $L(B_c)$ is non-decreasing as c increases, and $L(B_0) = 0$
- $L(A_c)$ is non-increasing as c increases, and $\lim_{c \rightarrow \infty} L(A_c) = 0$
- $L(A_c), L(B_c)$ both are continuous from the left in c .
- The above imply that $\exists c'$ such that $L(A_c) < L(B_c), \forall c > c'$.
- Thus, there exists $c_0 = \max\{c' : L(A_{c'}) \geq L(B_{c'})\}$

If $L(A_{c_0}) = L(B_{c_0})$, then we are done with this part: let $\tilde{\sigma}_{c_0} = \sigma \cup A_{c_0} \setminus B_{c_0} = \{\tau : \gamma(\tau) \geq c\}$.

Otherwise if $L(A_{c_0}) > L(B_{c_0})$ (which can happen if there is a mass of positive probability at the set $\{\tau : \gamma(\tau) = c\}$):

- By the definition of c_0 , for all $c > c_0$ we have $L(A_c) < L(B_c)$. Then

$$L(B_{c_0}) < L(A_{c_0}) < L(B_{c_0}) + L(\{\tau : \tau \in \sigma, \gamma(\tau) = c_0\})$$

- let $C \subseteq \{\tau : \tau \in \sigma, \gamma(\tau) = c_0\}$ such that $L(C) + L(B_{c_0}) = L(A_{c_0})$. Such C exists by F continuous.
- Let $\tilde{\sigma}_{c_0} = \sigma \cup A_{c_0} \setminus (C \cup B_{c_0})$

We now have $\tilde{\sigma}_{c_0} = \{\tau : \gamma(\tau) \geq c_0\} \setminus C$, where $C \subseteq \{\tau : \tau \in \sigma, \gamma(\tau) = c_0\}$, and $R(w, \tilde{\sigma}_{c_0}) > R(w, \sigma)$, unless σ already was of the form $\tilde{\sigma}_{c_0}$ for some c_0 .

2. Next, we construct $\sigma_{c_0} = \{\tau : \gamma(\tau) \geq c_0\}$ or $\sigma_{c_0} = \{\tau : \gamma(\tau) > c_0\}$ such that $R(w, \sigma_{c_0}) \geq R(w, \sigma)$.

- Suppose $c_0 \geq R(w, \tilde{\sigma}_{c_0})$. Then

$$\begin{aligned} R(w, \{\tau : \gamma(\tau) \geq c_0\}) &= \frac{\lambda \int_{\tau \in \tilde{\sigma}_{c_0}} w(\tau) dF(\tau) + \lambda \int_{\tau \in C} w(\tau) dF(\tau)}{1 + \lambda \int_{\tau \in \tilde{\sigma}_{c_0}} \tau dF(\tau) + \lambda \int_{\tau \in C} \tau dF(\tau)} \\ &\geq R(w, \tilde{\sigma}_{c_0}) \end{aligned} \tag{A.3}$$

Where the inequality follows from $R(w, \tilde{\sigma}_{c_0}) = \frac{\lambda \int_{\tau \in \tilde{\sigma}_{c_0}} w(\tau) dF(\tau)}{1 + \lambda \int_{\tau \in \tilde{\sigma}_{c_0}} \tau dF(\tau)}$,

$$\frac{\lambda \int_{\tau \in C} w(\tau) dF(\tau)}{\lambda \int_{\tau \in C} \tau dF(\tau)} = \frac{\lambda \int_{\tau \in C} \frac{w(\tau)}{\tau} \tau dF(\tau)}{\lambda \int_{\tau \in C} \tau dF(\tau)} = c_0, \text{ and } \frac{w}{y} \leq \frac{x}{z} \implies \frac{w+x}{y+z} \geq \frac{w}{y}.$$

Then let $\sigma_{c_0} = \{\tau : \gamma(\tau) \geq c_0\}$

- Similarly, suppose $c_0 < R(w, \tilde{\sigma}_{c_0})$. Then

$$\begin{aligned} R(w, \{\tau : \gamma(\tau) > c_0\}) &= \frac{\lambda \int_{\tau \in \tilde{\sigma}_{c_0}} w(\tau) dF(\tau) - \lambda \int_{\tau \in \{\tau : \tau \in \sigma, \gamma(\tau) = c_0\} \setminus C} w(\tau) dF(\tau)}{1 + \lambda \int_{\tau \in \tilde{\sigma}_{c_0}} \tau dF(\tau) - \lambda \int_{\tau \in \{\tau : \tau \in \sigma, \gamma(\tau) = c_0\} \setminus C} \tau dF(\tau)} \\ &> R(w, \tilde{\sigma}_{c_0}) \end{aligned} \tag{A.4}$$

Where the inequality follows from $\frac{w}{y} > \frac{x}{z} \implies \frac{w-x}{y-z} > \frac{w}{y}$.

Then let $\sigma_{c_0} = \{\tau : \gamma(\tau) > c_0\}$ (choosing arbitrarily if $c_0 = R(w, \tilde{\sigma}_{c_0})$).

Thus, we have shown that for all σ , there exists $\sigma_{c_0} = \{\tau : \gamma(\tau) > c_0\}$ or $\sigma_{c_0} = \{\tau : \gamma(\tau) \geq c_0\}$ such that $R(w, \sigma_{c_0}) \geq R(w, \sigma)$.

Let $\sigma_c^> = \{\tau : \gamma(\tau) > c\}$, and $\sigma_c^{\geq} = \{\tau : \gamma(\tau) \geq c\}$. We finish the proof by showing that $\exists c^*$ such that $\forall c, R(w, \sigma_{c^*}^{\geq}) \geq \max(R(w, \sigma_c^>), R(w, \sigma_c^{\geq}))$.

By assumption of $w(\tau)/\tau$, the reward is bounded: $0 \leq R(w, \sigma_c^>)$, $0 \leq R(w, \sigma_c^{\geq})$; further, there exists C such that $\forall c > C: R(w, \sigma_c^>) < R((0, \infty))$, $R(w, \sigma_c^{\geq}) < R((0, \infty))$ (which follows from F a i.e., , and so as $c \rightarrow \infty$, $F(\{\tau : \gamma(\tau) \geq c\}) \rightarrow 0$.

Further, $R(w, \sigma_c^>)$ is continuous from the right in c , and $R(w, \sigma_c^{\geq})$ is continuous from the left in c , and the two functions have the same points of discontinuities: c such that $F(\{\tau : \gamma(\tau) = c\}) > 0$ (and these are their only points of disagreement). Thus, the function $\max(R(w, \sigma_c^>), R(w, \sigma_c^{\geq}))$ of c attains its maximum at some $c^* \in [0, C]$.

In other words, there exists c^* such that $\forall c, \max(R(w, \sigma_{c^*}^{\geq}), R(w, \sigma_c^{\geq})) \geq \max(R(w, \sigma_c^>), R(w, \sigma_c^{\geq}))$. We finish by proving that $R(w, \sigma_{c^*}^{\geq}) \geq R(w, \sigma_{c^*}^>)$.

- Suppose $c^* \geq R(w, \sigma_{c^*}^>)$. Then, by the same argument as line (A.3), $R(w, \sigma_{c^*}^>) \leq R(w, \sigma_{c^*}^{\geq})$.
- Suppose $c^* < R(w, \sigma_{c^*}^>)$.
 - If $\exists B : c^* < B$ such that the mass $F(\{\tau : \gamma(\tau) \in (c^*, B]\}) = 0$, then note that $\sigma_{c^*}^>$ is equal to σ_B^{\geq} up to a set of measure 0, and so $R(w, \sigma_{c^*}^>) = R(w, \sigma_B^{\geq})$.
 - Otherwise, let $B : c^* < B < R(w, \sigma_{c^*}^>)$, and note that $F(\{\tau : \gamma(\tau) \in (c^*, B]\}) > 0$. Then, by the same argument as in line (A.4), $R(w, \sigma_{c^*}^>) < R(w, \sigma_B^{\geq}) \leq \max(R(w, \sigma_{c^*}^>), R(w, \sigma_{c^*}^{\geq})) = R(w, \sigma_{c^*}^{\geq})$.

Thus, $R(w, \sigma_{c^*}^{\geq}) \geq R(w, \sigma_{c^*}^>)$, for some c .

Thus, for some c^* , the policy $\sigma_{c^*}^{\geq} = \{\tau : \gamma(\tau) \geq c^*\}$ is optimal. \square

A.3.3 Proof of Proposition 2.3.1

With a single state, $w(\tau) = m\tau + a$ is incentive compatible if $0 \leq a \leq \frac{m}{\lambda}$.

Proof. Proof. Let $T = \int_{\tau \in (0, \infty)} \tau dF(\tau)$. Let $\sigma' = (0, \infty) \setminus \sigma$, for some σ .

$$\begin{aligned} R((0, \infty)) &= \frac{\lambda \int_{\tau \in (0, \infty)} w(\tau) dF(\tau)}{1 + \lambda T} \\ R(\sigma') &= \frac{\lambda \int_{\tau \in (0, \infty)} w(\tau) dF(\tau) - \lambda \int_{\tau \in \sigma} w(\tau) dF(\tau)}{1 + \lambda T - \lambda \int_{\tau \in \sigma} \tau dF(\tau)} \\ \implies R((0, \infty)) \geq R(\sigma') &\iff \frac{\lambda \int_{\tau \in (0, \infty)} w(\tau) dF(\tau)}{1 + \lambda T} \leq \frac{\int_{\tau \in \sigma} w(\tau) dF(\tau)}{\int_{\tau \in \sigma} \tau dF(\tau)} \end{aligned}$$

Where the last line follows from $\frac{w}{y} \geq \frac{w-x}{y-z} \iff \frac{w}{y} \leq \frac{x}{z}$.

Thus, a necessary and sufficient condition for incentive compatibility is that

$$\frac{\lambda \int_{\tau \in (0, \infty)} w(\tau) dF(\tau)}{1 + \lambda T} \leq \frac{\int_{\tau \in \sigma} w(\tau) dF(\tau)}{\int_{\tau \in \sigma} \tau dF(\tau)} \quad \forall \sigma.$$

Suppose $w(\tau) = m\tau + a$. Then, for $0 \leq a \leq \frac{m}{\lambda}$:

$$\begin{aligned} \frac{\lambda \int_{\tau \in (0, \infty)} w(\tau) dF(\tau)}{1 + \lambda T} &= \frac{\lambda(mT + a)}{1 + \lambda T} \\ &\leq \frac{m(\lambda T + 1)}{1 + \lambda T} = m && a \leq \frac{m}{\lambda} \\ &\leq m + a \left[\frac{F(\sigma)}{\int_{\tau \in \sigma} \tau dF(\tau)} \right] \quad \forall \sigma && a \geq 0 \\ &= \frac{\int_{\tau \in \sigma} w(\tau) dF(\tau)}{\int_{\tau \in \sigma} \tau dF(\tau)} \end{aligned}$$

□

A.3.4 Uniqueness of optimal policy for single-state model

Lemma A.3.1. *Consider the single-state model, and optimal policy σ^* of the form $\sigma^* = \{\tau : \frac{w(\tau)}{\tau} \geq c^*\}$. Then, $R(\sigma^*) = c^*$. Further, other policies of the form $\sigma_c = \{\tau : \frac{w(\tau)}{\tau} \geq c\}$ are not optimal unless $\sigma_c = \sigma^*$ (up to sets of measure 0).*

Proof. Proof. By Theorem 2.3.1, there exists an optimal policy of the form $\sigma^* = \{\tau : \frac{w(\tau)}{\tau} \geq c^*\}$, for some c^* . Here, we show (1) that $R(\sigma^*) = c^*$, and (2) this is the unique optimal policy of the form $\sigma^* = \{\tau : \frac{w(\tau)}{\tau} \geq c\}$.

1. $R(\sigma^*) = c^*$. The proof is identical to lines (A.3), (A.4).

Suppose $R(\sigma^*) > c^*$. Then, consider $c = R(\sigma^*)$, $\sigma_c = \{\tau : \frac{w(\tau)}{\tau} \geq c\}$. If $F(\sigma^* \setminus \sigma_c) > 0$:

$$\begin{aligned} R(\sigma_c) &= \frac{\lambda \int_{\tau \in \sigma^*} w(\tau) dF(\tau) - \lambda \int_{\tau \in \sigma^* \setminus \sigma_c} w(\tau) dF(\tau)}{1 + \lambda \int_{\tau \in \sigma^*} \tau dF(\tau) - \lambda \int_{\tau \in \sigma^* \setminus \sigma_c} \tau dF(\tau)} \\ &> R(\sigma^*) \end{aligned}$$

Which follows from $\frac{\lambda \int_{\tau \in \sigma^* \setminus \sigma_c} w(\tau) dF(\tau)}{\lambda \int_{\tau \in \sigma^* \setminus \sigma_c} \tau dF(\tau)} < c = R(\sigma^*) = \frac{\lambda \int_{\tau \in \sigma^*} w(\tau) dF(\tau)}{1 + \lambda \int_{\tau \in \sigma^*} \tau dF(\tau)}$, and $\frac{x}{z} < \frac{w}{y} \implies \frac{w-x}{y-z} > \frac{w}{y}$. This contradicts that σ^* is optimal.

Similarly, suppose $R(\sigma^*) < c^*$. Then, consider $c = R(\sigma^*)$, $\sigma_c = \{\tau : \frac{w(\tau)}{\tau} \geq c\}$. If $F(\sigma_c \setminus \sigma^*) >$

0:

$$\begin{aligned} R(\sigma_c) &= \frac{\lambda \int_{\tau \in \sigma^*} w(\tau) dF(\tau) + \lambda \int_{\tau \in \sigma_c \setminus \sigma^*} w(\tau) dF(\tau)}{1 + \lambda \int_{\tau \in \sigma^*} \tau dF(\tau) + \lambda \int_{\tau \in \sigma_c \setminus \sigma^*} \tau dF(\tau)} \\ &> R(\sigma^*) \end{aligned}$$

Which follows from $\frac{\lambda \int_{\tau \in \sigma_c \setminus \sigma^*} w(\tau) dF(\tau)}{\lambda \int_{\tau \in \sigma_c \setminus \sigma^*} \tau dF(\tau)} > c = R(\sigma^*) = \frac{\lambda \int_{\tau \in \sigma^*} w(\tau) dF(\tau)}{1 + \lambda \int_{\tau \in \sigma^*} \tau dF(\tau)}$, and $\frac{x}{z} > \frac{w}{y} \implies \frac{w+x}{y+z} > \frac{w}{y}$. This contradicts that σ^* is optimal.

2. Suppose there were two optimal $\sigma_{c_1}, \sigma_{c_2}$ of the appropriate form. Without loss of generality, let $c_1 < c_2$. Suppose $F(\sigma_{c_1} \setminus \sigma_{c_2}) > 0$, and so $\sigma_{c_1} \subsetneq \sigma_{c_2}$. By the previous part, $R(\sigma_{c_1}) = c_1, R(\sigma_{c_2}) = c_2$. Then

$$\begin{aligned} R(\sigma_{c_2}) &= \frac{\lambda \int_{\tau \in \sigma_{c_1}} w(\tau) dF(\tau) - \lambda \int_{\tau \in \sigma_{c_1} \setminus \sigma_{c_2}} w(\tau) dF(\tau)}{1 + \lambda \int_{\tau \in \sigma_{c_1}} \tau dF(\tau) - \lambda \int_{\tau \in \sigma_{c_1} \setminus \sigma_{c_2}} \tau dF(\tau)} \\ &< R(\sigma_{c_1}) \end{aligned}$$

Which follows from $\frac{\lambda \int_{\tau \in \sigma_{c_1} \setminus \sigma_{c_2}} w(\tau) dF(\tau)}{\lambda \int_{\tau \in \sigma_{c_1} \setminus \sigma_{c_2}} \tau dF(\tau)} > c_1 = R(\sigma_{c_1}) = \frac{\lambda \int_{\tau \in \sigma_{c_1}} w(\tau) dF(\tau)}{1 + \lambda \int_{\tau \in \sigma_{c_1}} \tau dF(\tau)}$, and $\frac{x}{z} > \frac{w}{y} \implies \frac{w}{y} > \frac{w-x}{y-z}$, contradicting the supposition that $R(\sigma_{c_1}) = c_1 < c_2 = R(\sigma_{c_2})$.

□

A.4 Proofs of dynamic model results

In this section, we provide proofs of the theorems and lemmas in the main text regarding the dynamic model. Section A.4.1 contains proofs for the dynamic model lemmas regarding driver reward and time spent in each state, Lemmas 2.2.1, 2.4.1, and 2.4.2.

Section A.4.2 contains an overview of the proof strategy for both Theorems 2.3.2 and 2.4.1, and in particular provides a statement of the main technical lemma used to prove both theorems. The proof for this lemma is deferred to Section A.4.5.

Section A.4.3 contains the statements of several appendix-only lemmas that are used to prove the main results. Proofs for these lemmas are deferred to Section A.4.6, as they are algebraically tedious.

Finally, Section A.4.4 contains the proofs for our main results, Theorems 2.3.2 and 2.4.1.

A.4.1 Driver reward

Lemma 2.2.1. *In the dynamic model, the earnings rate can be decomposed into each state i earnings rate $R_i(w_i, \sigma_i)$ and fraction of time $\mu_i(\sigma)$ spent in state i :*

$$R(w, \sigma) = \mu_1(\sigma)R_1(w_1, \sigma_1) + \mu_2(\sigma)R_2(w_2, \sigma_2) \quad \text{with probability 1.}$$

As in the single-state model, $R_i(w_i, \sigma_i) = \frac{W_i(\sigma_i)}{T_i(\sigma_i)}$, where

$$W_i(\sigma_i) = \frac{1}{F_i(\sigma_i)} \int_{\tau \in \sigma_i} w_i(\tau) dF_i(\tau), \quad T_i(\sigma_i) = \frac{1}{\lambda_i F_i(\sigma_i)} + \frac{1}{F_i(\sigma_i)} \int_{\tau \in \sigma_i} \tau dF_i(\tau)$$

Proof. Proof. Consider the renewal process (with cycles and sub-cycles) defined in the main text.

We use the following notation

- $M(t)$ is the total number of cycles that have been completed up to time t
- $N_j(M)$ is the number of sub-cycles in state j in the M th cycle – i.e., in the M th cycle of the single renewal process described above, the number of times that the driver is open in state j (after transitioning from the other state, or finishing a trip that started in the same state j)
- $S_j(k, M)$ is the length of the k th such sub-cycle in the M th cycle, with expected length $S_j(\sigma_j)$
- $W_j(k, M)$ is the earnings of the driver in the k th such sub-cycle in the M th cycle, with expected value $W_j(\sigma_j)$
- $p_{ji}(\sigma_j)$ is the probability that the current sub-cycle is the last in state j for the current cycle – as the next sub-cycle starts in the other state.
- $R_j(w_j, \sigma_j, M)$ is the total amount earned in state j after M such cycles

Then:

$$\begin{aligned} R_j(w_j, \sigma_j, M(t)) &= \sum_{M=1}^{M(t)} \sum_{k=1}^{N_j(M)} W_j(k, M) \\ \lim_{t \rightarrow \infty} \frac{R_j(w_j, \sigma_j, M(t))}{M(t)} &= \lim_{t \rightarrow \infty} \frac{1}{M(t)} \left[\sum_{M=1}^{M(t)} \sum_{k=1}^{N_j(M)} W_j(k, M) \right] \\ &= \frac{1}{p_{ji}(\sigma_j)} W_j(\sigma_j) \quad \text{almost surely} \end{aligned}$$

by the mean of a geometric random variable ($\frac{1}{p_{ji}(\sigma_j)}$ is the expected number of sub-cycles in j in a given cycle) and the basic law of large numbers for renewal processes.

Similarly, we know that $\frac{M(t)}{t}$ converges to its mean almost surely as $t \rightarrow \infty$, where the mean is based on the length of time in each state in each cycle. Let $\tilde{S}(\sigma)$ be the expected length of one of these cycles. Then:

$$\begin{aligned} \lim_{t \rightarrow \infty} \frac{M(t)}{t} &= \frac{1}{\tilde{S}(\sigma)} \\ \tilde{S}(\sigma) &= \mathbb{E} \left[\sum_{k=1}^{N_1(1)} S_1(k, 1) \right] + \mathbb{E} \left[\sum_{k=1}^{N_2(1)} S_2(k, 1) \right] \\ &= \mathbb{E}[N_1(1)]\mathbb{E}[S_1(k, 1)] + \mathbb{E}[N_2(1)]\mathbb{E}[S_2(k, 1)] && \text{Wald's identity} \\ &= \frac{1}{p_{12}(\sigma_1)} S_1(\sigma_1) + \frac{1}{p_{21}(\sigma_2)} S_2(\sigma_2) \\ \implies \lim_{t \rightarrow \infty} \frac{M(t)}{t} &= \frac{p_{21}(\sigma_2)p_{12}(\sigma_1)}{p_{21}(\sigma_2)S_1(\sigma_1) + p_{12}(\sigma_1)S_2(\sigma_2)} \end{aligned}$$

Then, by standard algebra on multiplication with almost sure convergence

$$\begin{aligned} \lim_{t \rightarrow \infty} \frac{R_j(w_j, \sigma_j, M(t))}{t} &= \lim_{t \rightarrow \infty} \frac{R_j(w_j, \sigma_j, M(t))}{M(t)} \frac{M(t)}{t} \\ &= \frac{1}{p_{ji}(\sigma_j)} W_j(w_j, \sigma_j) \left[\frac{p_{21}(\sigma_2)p_{12}(\sigma_1)}{p_{21}(\sigma_2)S_1(\sigma_1) + p_{12}(\sigma_1)S_2(\sigma_2)} \right] \\ &= \left[\frac{p_{ij}(\sigma_i)S_j(\sigma_j)}{p_{21}(\sigma_2)S_1(\sigma_1) + p_{12}(\sigma_1)S_2(\sigma_2)} \right] R_j(w_j, \sigma_j) \end{aligned}$$

Let $\mu_j(\sigma) \triangleq \frac{p_{ij}(\sigma_i)S_j(\sigma_j)}{p_{21}(\sigma_2)S_1(\sigma_1) + p_{12}(\sigma_1)S_2(\sigma_2)}$. Putting everything together:

$$\begin{aligned} \liminf_{t \rightarrow \infty} \frac{R(w, \sigma, t)}{t} &= \liminf_{t \rightarrow \infty} \frac{R_1(w_1, \sigma_1, M(t))}{t} + \liminf_{t \rightarrow \infty} \frac{R_2(w_2, \sigma_2, M(t))}{t} \\ &= \mu_1(\sigma)R_1(w_1, \sigma_1) + \mu_2(\sigma)R_2(w_2, \sigma_2) \end{aligned}$$

□

Lemma 2.4.1. *Suppose the world is in state i at time t . Let $q_{i \rightarrow j}(s)$ denote the probability that the world will be in state $j \neq i$ at time $t + s$. Then,*

$$q_{i \rightarrow j}(s) = \frac{\lambda_{i \rightarrow j}}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \left[1 - e^{-(\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i})s} \right]$$

Proof. Proof. Given the state dynamics in the model, $q_{i \rightarrow j}(s)$ is determined by the evolution of a CTMC in time s , given that the current state is i . We can use standard CTMC results here. Let Q

denote the Q -matrix for the world state CTMC. From the model definition,

$$Q = \begin{bmatrix} -\lambda_{1 \rightarrow 2} & \lambda_{1 \rightarrow 2} \\ \lambda_{2 \rightarrow 1} & -\lambda_{2 \rightarrow 1} \end{bmatrix}$$

Recall that the state transition matrix after time t is then given by the matrix exponential e^{Qt} , which is equal to the inverse of the Laplace transform of the inverse of the resolvent of Q :

$$\begin{aligned} q_{i \rightarrow j}(\tau) &= (e^{Q\tau})_{ij} \\ &= \mathcal{L}^{-1}((wI - Q)_{ij}^{-1})(\tau) && w \text{ is a Laplace transform parameter} \\ &= \frac{\lambda_{i \rightarrow j}}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \left[1 - e^{-(\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i})\tau} \right] \end{aligned}$$

where the closed form in the last line emerges due to the 2 state model assumption. \square

Lemma 2.4.2. *Let $T_i(\sigma_i)$ be as defined in Lemma 2.2.1. The fraction of time a driver following strategy $\sigma = \{\sigma_1, \sigma_2\}$ spends either open in state i or on a trip started in state i is*

$$\begin{aligned} \mu_i(\sigma) &= \frac{\lambda_i F_i(\sigma_i) T_i(\sigma_i) Q_j(\sigma_j)}{\lambda_j F_j(\sigma_j) T_j(\sigma_j) Q_i(\sigma_i) + \lambda_i F_i(\sigma_i) T_i(\sigma_i) Q_j(\sigma_j)} \\ \text{where } Q_i(\sigma_i) &= \lambda_{i \rightarrow j} + \lambda_i \int_{\tau \in \sigma_i} q_{i \rightarrow j}(\tau) dF_i(\tau) \end{aligned}$$

Proof. Proof. From the proof of Lemma 2.2.1, we have

$$\mu_i(\sigma) = \frac{p_{ji}(\sigma_j) S_i(\sigma_i)}{p_{21}(\sigma_2) S_1(\sigma_1) + p_{12}(\sigma_1) S_2(\sigma_2)}$$

where $S_i(\sigma_i)$ is the expected length of the time between being open in a state i to being open again, either after a state transition or after finishing a job; and $p_{ij}(\sigma_i)$ is the probability that the driver is next open in state j given they are currently open in state i . These are:

$$\begin{aligned} S_i(\sigma_i) &= \frac{1}{\lambda_i F_i(\sigma_i) + \lambda_{i \rightarrow j}} + \frac{\lambda_i F_i(\sigma_i)}{\lambda_i F_i(\sigma_i) + \lambda_{i \rightarrow j}} \int_{\tau \in \sigma_i} \tau \frac{f_i(\tau)}{F_i(\sigma_i)} d\tau \\ &= \frac{1}{\lambda_i F_i(\sigma_i) + \lambda_{i \rightarrow j}} \left[1 + \lambda_i \int_{\tau \in \sigma_i} \tau dF_i(\tau) \right] \\ &= \left[\frac{\lambda_i F_i(\sigma_i)}{\lambda_i F_i(\sigma_i) + \lambda_{i \rightarrow j}} \right] T_i(\sigma_i) \end{aligned}$$

The first part of the sum $\frac{1}{\lambda_i F_i(\sigma_i) + \lambda_{i \rightarrow j}}$ is the expected time until either the driver receives and accepts a request, or the world state transitions to the other state. This form emerges because there are two competing independent exponential clocks – that for a request and that for the world state changing. The second part of the sum is the probability of receiving an accepted trip request before

a state transition, times the expected length of an accepted trip.

The next step is to find an expression for $p_{ij}(\sigma_i)$, the probability that the next renewal cycle is at state j , given the current one is at state i . We find it for $j \neq i$, and then $p_{ii} = 1 - p_{ij}$.

$$\begin{aligned} p_{ij}(\sigma) &= \frac{\lambda_{i \rightarrow j}}{\lambda_i F_i(\sigma_i) + \lambda_{i \rightarrow j}} + \frac{\lambda_i F_i(\sigma_i)}{\lambda_i F_i(\sigma_i) + \lambda_{i \rightarrow j}} \frac{1}{F_i(\sigma_i)} \int_{\sigma_i} q_{i \rightarrow j}(\tau) dF_i(\tau) \\ &= \left[\frac{1}{\lambda_i F_i(\sigma_i) + \lambda_{i \rightarrow j}} \right] Q_i(\sigma_i) \end{aligned}$$

The first part of the summation is the probability that the world state transitions to state j before the driver accepts a trip request. The second part is the probability that the driver accepts a trip request before the state transitions, times the probability $q_{i \rightarrow j}(\sigma_i) = \frac{1}{F_i(\sigma_i)} \int_{\sigma_i} q_{i \rightarrow j}(\tau) dF_i(\tau)$ that the world will be in state j when the driver's trip ends. The result follows. \square

A.4.2 Proof strategy for incentive compatible pricing and structural results

We now give an overview of the proof strategy for both Theorems 2.3.2 and 2.4.1. The key step to both is Lemma A.4.1 below, which shows how to use properties of the derivative of a reward function with respect to an element of a driver policy, to establish the structural properties of optimal driver policies. Then, Section A.4.3 provides lemmas that help us establish the properties of this derivative as they depend on the pricing function. We put things together in Section A.4.4 to prove Theorems 2.3.2 and 2.4.1. Proofs of Lemma A.4.1 and the lemmas in Section A.4.3 are in Sections A.4.5 and A.4.6, respectively.

Lemma A.4.1. *Consider a function $\hat{R}(\sigma)$ that maps open, measurable subsets $\sigma = \cup_k^\infty (\ell_k, u_k) \subseteq (0, \infty)$ to the non-negative reals, and probability measure F such that F is continuous, i.e. f is bounded.*

Let $\frac{\partial}{\partial u} \hat{R}(\sigma)$ denote the partial derivative of \hat{R} with respect to an upper end-point u_k of the intervals that make up $\sigma = \cup_k^\infty (\ell_k, u_k)$, i.e., it is the infinitesimal gain in the driver reward by adding u to the driver policy.

Consider an open measurable subset $\sigma' \subseteq (0, \infty)$. Suppose,

1. $F(\sigma') > 0$, and $\hat{R}(\sigma') > \hat{R}(\emptyset)$.
2. $\hat{R}(\sigma)$ is continuous in σ , and $\frac{\partial}{\partial u} \hat{R}(\sigma)$ exists, for all σ and its endpoints u_k .
3. $\frac{\partial}{\partial u} \hat{R}(\sigma)$ is continuous in u , for each fixed σ .
4. $\frac{\partial}{\partial u} \hat{R}(\sigma)$ is continuous in σ , for each fixed u .

Finally, suppose that there exists a function $r(u, \sigma)$ that has the same sign as $\frac{\partial}{\partial u} \hat{R}(\sigma)$, for all u, σ and has one of the following properties. Then, each of the following hold, depending on the properties of $r(u, \sigma)$.

- Suppose $r(u, \sigma)$ is **non-negative** for all u, σ . Then $\hat{R}((0, \infty)) \geq \hat{R}(\sigma')$.
- Suppose $\exists \epsilon > 0$ s.t. $r(u, \sigma)$ is **strictly increasing** in u (for a fixed σ), for all σ such that $\hat{R}(\sigma) \geq \hat{R}(\sigma') - \epsilon$. Then, there exists a value $\ell^* \in \mathbb{R}_+ \cup \{\infty\}$ such that $\hat{R}((\ell^*, \infty)) \geq \hat{R}(\sigma')$.
- Suppose $\exists \epsilon > 0$ s.t. $r(u, \sigma)$ is **strictly quasi-convex** in u (for a fixed σ), for all σ such that $\hat{R}(\sigma) \geq \hat{R}(\sigma') - \epsilon$. Then, there exists exist $\ell^*, u^* \in \mathbb{R}_+ \cup \{\infty\}$ such that $\hat{R}((0, \ell^*) \cup (u^*, \infty)) \geq \hat{R}(\sigma')$, and it is not the case that both $\ell^* = 0, u^* = \infty$.
- Suppose $\exists \epsilon > 0$ s.t. $r(u, \sigma)$ is **strictly decreasing** in u (for a fixed σ), for all σ such that $\hat{R}(\sigma) \geq \hat{R}(\sigma') - \epsilon$. Then, there exists a value $u^* \in \mathbb{R}_+ \cup \{\infty\}$ such that $\hat{R}((0, u^*)) \geq \hat{R}(\sigma')$.
- Suppose $\exists \epsilon > 0$ s.t. $r(u, \sigma)$ is **strictly quasi-concave** in u (for a fixed σ), for all σ such that $\hat{R}(\sigma) \geq \hat{R}(\sigma') - \epsilon$. Then, there exists exist $\ell^*, u^* \in \mathbb{R}_+ \cup \{\infty\}$ such that $\hat{R}((\ell^*, u^*)) \geq \hat{R}(\sigma')$.

A.4.3 Necessary lemmas

Here we present lemmas necessary to prove the main theorems regarding incentive compatibility and optimal driver policies. Proofs are deferred to Section A.4.6.

These lemmas primarily involve properties of derivatives of the reward function $R(w, \sigma)$ and its components in the dynamic model, as a function of the pricing.

Notation and assumptions

Recall in the dynamic model that we constrain σ_i to be measurable, *open*, subsets of the \mathbb{R}_+ . Then, σ_i can be written as a countable union of disjoint subsets of \mathbb{R}_+ , i.e. $\sigma_i = \cup_{k=0}^{\infty} (\ell_k, u_k)$. We further assume that $u_k \neq \ell_m$, for any k, m ; we can do so without loss of generality by making a measure 0 change to σ_i , by adding $u_k = \ell_m$ to σ_i .

Suppose u is an upper-endpoint of σ_i , ie. $\exists k$ such that $u = u_k$. Then, we use $\frac{\partial}{\partial u} H(\sigma_i)$ to denote the derivative of the set function H with respect to u at σ_i . Similarly, $\frac{\partial}{\partial \ell} H(\sigma_i)$ is the derivative of H at σ_i with respect to a lower-endpoint of σ_i .

Note that we also derive $\frac{\partial}{\partial u} R(w, \{\sigma_1, \sigma_2\})$, $\frac{\partial}{\partial \ell} R(w, \{\sigma_1, \sigma_2\})$. We will make it clear in each instance whether u or ℓ is an endpoint of σ_1 or σ_2 . For all the derivatives in this subsection $\frac{\partial}{\partial u}$ refers to the derivative with respect to an upper endpoint in σ_i .

Furthermore:

- We use σ in the function argument when the function depends on policies in both states, and σ_i when it only depends on the policy in state i .

- We use \propto to denote *has the same sign as*, rather than *proportional to*
- All policy equalities are up to measure 0.

Let

$$\Delta(\sigma_i, \sigma_j) = R_i(w_i, \sigma_i) - R_j(w_j, \sigma_j).$$

Finally, when σ_i, σ_j are clear from context, let

$$\begin{aligned} Q_i &\triangleq Q_i(\sigma_i) = \lambda_{i \rightarrow j} + \lambda_i \int_{\sigma_i} q_{i \rightarrow j}(\tau) dF_i(\tau) \\ T_i &\triangleq \lambda_i F_i(\sigma_i) T_i(\sigma_i) = 1 + \lambda_i \int_{\tau \in \sigma_i} \tau dF_i(\tau) \\ W_i &\triangleq \lambda_i F_i(\sigma_i) W_i(\sigma_i) = \lambda_i \int_{\tau \in \sigma_i} w_i(\tau) dF_i(\tau) \\ \Delta_{ji} &\triangleq \Delta(\sigma_j, \sigma_i) = R_j(w_j, \sigma_j) - R_i(w_i, \sigma_i) \end{aligned}$$

We assume throughout:

- Distribution of jobs F, F_i is a continuous probability measure, i.e., f, f_i bounded.
- There exists a policy in state 2 that dominates state 1: $\exists \sigma_2$ such that $\Delta(\sigma_2, \sigma_1) > 0, \forall \sigma_1 \subseteq (0, \infty)$.
- σ, σ_i constrained to be measurable with respect to F, F_i , and σ_i are open.

Lemmas for driver policy in response to affine pricing

Lemma A.4.2. *Let*

$$r(u, i, w, \sigma) \triangleq \frac{q_{i \rightarrow j}(u)}{u} \Delta_{ji} + \frac{w_i(u)}{u} \left(\frac{Q_i}{T_i} + \frac{Q_j}{T_j} \right) - \left(\frac{Q_i}{T_i} R_j + \frac{Q_j}{T_j} R_i \right)$$

Then, $\frac{\partial}{\partial u} R(w, \sigma) \propto r(u, i, w, \sigma)$.

In other words, $r(u, i, w, \sigma)$ has the *same sign* as the derivative of the overall reward with respect to u (an upper endpoint of σ_i) at w, σ , but it is not necessarily monotonic with it.

Remark A.4.1. *Given assumptions on F_i, w_i :*

- $R_i(\sigma), R(\sigma), \mu_i$ are continuous in σ
- $\frac{\partial}{\partial u} R(w, \sigma), r(u, i, w, \sigma)$ are both continuous u (for fixed σ), and continuous in σ .
- $\frac{q_{i \rightarrow j}(u)}{u}$ is strictly decreasing in u .

- If $\Delta_{ji} < 0$ (i.e. $i = 2$ the surge state) and $\frac{w_i(u)}{u}$ is non-decreasing in u , then $r(u, i, w, \sigma)$ is strictly increasing in u for a fixed σ . Thus, $\frac{\partial}{\partial u}R(w, \sigma)$ is negative up to a certain point $U \in (0, \infty) \cup \{\infty\}$ and then positive thereafter.
- If $\Delta_{ji} > 0$ (i.e. $i = 1$ the non-surge state) and $\frac{w_i(u)}{u}$ is non-increasing in u , then $r(u, i, w, \sigma)$ is strictly decreasing in u for a fixed σ . Thus, $\frac{\partial}{\partial u}R(w, \sigma)$ is positive up to a certain point $U \in (0, \infty) \cup \{\infty\}$ and then negative thereafter.
- $\frac{\partial}{\partial \ell}R(w, \sigma)$ at a lower endpoint of σ_i is just the negative of the derivative at the same place if a lower endpoint.

Lemma A.4.3. Suppose $w_i(\tau) = m\tau + a$, where $m, a > 0$. Then, $r(u, i, w, \sigma)$ is strictly quasi-convex in u , for each fixed σ where $\Delta_{ji} < 0$.

Lemma A.4.4. Suppose $w_i(\tau) = m\tau + a$, where $m > 0$ and $a < 0$. Then, $r(u, i, w, \sigma)$ is strictly quasi-concave in u , for each fixed σ where $\Delta_{ji} > 0$.

Lemmas for IC policy

Remark A.4.2.

$$\text{Let } w_i(u) = mu + zq_{i \rightarrow j}(u)$$

$$\text{Then } W_i = m(T_i - 1) + z(Q_i - \lambda_{i \rightarrow j})$$

$$\begin{aligned} \frac{\partial}{\partial u}R(w, \sigma) \propto q_{i \rightarrow j}(u) [(R_j - m)T_j T_i + mT_j + zQ_j T_i + zT_j \lambda_{i \rightarrow j}] \\ + u [Q_i T_j (m - R_j) + Q_j (m - zQ_i + z\lambda_{i \rightarrow j})] \end{aligned}$$

Remark A.4.3. $\lim_{u \rightarrow 0} \frac{q_{i \rightarrow j}(u)}{u} = \lambda_{i \rightarrow j}$.

Proof. Proof. Simple application of L'Hopital's rule.

$$\lim_{u \rightarrow 0} \frac{q_{i \rightarrow j}(u)}{u} = \lim_{u \rightarrow 0} \frac{\partial}{\partial u} q_{i \rightarrow j}(u) = \lim_{u \rightarrow 0} \frac{\partial}{\partial u} \frac{\lambda_{i \rightarrow j}}{\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i}} \left[1 - e^{-(\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i})u} \right] = \lambda_{i \rightarrow j}$$

□

Remark A.4.4. $\lambda_{i \rightarrow j} T_i - Q_i \geq 0$ and maximized when $\sigma_i = (0, \infty)$. Similarly, $Q_i \geq 0$ and maximized when $\sigma_i = (0, \infty)$.

In the next lemma, we consider u an upper endpoint of σ_2 , and so $\frac{\partial}{\partial u}R(w = \{w_1, w_2\}, \sigma = \{\sigma_1, \sigma_2\})$ is a derivative with respect to an upper endpoint of σ_2 .

Lemma A.4.5. Fix arbitrary σ_1 , and thus Q_1, T_1, R_1 . Let \bar{Q}_2, \bar{T}_2 be the respective values of Q_2, T_2 at $\sigma_2 = (0, \infty)$. Let $w_2(\tau) = m\tau + zq_{2 \rightarrow 1}(\tau)$, where $m > R_1$.

If

$$\frac{T_1(\lambda_{2 \rightarrow 1}\bar{T}_2 - \bar{Q}_2) - (Q_1 + T_1\lambda_{2 \rightarrow 1})}{(Q_1(\lambda_{2 \rightarrow 1}\bar{T}_2 - \bar{Q}_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1}))} \leq \frac{z}{m - R_1} \leq \frac{\bar{Q}_2 T_1 + Q_1}{Q_1(\bar{Q}_2 - \lambda_{2 \rightarrow 1})}$$

Then $\frac{\partial}{\partial u} R(w, \sigma) \geq 0$, for all u, σ_2 . Furthermore, the constraint set is feasible regardless of the primitives.

We can now do the same thing for the first state, assuming that $w_1(\tau)$ is of the form $w_1(\tau) = m\tau + zq_{1 \rightarrow 2}(\tau)$, where now $z \leq$ and $m = R_2$. In the next lemma, we consider u an upper endpoint of σ_1 , and so $\frac{\partial}{\partial u} R(w = \{w_1, w_2\}, \sigma = \{\sigma_1, \sigma_2\})$ is a derivative with respect to an upper endpoint of σ_1 . Then,

Lemma A.4.6. Fix arbitrary σ_2 , and thus Q_2, T_2, R_2 . Let \bar{Q}_1, \bar{T}_1 be the respective values of Q_1, T_1 at $\sigma_1 = (0, \infty)$. Let $w_1(\tau) = m\tau + zq_{1 \rightarrow 2}(\tau)$, where $m = R_2$.

If

$$-\frac{(T_2\lambda_{1 \rightarrow 2} + Q_2)}{Q_2(\lambda_{1 \rightarrow 2}\bar{T}_1 - \bar{Q}_1) + \lambda_{1 \rightarrow 2}(T_2\lambda_{1 \rightarrow 2} + Q_2)} \leq \frac{z}{R_2} \leq \frac{1}{(\bar{Q}_1 - \lambda_{1 \rightarrow 2})}$$

Then $\frac{\partial}{\partial u} R(w, \sigma) \geq 0$, for all u, σ_1 . Furthermore, the constraint set is feasible regardless of the primitives.

A.4.4 Proofs of main results, Theorems 2.3.2 and 2.4.1

We are now ready to combine the results above to prove our main results. The following theorem subsumes Theorem 2.3.2, (slightly expanding it to make it useful to prove Theorem 2.4.1).

Theorem A.4.1. Consider pricing function $w = \{w_1, w_2\}$, where $i = 2$ is the surge state as defined. Then, there exists an optimal policy $\sigma = \{\sigma_1, \sigma_2\}$ that maximizes $R(w, \sigma)$, with the following properties.

- Non-surge state driver optimal policy σ_1 :
 - If $w_1(\tau) = m_1\tau + a_1$, for $a_1 \geq 0$, then $\sigma_1 = (0, t_1)$, for some $t_1 \in [0, \infty) \cup \{\infty\}$.
 - If $w_1(\tau) = m_1\tau - a_1$, for $a_1 > 0$, then $\sigma_1 = (t_2, t_3)$, for some $t_2, t_3 \in [0, \infty) \cup \{\infty\}$.
 - If w_1 such that $\frac{\partial}{\partial u} R(w, \sigma' = \{\sigma'_1, \sigma'_2\}) \geq 0$ for all σ' , where u is an upper endpoint of an interval that makes up σ'_1 , then $\sigma_1 = (0, \infty)$.
- Surge state driver optimal policy σ_2 :

- If $w_2(\tau) = m_2\tau - a_2$, for $a_2 \geq 0$, then $\sigma_1 = (t_4, \infty)$, for some $t_4 \in [0, \infty)$.
- If $w_2(\tau) = m_2\tau + a_2$, for $a_2 > 0$, then $\sigma_1 = (0, t_5) \cup (t_6, \infty)$, for some $t_5, t_6 \in [0, \infty) \cup \{\infty\}$.
- If w_2 such that $\frac{\partial}{\partial u} R(w, \sigma' = \{\sigma'_1, \sigma'_2\}) \geq 0$ for all σ' , where u is an upper endpoint of an interval that makes up σ'_2 , then $\sigma_2 = (0, \infty)$.

Proof. Proof. The proof strategy is as follows:

- Start with some arbitrary policy $\sigma = \{\sigma_1, \sigma_2\}$.
- With assumption on the surge state providing higher potential earnings, replace σ_2 with a policy that provides higher earnings in state 2 than σ_1 does in state 1, without decreasing total reward.
- Using Lemma A.4.1, replace σ_1 with policy of the appropriate form, without decreasing total reward.
- Using Lemma A.4.1, replace σ_2 with policy of the appropriate form, without decreasing total reward.

Let $\Delta(\sigma_i, \sigma_{-i}) = R_i(\sigma_i) - R_j(\sigma_{-i})$, where $\sigma_{-i} \triangleq \sigma_{3-i}$. Let $r(u, i, w, \sigma)$ be a function that has the same sign as $\frac{\partial}{\partial u} R(w, \sigma)$, where u is an upper endpoint of an interval that is part of σ_i . In various results above, we show

- (Remark A.4.1). $\Delta(\sigma_i, \sigma_{-i}) > 0$ and $\frac{w_i(\tau)}{\tau}$ non-decreasing implies $r(u, i, w, \sigma)$ strictly increasing in $u \in \sigma_i$.
- (Remark A.4.1). $\Delta(\sigma_i, \sigma_{-i}) < 0$ and $\frac{w_i(\tau)}{\tau}$ non-increasing implies $r(u, i, w, \sigma)$ strictly decreasing in $u \in \sigma_i$.
- (Lemma A.4.3). $w(\tau) = m\tau + a$ for $m, a > 0$ and $\Delta(\sigma_i, \sigma_{-i}) > 0$ implies $r(u, i, w, \sigma)$ is strictly quasi-convex in $u \in \sigma_i$.
- (Lemma A.4.4). $w(\tau) = m\tau - a$ for $m, a > 0$ and $\Delta(\sigma_i, \sigma_{-i}) < 0$ implies $r(u, i, w, \sigma)$ is strictly quasi-concave in $u \in \sigma_i$.

We need to show that there exists a σ of the appropriate form such that $R(w, \sigma) \geq R(w, \sigma')$, for all σ' .

Start with arbitrary $\sigma' = \{\sigma'_1, \sigma'_2\}$ where $\sigma'_1, \sigma'_2 \subseteq \mathbb{R}_+$ are open, measurable sets, but not of the correct form in the theorem statement. Invoking the theorems in Section A.4.5 as appropriate, we construct a sequence of changes to σ' such that the overall reward does not decrease with each change, and the sequence ends with a policy consistent with the theorem statement.

Step A First, we replace σ'_2 with a policy σ_2^A such that $R_2(\sigma_2^A) > R_1(\sigma_1), \forall \sigma_1$. This allows us to cite the appropriate theorems regarding the properties of the derivative of R , that only hold when the surge state provides higher earnings than the non-surge state.

Let σ_2^A be such that $\Delta(\sigma_2^A, \sigma''_1) > 0$, for all σ''_1 open and measurable, and $\sigma_2^A \geq \sigma'_2$. Such σ_2^A exists by our assumptions on F_i, w_i . Then, let $\sigma^A \triangleq \{\sigma_1^A = \sigma'_1, \sigma_2^A\}$. $R(w, \sigma^A) \geq R(w, \sigma')$: time spent earning reward at the rate of $R_2(w_2, \sigma'_2)$ is replaced by time spent earning at rate $R_1(w_1, \sigma'_1)$ or earning at rate $R_2(w_2, \sigma_2^A)$; time spent earning at $R_1(w_1, \sigma'_1)$ may be replaced by time earning at rate $R_2(w_2, \sigma_2^A)$.

Step B Now, we replace σ_1 with a policy that is of the appropriate form.

For $i = 1$: Let $\hat{R}(\sigma_1) \triangleq R(w, \{\sigma_1, \sigma_2^A\})$.

By Lemma A.4.1, there exists σ_1^B such that $R(w, \{\sigma_1^B, \sigma_2^A\}) \geq R(w, \{\sigma_1^A, \sigma_2^A\})$, and σ_1^B is of the required form according to the table. Let $\sigma^B \triangleq \{\sigma_1^B, \sigma_2^B = \sigma_2^A\}$.

Note that all the assumptions of the theorem are met for each appropriate case: σ_2^B such that $\Delta(\sigma_2^B, \sigma'_1) > 0, \forall \sigma'_1$, and so the scaled derivatives remain decreasing / strictly quasi-concave as necessary.

Step C For $i = 2$: Now, we replace σ_2 with a policy that is of the appropriate form.

Let $\hat{R}(\sigma_2) \triangleq R(w, \{\sigma_1^B, \sigma_2\})$.

By Lemma A.4.1, there exists σ_2^C such that $R(w, \{\sigma_1^B, \sigma_2^C\}) \geq R(w, \{\sigma_1^B, \sigma_2^B\})$, and σ_2^C is of the required form according to the table. Let $\sigma^C \triangleq \{\sigma_1^C = \sigma_1^B, \sigma_2^C\}$.

Note that, in this step, we need to confirm the assumption that $r(u, 2, w, \sigma)$ remaining strictly increasing / strictly quasi-convex in u for a fixed σ , for all σ such that $\hat{R}(\sigma) \geq \hat{R}(\sigma_2^B)$ or $F(\sigma_2^B \setminus \sigma \cup \sigma \setminus \sigma_2^B) < \delta$, for some $\delta > 0$.

The chief concern is that because σ_2 is changing within the appropriate theorem, $\Delta(\sigma_2, \sigma_1^B)$ may not remain greater than 0, and so this condition might not be met. However this is not the case: $\Delta(\sigma_2, \sigma_1^B)$ remains positive – by continuity, σ_2 close to σ_2^B (by measure of set difference) implies $\Delta(\sigma_2, \sigma_1^B)$ positive. Furthermore

$$\begin{aligned} \hat{R}(\sigma''_2) \geq \hat{R}(\sigma_2^B) &\iff R(w, \{\sigma_1^B, \sigma''_2\}) \geq R(w, \{\sigma_1^B, \sigma_2^B\}) && \text{definition of } \hat{R} \\ &\iff \pi_1 R_1(w_1, \sigma_1^B) + \pi_2 R_2(w_2, \sigma''_2) \geq \pi_a R_1(w_1, \sigma_1^B) + \pi_b R_2(w_2, \sigma_2^B) && (\pi_k \text{ policy dependent}) \\ \Delta(\sigma_2, \sigma_1^B) \leq 0 &\implies \pi_1 R_1(w_1, \sigma_1^B) + \pi_2 R_2(w_2, \sigma''_2) \leq R_1(w_1, \sigma_1^B) \\ \Delta(\sigma_2^B, \sigma_1^B) > 0, \pi_b > 0 &\implies \pi_1 R_1(w_1, \sigma_1^B) + \pi_2 R_2(w_2, \sigma''_2) > R_1(w_1, \sigma_1^B) \end{aligned}$$

By **Step A**, $\Delta(\sigma_2^B, \sigma_1^B) > 0, \pi_b > 0$, and so $\Delta(\sigma_2, \sigma_1^B) \leq 0$ would be a contradiction for $R(w, \sigma) \geq \hat{R}(\sigma_2^B)$.

Thus, we have constructed $\sigma^* = \{\sigma_1^* = \sigma_1^C, \sigma_2^* = \sigma_2^C\}$ such that σ_1^*, σ_2^* correspond to theorem statement for the appropriate cases, respectively, and $R(w, \sigma^*) \geq R(w, \sigma)$, for all $\sigma = \{\sigma_1, \sigma_2\}$ where $\sigma_1, \sigma_2 \subseteq \mathbb{R}_+$ are open, measurable sets.

□

Theorem 2.4.1. *Let $R_1 < R_2$ be target earning rates during non-surged and surge states, respectively. There exist prices $w = \{w_1, w_2\}$ of the form*

$$w_i(\tau) = m_i\tau + z_i q_{i \rightarrow j}(\tau),$$

where $m_1, m_2, z_2 \geq 0$ (but z_1 may be either positive or negative), such that the optimal driver policy is to accept every trip in the surge state and all trips up to a certain length in the non-surge state. Furthermore, for $\frac{R_1}{R_2} \in [C, 1]$, there exist fully incentive compatible prices of this form, where

$$C = 1 - \frac{1}{T_1} \frac{Q_2(\lambda_{12}T_1 - Q_1) + Q_1(T_2\lambda_{1 \rightarrow 2} + Q_2)}{Q_2(\lambda_{1 \rightarrow 2}T_1 - Q_1) + \lambda_{1 \rightarrow 2}(T_2\lambda_{1 \rightarrow 2} + Q_2)} \in [0, 1),$$

and $T_i = \lambda_i F_i(\sigma_i) T_i((0, \infty))$, and $Q_i = Q_i((0, \infty))$.

Proof. Proof. Note that in the theorem statement we defined Q_i, T_i as what we call \bar{Q}_i, \bar{T}_i in the helper lemmas in Section A.4.3, i.e., they refer to their respective values when every trip is accepted.

Let $w_2(\tau) = m_2\tau + z_2 q_{2 \rightarrow 1}(\tau)$, and $w_1(\tau) = m_1\tau + z_1 q_{1 \rightarrow 2}(\tau)$.

From Lemmas A.4.5 and A.4.6 in Appendix Section A.4.3, the following constraints are sufficient for these prices to have always positive derivatives, with respect to upper endpoints u of the intervals that compose either σ_1 or σ_2 :

$$\begin{aligned} \frac{T_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) - (Q_1 + T_1\lambda_{2 \rightarrow 1})}{(Q_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1}))} &\leq \frac{z_2}{m_2 - R_1} \leq \frac{Q_2T_1 + Q_1}{Q_1(Q_2 - \lambda_{2 \rightarrow 1})} \\ m_1 &= R_2 \\ -\frac{(T_2\lambda_{1 \rightarrow 2} + Q_2)}{Q_2(\lambda_{1 \rightarrow 2}T_1 - Q_1) + \lambda_{1 \rightarrow 2}(T_2\lambda_{1 \rightarrow 2} + Q_2)} &\leq \frac{z_1}{R_2} \leq \frac{1}{(Q_1 - \lambda_{1 \rightarrow 2})} \end{aligned}$$

Now, applying Theorem A.4.1, the policy that accepts everything, $\sigma = \{(0, \infty), (0, \infty)\}$, is optimal, given these constraints are satisfied, as the derivative is always positive.

Resulting constraints on R_1, R_2 These constraints limit R_1, R_2 with respect to each other. From Remark A.4.2,

$$\begin{aligned} W_2 &= m_2(T_2 - 1) + z_2(Q_2 - \lambda_{2 \rightarrow 1}) \\ W_1 &= m_1(T_1 - 1) + z_1(Q_1 - \lambda_{1 \rightarrow 2}) \end{aligned}$$

Given R_2 , what's the range R_1 can be to still have IC in state 1?

$$\begin{aligned}
W_1 &\leq R_2 \left[T_1 - 1 + \left[\frac{1}{(Q_1 - \lambda_{1 \rightarrow 2})} (Q_1 - \lambda_{1 \rightarrow 2}) \right] \right] \\
\implies \frac{R_1}{R_2} &\leq 1 \\
W_1 &\geq R_2 \left[T_1 - 1 - \left[\frac{(T_2 \lambda_{1 \rightarrow 2} + Q_2)}{Q_2 (\lambda_{1 \rightarrow 2} T_1 - Q_1) + \lambda_{1 \rightarrow 2} (T_2 \lambda_{1 \rightarrow 2} + Q_2)} \right] (Q_1 - \lambda_{1 \rightarrow 2}) \right] \\
\implies \frac{R_1}{R_2} &= \frac{W_1}{T_1} \frac{1}{R_2} \\
&\geq \frac{1}{T_1} \left[T_1 - 1 - \left[\frac{(T_2 \lambda_{1 \rightarrow 2} + Q_2)}{Q_2 (\lambda_{1 \rightarrow 2} T_1 - Q_1) + \lambda_{1 \rightarrow 2} (T_2 \lambda_{1 \rightarrow 2} + Q_2)} \right] (Q_1 - \lambda_{1 \rightarrow 2}) \right] \\
&= 1 - \frac{1}{T_1} \left[1 + \frac{(T_2 \lambda_{1 \rightarrow 2} + Q_2)(Q_1 - \lambda_{1 \rightarrow 2})}{Q_2 (\lambda_{1 \rightarrow 2} T_1 - Q_1) + \lambda_{1 \rightarrow 2} (T_2 \lambda_{1 \rightarrow 2} + Q_2)} \right] \\
&= 1 - \frac{1}{T_1} \frac{Q_2 (\lambda_{1 \rightarrow 2} T_1 - Q_1) + \lambda_{1 \rightarrow 2} (T_2 \lambda_{1 \rightarrow 2} + Q_2) + (T_2 \lambda_{1 \rightarrow 2} + Q_2)(Q_1 - \lambda_{1 \rightarrow 2})}{Q_2 (\lambda_{1 \rightarrow 2} T_1 - Q_1) + \lambda_{1 \rightarrow 2} (T_2 \lambda_{1 \rightarrow 2} + Q_2)} \\
&= 1 - \frac{1}{T_1} \frac{Q_2 (\lambda_{1 \rightarrow 2} T_1 - Q_1) + Q_1 (T_2 \lambda_{1 \rightarrow 2} + Q_2)}{Q_2 (\lambda_{1 \rightarrow 2} T_1 - Q_1) + \lambda_{1 \rightarrow 2} (T_2 \lambda_{1 \rightarrow 2} + Q_2)} = C
\end{aligned}$$

What about incentive compatible pricing in state 2? If we only care about that state, can we can support any ratio of payments:

$$\begin{aligned}
\text{Let } z_2 &= \frac{Q_2 T_1 + Q_1}{Q_1 (Q_2 - \lambda_{2 \rightarrow 1})} (m_2 - R_1) \triangleq c(m_2 - R_1) \\
R_2 &= \frac{1}{T_2} [m_2 (T_2 - 1) + z_2 (Q_2 - \lambda_{2 \rightarrow 1})] \\
\implies \frac{R_2}{R_1} &= \frac{1}{R_1 T_2} [m_2 (T_2 - 1) + (m_2 - R_1) c (Q_2 - \lambda_{2 \rightarrow 1})] \\
&\rightarrow 1 - \frac{1}{T_2} \leq 1 \text{ as } m_2 \rightarrow R_1 \\
&\rightarrow \infty \text{ as } m_2 \rightarrow \infty
\end{aligned}$$

Thus, we can make the surge state IC for any ratio of payments $\frac{R_2}{R_1} \geq 1$, i.e., $\frac{R_1}{R_2} \leq 1$.

Now, suppose we want to achieve R_1, R_2 such that $\frac{R_1}{R_2} \in [0, C]$. From the previous line, we can still set w_2 such that every trip in state 2 is accepted (the derivative with respect to the surge policy is positive everywhere). Then, setting $z_1 = 0$, and m_1 to meet R_1 , all trips up to a certain length will be accepted in the non-surge state: By Remark A.4.1, $\frac{\partial}{\partial u} R(w, \sigma)$ is positive up to a certain value and then negative after that, where u is an upper endpoint of σ_1 . Thus, by Theorem A.4.1, the optimal policy is of the form $\sigma = \{(0, t_1), (0, \infty)\}$. \square

A.4.5 Optimal policies as depend on derivatives

Here we prove how the optimal policies depend on the derivative of the reward function (whether they are always positive, strictly increasing, strictly decreasing, strictly quasi-convex, or strictly quasi-concave). In each setting, we start with some fixed set σ' , and then make a sequence of changes to the policy that result in a set σ of the appropriate form. The set-up in each proof is the same; only the exact changes made to improve the policy differ. These changes depend on the structure of the derivative of the reward with respect to the endpoints of the sets that make up the policy, $\frac{\partial}{\partial u}R(w, \sigma)$. The idea is that as long as the derivative can be shown to be non-negative for some u that is an endpoint of σ_i , that policy can be locally modified to accept more trips while not decreasing the overall reward function.

Lemma A.4.1. *Consider a function $\hat{R}(\sigma)$ that maps open, measurable subsets $\sigma = \cup_k^\infty (\ell_k, u_k) \subseteq (0, \infty)$ to the non-negative reals, and probability measure F such that F is continuous, i.e. f is bounded.*

Let $\frac{\partial}{\partial u}\hat{R}(\sigma)$ denote the partial derivative of \hat{R} with respect to an upper end-point u_k of the intervals that make up $\sigma = \cup_k^\infty (\ell_k, u_k)$, i.e., it is the infinitesimal gain in the driver reward by adding u to the driver policy.

Consider an open measurable subset $\sigma' \subseteq (0, \infty)$. Suppose,

1. $F(\sigma') > 0$, and $\hat{R}(\sigma') > \hat{R}(\emptyset)$.
2. $\hat{R}(\sigma)$ is continuous in σ , and $\frac{\partial}{\partial u}\hat{R}(\sigma)$ exists, for all σ and its endpoints u_k .
3. $\frac{\partial}{\partial u}\hat{R}(\sigma)$ is continuous in u , for each fixed σ .
4. $\frac{\partial}{\partial u}\hat{R}(\sigma)$ is continuous in σ , for each fixed u .

Finally, suppose that there exists a function $r(u, \sigma)$ that has the same sign as $\frac{\partial}{\partial u}\hat{R}(\sigma)$, for all u, σ and has one of the following properties. Then, each of the following hold, depending on the properties of $r(u, \sigma)$.

- Suppose $r(u, \sigma)$ is **non-negative** for all u, σ . Then $\hat{R}((0, \infty)) \geq \hat{R}(\sigma')$.
- Suppose $\exists \epsilon > 0$ s.t. $r(u, \sigma)$ is **strictly increasing** in u (for a fixed σ), for all σ such that $\hat{R}(\sigma) \geq \hat{R}(\sigma') - \epsilon$. Then, there exists a value $\ell^* \in \mathbb{R}_+ \cup \{\infty\}$ such that $\hat{R}((\ell^*, \infty)) \geq \hat{R}(\sigma')$.
- Suppose $\exists \epsilon > 0$ s.t. $r(u, \sigma)$ is **strictly quasi-convex** in u (for a fixed σ), for all σ such that $\hat{R}(\sigma) \geq \hat{R}(\sigma') - \epsilon$. Then, there exists exist $\ell^*, u^* \in \mathbb{R}_+ \cup \{\infty\}$ such that $\hat{R}((0, \ell^*) \cup (u^*, \infty)) \geq \hat{R}(\sigma')$, and it is not the case that both $\ell^* = 0, u^* = \infty$.
- Suppose $\exists \epsilon > 0$ s.t. $r(u, \sigma)$ is **strictly decreasing** in u (for a fixed σ), for all σ such that $\hat{R}(\sigma) \geq \hat{R}(\sigma') - \epsilon$. Then, there exists a value $u^* \in \mathbb{R}_+ \cup \{\infty\}$ such that $\hat{R}((0, u^*)) \geq \hat{R}(\sigma')$.

- Suppose $\exists \epsilon > 0$ s.t. $r(u, \sigma)$ is **strictly quasi-concave** in u (for a fixed σ), for all σ such that $\hat{R}(\sigma) \geq \hat{R}(\sigma') - \epsilon$. Then, there exists exist $\ell^*, u^* \in \mathbb{R}_+ \cup \{\infty\}$ such that $\hat{R}((\ell^*, u^*)) \geq \hat{R}(\sigma')$.

Proof. Proof. The general approach is as follows: Start at subset $\sigma' \subseteq (0, \infty) = \cup_k^\infty (\ell_k, u_k) = \cup_k^\infty \zeta_k$, where the intervals are disjoint and $\zeta_k = (\ell_k, u_k)$ denotes the k th interval. (recall that any open subset of \mathbb{R} can be uniquely written as the countable union of such disjoint intervals).

Then, do the following:

1. Create a sequence $\sigma'_\delta \rightarrow \sigma'$ (as $\delta \rightarrow 0$), where, for each δ , σ'_δ is δ -close to σ' : $F((\sigma' \setminus \sigma'_\delta) \cup (\sigma'_\delta \setminus \sigma')) < \delta$.
2. Show that there exists a σ^* of the appropriate form (according to the property that holds above), such that $\hat{R}(\sigma'_\delta) \leq R(\sigma^*), \forall \delta$.

By continuity of the set function \hat{R} , this implies that $\hat{R}(\sigma') \leq R(\sigma^*)$.

The second step is the only one that differs substantially depending on the property of the function $r(u, \sigma)$.

Step one: a sequence $\sigma'_\delta \rightarrow \sigma'$ Each σ'_δ will be of the form $\sigma'_\delta = (0, L) \cup (\cup_{k=1}^K (\ell_k, u_k)) \cup (B, \infty)$, for some K, B, L that depend on δ . We construct a σ'_δ such that $F(\sigma' \setminus \sigma'_\delta \cup \sigma'_\delta \setminus \sigma') < \delta$ as follows:

- F is a finite (probability) measure, and so there exists K such that $F(\cup_{k=K+1}^\infty (\ell_k, u_k)) < \delta/2$. (Since $F(\sigma') \leq 1$, it follows by the Cauchy condition).
- Let $B \in \mathbb{R}$ s.t. $F((B, \infty)) < \delta/4$. Let $L \in \mathbb{R}$ s.t. $F((0, L)) < \delta/4$. Such B, L exist by condition on F .
- Set $\sigma'_\delta = (0, L) \cup (\cup_{k=1}^K (\ell_k, u_k)) \cup (B, \infty)$.
- For convenience, we re-index the disjoint intervals $\{\zeta_k\}_{k=1}^{K+2}$ such that they are in increasing order, i.e. $u_k > \ell_k \geq u_{k-1}, \forall k > 1$, starting at $(0, L)$, with the last interval (B, ∞) . If there exist any intervals such that $\ell_k = u_{k-1}$, replace them with the combined interval (ℓ_{k-1}, u_k) . If $\{B, \infty\}$ overlaps with the last interval, combine them.

Step 2: showing that $\hat{R}(\sigma^*) \geq \hat{R}(\sigma')$, where $\sigma^* = (0, \infty)$ Now, starting at $\sigma = \sigma'_\delta$, we describe a sequence of modifications to σ , such that each modification does not reduce the reward $\hat{R}(\sigma)$. The limit of this sequence of modifications is a policy of the appropriate form, regardless of the starting σ'_δ . This shows that $\hat{R}(\sigma^*) \geq \hat{R}(\sigma'_\delta)$.

We now carry out this step separately for each case. The general argument is that the properties force the derivative $r(u, \sigma)$ to be positive at certain points, which allows expanding the policy until a policy of the appropriate form is reached.

Setting where $r(u, \sigma)$ is non-negative Let $\sigma = \cup_{k=1}^K (\ell_k, u_k)$, and note that $\ell_1 = 0$ from above. By supposition that $\frac{\partial}{\partial u} \hat{R}(\sigma)$ is always non-negative in u , we can increase u_1 (merging with other intervals) without decreasing $\hat{R}(\sigma)$. Thus, we can keep increasing u_1 , and $u_1 \rightarrow B$, and so $R((0, \infty)) \geq R(\sigma'_\delta)$.

Setting where $r(u, \sigma)$ is strictly increasing By suppositions, $\exists \delta_0$ small enough such that $r(u, \sigma)$ is strictly increasing for all σ such that $\hat{R}\sigma \geq \hat{R}(\sigma'_\delta)$, $\forall \delta < \delta_0$. Suppose $\delta < \delta_0$.

Now, starting at $\sigma = \sigma'_\delta$, the limit of the sequence of modifications is a policy $\sigma^* = (\ell^*, \infty)$.

We continue to overload notation, with $r(\ell, \sigma)$ indicating a function that has the same sign as the derivative of a lower endpoint of σ_i .

By the supposition that $r(u, \sigma)$ strictly increasing in u , we have:

$$\begin{aligned}
& r(\ell, \sigma) \text{ strictly decreasing} \\
& r(\ell_1, \sigma) \leq 0 \implies r(u_1, \sigma) > 0 \qquad \ell_1 < u_1 \\
& \equiv \frac{\partial}{\partial \ell_1} \hat{R}(\sigma) \leq 0 \implies \frac{\partial}{\partial u_1} \hat{R}(\sigma) > 0 \qquad xf(x) \geq 0 \\
& r(\ell_1, \sigma) > 0 \iff r(u_1, \sigma) \leq 0 \\
& \equiv \frac{\partial}{\partial \ell_1} \hat{R}(\sigma) > 0 \iff \frac{\partial}{\partial u_1} \hat{R}(\sigma) \leq 0
\end{aligned}$$

Case 1: $\exists \zeta_1, \zeta_2 \subset \sigma$ such that $\ell_2 > u_1, |\zeta_1|, |\zeta_2|$, i.e. there is more than one interval that makes up σ , and ζ_1, ζ_2 are the first and second such intervals, respectively, with positive mass.

Then we make the following sequence of changes (forming new σ), depending on $\frac{\partial}{\partial \ell_1} \hat{R}(\sigma), \frac{\partial}{\partial u_1} \hat{R}(\sigma)$:

Subcase 1A, $\frac{\partial}{\partial u_1} \hat{R}(\sigma) > 0$: Increase u_1 until $u_1 = \ell_2$ (exit Case 1), or $\frac{\partial}{\partial u_1} \hat{R}(\sigma) \leq 0$ (go to Case 1B).

Sub-subcase 1AA, $\frac{\partial}{\partial \ell_1} \hat{R}(\sigma) < 0, \ell_1 > 0$: Simultaneously, decrease ℓ_1 .

Sub-subcase 1AB, $\frac{\partial}{\partial \ell_1} \hat{R}(\sigma) \geq 0$ or $\ell_1 = 0$: Hold ℓ_1 fixed.

Subcase 1B, $\frac{\partial}{\partial u_1} \hat{R}(\sigma) \leq 0 \implies \frac{\partial}{\partial \ell_1} \hat{R}(\sigma) > 0$: Increase ℓ_1 until $\ell_1 = u_1$ (exit Case 1), or $\frac{\partial}{\partial \ell_1} \hat{R}(\sigma) \leq 0$ (which implies $\frac{\partial}{\partial u_1} \hat{R}(\sigma) > 0$, i.e. go to Case 1A).

Each of these changes cannot decrease $\hat{R}(\sigma)$, due to the direction of the changes in u_1, ℓ_1 and the corresponding derivatives (and thus the scaled gradient remains strictly increasing by supposition). Note that these subcases are mutually-exclusive, and one is true as long as $\exists \zeta_1, \zeta_2 \subset \sigma, \ell_2 > u_1$. Further, note that u_1 is increasing in Subcase 1A and constant in Subcase 1B. Thus, with ℓ_2 fixed and bounded, eventually:

- $\ell_1 \rightarrow u_1$, in Subcase 1B (i.e. the first interval collapses to mass 0). OR
- $u_1 \rightarrow \ell_2$, in Subcase 1A (i.e. the first interval merges with the second).

Thus, this sequence of changes cannot decrease the reward, and results in there being one fewer interval than before (after combining the bottom 2 intervals by adding the point $u_1 = \ell_2$ of 0 measure). Case 1 can be iteratively applied until there is just a single interval $\sigma = (\ell', \infty)$.

Case 2: $\sigma = (\ell', \infty)$, i.e. there is a single interval that makes up σ

By supposition, $\hat{R}(\sigma') > \hat{R}(\emptyset)$ and so $\hat{R}((\ell', \infty)) > \hat{R}(\emptyset)$. Further $\hat{R}((\ell, \infty))$ is a continuous function in ℓ . Thus, there exists L such that $\forall \ell > L, \hat{R}((\ell', \infty)) > \hat{R}((\ell, \infty))$.

Thus, there exists $\ell^* \in [0, L]$ such that $\hat{R}((\ell^*, \infty)) \geq \hat{R}((\ell, \infty)), \forall \ell \in \mathbb{R}_+ \cup \{\infty\}$ (continuous functions in a compact domain have a maximum).

Setting where $r(u, \sigma)$ is strictly quasi-convex We show that there exists a $\sigma^* = (0, \ell^*) \cup (u^*, \infty)$, for some $u^*, \ell^* \in \mathbb{R}_+$, such that $\hat{R}(\sigma'_\delta) \leq R(\sigma^*), \forall \delta$.

By suppositions, $\exists \delta_0$ small enough such that $r(u, \sigma)$ is strictly quasi-convex for all σ such that $\hat{R}\sigma \geq \hat{R}(\sigma'_\delta), \forall \delta < \delta_0$. Suppose $\delta < \delta_0$.

Now, starting at $\sigma = \sigma'_\delta$, we describe a sequence of modifications to σ , such that each modification does not reduce the reward $\hat{R}(\sigma)$. The limit of this sequence of modifications is the policy $\sigma^* = (0, \ell^*) \cup (u^*, \infty)$, regardless of the starting σ'_δ . This shows that $\hat{R}(\sigma^*) \geq \hat{R}(\sigma'_\delta)$.

Then, let $\sigma'_\delta = \cup_{k=1}^K (\ell_k, u_k)$, where $\zeta_k \triangleq (\ell_k, u_k)$.

The key step is noting that quasi-convexity of the transformed derivative implies that any σ with three intervals $\zeta_1, \zeta_2, \zeta_3$ can be improved by eliminating the middle interval (or joining it with one of the others).

Case 1: \exists disjoint $\zeta_1 = (0, u_1), \zeta_2 = (\ell_2, u_2), \zeta_3 = (\ell_3, u_3)$, s.t. $|\zeta_1|, |\zeta_2|, |\zeta_3| > 0$, i.e. σ is composed of at least three intervals, and $\zeta_1, \zeta_2, \zeta_3$ are the first three such intervals with positive mass. (u_3 may be ∞).

By supposition, the transformed derivative with respect to any of the upper end-points u_k , $r(u_k, \sigma)$, is strictly quasi-convex in u . Then, the transformed derivative with respect to any of the lower end-points ℓ_k , $r(\ell_k, \sigma)$, is strictly quasi-concave in u , and further is the negative of $r(u, \sigma)$ when $u = \ell_k$.

Then, we have:

$$\begin{aligned} \frac{\partial}{\partial u_1} \hat{R}(\sigma) \leq 0 \text{ and } \frac{\partial}{\partial \ell_3} \hat{R}(\sigma) \geq 0 &\implies \frac{\partial}{\partial \ell_2} \hat{R}(\sigma) > 0 \text{ and } \frac{\partial}{\partial u_2} \hat{R}(\sigma) < 0 \\ \frac{\partial}{\partial \ell_2} \hat{R}(\sigma) \leq 0 \text{ or } \frac{\partial}{\partial u_2} \hat{R}(\sigma) \geq 0 &\implies \frac{\partial}{\partial u_1} \hat{R}(\sigma) > 0 \text{ or } \frac{\partial}{\partial \ell_3} \hat{R}(\sigma) < 0 \end{aligned}$$

Then we make the following sequence of changes (forming new σ):

Subcase 1A, $\frac{\partial}{\partial u_1} \hat{R}(\sigma) \leq 0$ and $\frac{\partial}{\partial \ell_3} \hat{R}(\sigma) \geq 0 \implies \frac{\partial}{\partial \ell_2} \hat{R}(\sigma) > 0$ and $\frac{\partial}{\partial u_2} \hat{R}(\sigma) < 0$: Increase ℓ_2 and decrease u_2 simultaneously until $\ell_2 = u_2$ (exit Case 1), $\frac{\partial}{\partial u_2} \hat{R}(\sigma) \geq 0$, or $\frac{\partial}{\partial \ell_2} \hat{R}(\sigma) \leq 0$ (go to **1B** or **1C**).

Subcase 1B, $\frac{\partial}{\partial u_1} \hat{R}(\sigma) > 0$: Increase u_1 until $u_1 = \ell_2$ (exit Case 1), or $\frac{\partial}{\partial u_1} \hat{R}(\sigma) \leq 0$ (go to **1A** or **1C**)

Subcase 1C, $\frac{\partial}{\partial \ell_3} \hat{R}(\sigma) < 0$: Decrease ℓ_3 until $u_2 = \ell_3$ (exit Case 1), or $\frac{\partial}{\partial \ell_3} \hat{R}(\sigma) \geq 0$ (go to **1B** or **1A**).

Each of these changes strictly increase $\hat{R}(\sigma)$. **1B** and **1C** may both be true, in which case arbitrarily decide between them. At least one of the three subcases is true as long as the Case 1 condition holds. Thus, eventually:

- $\ell_2 = u_2$, in Subcase 1A (i.e. the middle interval collapses to mass 0). OR
- $u_1 = \ell_2$, in Subcase 1B (i.e. the first interval merges with the second). OR
- $u_2 = \ell_3$, in Subcase 1C (i.e. the third interval merges with the second).

Thus, this sequence of changes cannot decrease the reward, and result in there being one fewer interval than before. Case 1 can be iteratively applied until there are just two intervals $\sigma = (0, t_1) \cup (t_2, \infty)$.

Case 2: $\sigma = (0, t_1) \cup (t_2, \infty)$.

By supposition, $\hat{R}(\sigma') > \hat{R}(\emptyset)$ and so $\hat{R}((0, t_1) \cup (t_2, \infty)) > \hat{R}(\emptyset)$ for $t_1 > 0$ or $t_2 < \infty$.

Further $\hat{R}((0, t_1) \cup (t_2, \infty))$ is a continuous function in t_1, t_2 . Thus $\hat{R}((0, t_1) \cup (t_2, \infty)) \rightarrow \hat{R}(\emptyset)$ as $t_1 \rightarrow 0, t_2 \rightarrow \infty$ together.

Further, $\hat{R}((0, t_1) \cup (t_2, \infty)) \rightarrow \hat{R}((0, \infty))$ as $t_1 \rightarrow \infty$, regardless of how t_2 behaves. Similarly, fixing t_1 , $\hat{R}((0, t_1) \cup (t_2, \infty)) \rightarrow \hat{R}((0, t_1))$ as $t_2 \rightarrow \infty$.

- If $\hat{R}((0, t_1) \cup (t_2^*(t_1), \infty))$ is increasing for $t_1 > T_1$, for however $t_2^*(t_1)$ behaves as a function of t_1 then $\hat{R}((0, \infty)) \geq \hat{R}((0, t_1) \cup (t_2, \infty)), \forall t_1 > T_1, t_2$.
- For any fixed t_1 , if $\hat{R}((0, t_1) \cup (t_2, \infty))$ is increasing for $t_2 > T_2$, then $\hat{R}((0, t_1)) \geq \hat{R}((0, t_1) \cup (t_2, \infty)), \forall t_2$.

These limiting values eliminate the possible cases where t_1 or t_2 increasing to infinity, but the asymptotic values at ∞ produce lower rewards, which would have implied that the maximum is not achieved. Thus, either

1. $\exists t_1^* \in (0, \infty) : \hat{R}((0, t_1^*)) \geq \hat{R}((0, t_1) \cup (t_2, \infty)), \forall t_1, t_2$
 2. $\exists t_1^*, t_2^* \in [0, \infty) : \hat{R}((0, t_1^*) \cup (t_2^*, \infty)) \geq \hat{R}((0, t_1) \cup (t_2, \infty)), \forall t_1, t_2$
-

Setting where $r(u, \sigma)$ is strictly decreasing The proof is extremely similar to the strictly increasing case. However, we now need to modify the starting σ' so it *does not* contain an interval (B, ∞) , and each case from above is duplicated but moves the policy in different directions.

When creating σ'_δ from above, instead *remove* the interval B : set $\sigma'_\delta = (0, L) \cup (\cup_{k=1}^K (\ell_k, u_k)) \setminus (B, \infty)$.

Showing that $\exists u^*$ such that $\exists \delta_0 : \forall \delta < \delta_0, \hat{R}(\sigma'_\delta) \leq \hat{R}((0, u^*))$ By suppositions, $\exists \delta_0$ small enough such that $r(u, \sigma)$ is strictly decreasing for all σ such that $\hat{R}\sigma \geq \hat{R}(\sigma'_\delta), \forall \delta < \delta_0$. Suppose $\delta < \delta_0$.

Now, starting at $\sigma = \sigma'_\delta$, we describe a sequence of modifications to σ , such that each modification does not reduce the reward $\hat{R}(\sigma)$. The limit of this sequence of modifications is a policy $\sigma^* = (0, u^*)$, regardless of the starting σ'_δ . This shows that $\hat{R}(\sigma^*) \geq \hat{R}(\sigma'_\delta)$.

We continue to overload notation, with $r(\ell, \sigma)$ indicating a function that has the same sign as the derivative of a lower endpoint of σ_i . By the supposition that $r(u, \sigma)$ strictly decreasing in u , we have:

$$\begin{aligned}
& r(\ell, \sigma) \text{ strictly increasing} \\
& r(u_K, \sigma) \geq 0 \implies r(\ell_K, \sigma) < 0 \qquad \ell_1 < u_1 \\
& \equiv \frac{\partial}{\partial u_K} \hat{R}(\sigma) \geq 0 \implies \frac{\partial}{\partial \ell_K} \hat{R}(\sigma) < 0 \qquad xf(x) \geq 0 \\
& r(u_K, \sigma) < 0 \iff r(\ell_K, \sigma) \geq 0 \\
& \equiv \frac{\partial}{\partial u_K} \hat{R}(\sigma) < 0 \iff \frac{\partial}{\partial \ell_K} \hat{R}(\sigma) \geq 0
\end{aligned}$$

Case 1: $\exists \zeta_{K-1}, \zeta_K \subset \sigma$ such that $\ell_K > u_{K-1}, |\zeta_K|, |\zeta_{K-1}|$, i.e. there is more than one interval that makes up σ , and ζ_K, ζ_{K-1} are the last two such intervals, respectively, with positive mass.

Then we make the following sequence of changes (forming new σ), depending on $\frac{\partial}{\partial \ell_K} \hat{R}(\sigma), \frac{\partial}{\partial u_K} \hat{R}(\sigma)$:

Subcase 1A, $\frac{\partial}{\partial \ell_K} \hat{R}(\sigma) < 0$: Decreasing ℓ_K until $\ell_K = u_{K-1}$ (exit Case 1), or $\frac{\partial}{\partial \ell_K} \hat{R}(\sigma) \geq 0$ (go to Case 1B).

Sub-subcase 1AA, $\frac{\partial}{\partial u_K} \hat{R}(\sigma) > 0$: Simultaneously, increase u_K .

Sub-subcase 1AB, $\frac{\partial}{\partial u_K} \hat{R}(\sigma) \leq 0$: Hold u_K fixed.

Subcase 1B, $\frac{\partial}{\partial \ell_K} \hat{R}(\sigma) \geq 0 \implies \frac{\partial}{\partial u_K} \hat{R}(\sigma) < 0$: Decrease u_K until $u_K = \ell_K$ (exit Case 1), or $\frac{\partial}{\partial u_K} \hat{R}(\sigma) \geq 0$ (which implies $\frac{\partial}{\partial \ell_K} \hat{R}(\sigma) < 0$, i.e. go to Case 1A).

Each of these changes cannot decrease $\hat{R}(\sigma)$, due to the direction of the changes in ℓ_K, u_K and the corresponding derivatives (and thus the scaled gradient remains strictly increasing by supposition). Note that these subcases are mutually-exclusive, and one is true as long as \exists such $\zeta_K, \zeta_{K-1} \subset \sigma$. Further, note that ℓ_K is decreasing in Subcase 1A and constant in Subcase 1B. Thus, eventually:

- $u_K \rightarrow \ell_K$, in Subcase 1B (i.e. the last interval collapses to mass 0). OR
- $\ell_K \rightarrow u_{K-1}$, in Subcase 1A (i.e. the last interval merges with the second to last).

Thus, this sequence of changes cannot decrease the reward, and results in there being one fewer interval than before Case 1 can be iteratively applied until there is just a single interval $\sigma = (0, u')$.

Case 2: $\sigma = (0, u')$, i.e. there is a single interval that makes up σ By supposition, $\hat{R}((0, u))$ is a continuous function for $u \in [0, \infty) \cup \{\infty\}$. Further, $\frac{\partial}{\partial u} \hat{R}((0, u))$ is strictly decreasing for all u such that $\hat{R}((0, u)) \geq \hat{R}((0, u'))$. If $\frac{\partial}{\partial u} \hat{R}((0, u)) > 0, \forall u$, then $u^* = \infty$ is optimal. Otherwise if $\frac{\partial}{\partial u} \hat{R}((0, u)) < 0 \forall u$, then u^* is optimal. Otherwise $\exists u^* \in (0, \infty)$ such that $\hat{R}(\sigma^*) \geq \hat{R}(\sigma')$.

Setting where $r(u, \sigma)$ is strictly quasi-concave The proof is extremely similar to the strictly quasi-convex case. However, we now need to modify the starting σ' so it *does not* contain an intervals $(0, L)$ or (B, ∞) .

Let $\sigma'_\delta = (\cup_{k=1}^K (\ell_k, u_k)) \setminus (0, L) \setminus (B, \infty)$. Now, starting at $\sigma = \sigma'_\delta$, we describe a sequence of modifications to σ , such that each modification does not reduce the reward $\hat{R}(\sigma)$. The limit of this sequence of modifications is the policy $\sigma^* = (\ell^*, u^*)$, regardless of the starting σ'_δ .

Similar to before, the key step is noting that quasi-concavity of the transformed derivative implies that any σ with two intervals ζ_1, ζ_2 can be improved by eliminating one (or joining the two).

Case 1: \exists **disjoint** $\zeta_1 = (\ell_1, u_1), \zeta_2 = (\ell_2, u_2)$, **s.t.** $|\zeta_1|, |\zeta_2| > 0$, i.e. σ is composed of at least two intervals with positive mass, and ζ_1, ζ_2 are the first two such intervals.

By supposition, the transformed derivative with respect to any of the upper end-points u_k , $r(u_k, \sigma)$, is strictly quasi-concave in u . Then, the transformed derivative with respect to any of the lower end-points ℓ_k , $r(\ell_k, \sigma)$, is strictly quasi-convex in u , and further is the negative of $r(u, \sigma)$ when $u = \ell_k$.

Then, we have:

$$\begin{aligned} \frac{\partial}{\partial \ell_1} \hat{R}(\sigma) \leq 0 \text{ and } \frac{\partial}{\partial \ell_2} \hat{R}(\sigma) \leq 0 &\implies \frac{\partial}{\partial u_1} \hat{R}(\sigma) > 0 \\ \frac{\partial}{\partial u_1} \hat{R}(\sigma) \leq 0 &\implies \frac{\partial}{\partial \ell_1} \hat{R}(\sigma) > 0 \text{ or } \frac{\partial}{\partial \ell_2} \hat{R}(\sigma) > 0 \end{aligned}$$

Then we make the following sequence of changes (forming new σ):

Subcase 1A, $\frac{\partial}{\partial \ell_1} \hat{R}(\sigma) \leq 0$ and $\frac{\partial}{\partial \ell_2} \hat{R}(\sigma) \leq 0 \implies \frac{\partial}{\partial u_1} \hat{R}(\sigma) > 0$: Increase u_1 until $u_1 = \ell_2$ (exit Case 1) or $\frac{\partial}{\partial u_1} \hat{R}(\sigma) \leq 0$ (go to **1B** or **1C**).

Subcase 1B, $\frac{\partial}{\partial \ell_1} \hat{R}(\sigma) > 0$: Increase ℓ_1 until $u_1 = \ell_1$ (exit Case 1), or $\frac{\partial}{\partial \ell_1} \hat{R}(\sigma) \leq 0$ (go to **1A** or **1C**)

Subcase 1C, $\frac{\partial}{\partial \ell_2} \hat{R}(\sigma) > 0$: Increase ℓ_2 until $u_2 = \ell_2$ (exit Case 1), or $\frac{\partial}{\partial \ell_2} \hat{R}(\sigma) \leq 0$ (go to **1B** or **1A**).

Each of these changes strictly increase $R(\sigma)$. **1B** and **1C** may both be true, in which case arbitrarily decide between them. At least one of the three subcases is true as long as the Case 1 condition holds. Thus, eventually:

- $\ell_2 = u_1$, in Subcase 1A (i.e. the intervals combine). OR
- $u_1 = \ell_1$, in Subcase 1B (i.e. the first interval collapses to mass 0). OR
- $u_2 = \ell_2$, in Subcase 1C (i.e. the second interval collapses to mass 0).

Thus, this sequence of changes cannot decrease the reward, and result in there being one fewer interval than before. Case 1 can be iteratively applied until there is just one interval $\sigma_i = (t_1, t_2)$.

Case 2: $\sigma_i = (t_1, t_2)$. Similar to the same case in the previous theorem.

Further $\hat{R}((t_1, t_2))$ is a continuous function in t_1, t_2 . Thus $\hat{R}((t_1, t_2)) \rightarrow \hat{R}(\emptyset)$ as $t_1 \rightarrow t_2$.

Further, $\hat{R}((t_1, t_2)) \rightarrow \hat{R}(\emptyset)$ as $t_1 \rightarrow \infty$, regardless of how $t_2 \geq t_1$ behaves. Similarly, fixing t_1 , $\hat{R}((t_1, t_2)) \rightarrow \hat{R}((t_1, \infty))$ as $t_2 \rightarrow \infty$.

- If $\hat{R}((t_1, t_2^*(t_1)))$ is increasing for $t_1 > T_1$, for however $t_2^*(t_1)$ behaves as a function of t_1 then $\hat{R}(\emptyset) \geq \hat{R}((t_1, t_2)), \forall t_1 > T_1, t_2$.
- For any fixed t_1 , if $\hat{R}((t_1, t_2))$ is increasing for $t_2 > T_2$, then $\hat{R}((t_1, \infty)) \geq \hat{R}((t_1, t_2)), \forall t_2 > T_2$.

These limiting values eliminate the possible cases where t_1 or t_2 increasing to infinity, but the asymptotic values at ∞ produce lower rewards, which would have implied that the maximum is not achieved. Thus, either

1. $\hat{R}(\emptyset) \geq \hat{R}((t_1, t_2)), \forall t_1, t_2$
2. $\exists t_1^* \in [0, \infty) : \hat{R}((t_1^*, \infty)) \geq \hat{R}((t_1, t_2)), \forall t_1, t_2$
3. $\exists t_1^*, t_2^* \in [0, \infty) : \hat{R}(t_1^*, t_2^*) \geq \hat{R}((t_1, t_2)), \forall t_1, t_2$

□

A.4.6 Proofs of appendix-only lemmas

Remark A.4.2.

$$\text{Let } w_i(u) = mu + zq_{i \rightarrow j}(u)$$

$$\text{Then } W_i = m(T_i - 1) + z(Q_i - \lambda_{i \rightarrow j})$$

$$\begin{aligned} \frac{\partial}{\partial u} R(w, \sigma) \propto q_{i \rightarrow j}(u) [(R_j - m)T_j T_i + mT_j + zQ_j T_i + zT_j \lambda_{i \rightarrow j}] \\ + u [Q_i T_j (m - R_j) + Q_j (m - zQ_i + z\lambda_{i \rightarrow j})] \end{aligned}$$

Proof. Proof.

$$w_i(u) = mu + zq_{i \rightarrow j}(u)$$

$$m, z \geq 0$$

$$W_i = \lambda_i \int_{\tau \in \sigma_i} w_i(\tau) dF_i(\tau) = \lambda_i \int_{\tau \in \sigma_i} [m\tau + zq_{i \rightarrow j}(\tau)] dF_i(\tau) = m(T_i - 1) + z(Q_i - \lambda_{i \rightarrow j})$$

Then

$$\begin{aligned} W_j T_i - T_j W_i &= R_j T_j T_i - m T_j (T_i - 1) - z T_j (Q_i - \lambda_{i \rightarrow j}) \\ w_i(u)(Q_i T_j + Q_j T_i) &= (mu + zq_{i \rightarrow j}(u))(Q_i T_j + Q_j T_i) \\ &= q_{i \rightarrow j}(u)(zQ_i T_j + zQ_j T_i) + u(mQ_i T_j + mQ_j T_i) \\ \frac{\partial}{\partial u} R(w, \sigma) \propto q_{i \rightarrow j}(u) [W_j T_i - T_j W_i] &+ w_i(u)(Q_i T_j + Q_j T_i) - u(Q_i W_j + Q_j W_i) \\ &= q_{i \rightarrow j}(u) [R_j T_j T_i - m T_j (T_i - 1) - z T_j (Q_i - \lambda_{i \rightarrow j}) + z Q_i T_j + z Q_j T_i] \\ &\quad + u [m Q_i T_j + m Q_j T_i - Q_i R_j T_j - Q_j (m(T_i - 1) + z(Q_i - \lambda_{i \rightarrow j}))] \\ &= q_{i \rightarrow j}(u) [(R_j - m) T_j T_i + m T_j + z Q_j T_i + z T_j \lambda_{i \rightarrow j}] \\ &\quad + u [Q_i T_j (m - R_j) + Q_j (m - z Q_i + z \lambda_{i \rightarrow j})] \end{aligned}$$

□

Remark A.4.4. $\lambda_{i \rightarrow j} T_i - Q_i \geq 0$ and maximized when $\sigma_i = (0, \infty)$. Similarly, $Q_i \geq 0$ and maximized when $\sigma_i = (0, \infty)$.

Proof. Proof.

$$\begin{aligned} \lambda_{i \rightarrow j} T_i - Q_i &= \lambda_{i \rightarrow j} \left[1 + \lambda_i \int_{\tau \in \sigma_i} \tau dF_i(\tau) \right] - \lambda_{i \rightarrow j} - \lambda_i \int_{\sigma_i} q_{i \rightarrow j}(\tau) dF_i(\tau) \\ &= \lambda_i \int_{\tau \in \sigma_i} [\lambda_{i \rightarrow j} \tau - q_{i \rightarrow j}(\tau)] dF_i(\tau) \end{aligned}$$

$\lambda_{i \rightarrow j} \tau - q_{i \rightarrow j}(\tau)$ is increasing in τ :

$$\frac{\partial}{\partial \tau} [\lambda_{i \rightarrow j} \tau - q_{i \rightarrow j}(\tau)] = \lambda_{i \rightarrow j} - [\lambda_{i \rightarrow j} e^{-(\lambda_{i \rightarrow j} + \lambda_{j \rightarrow i})\tau}] \geq 0$$

and $\lambda_{i \rightarrow j} * 0 - q_{i \rightarrow j}(0) = 0$. Thus, the function being integrated is positive, and so $\lambda_{i \rightarrow j} T_i - Q_i > 0$ and maximized when $\sigma_i = (0, \infty)$. Identical proof holds for Q_i .

□

Lemma A.4.2. *Let*

$$r(u, i, w, \sigma) \triangleq \frac{q_{i \rightarrow j}(u)}{u} \Delta_{ji} + \frac{w_i(u)}{u} \left(\frac{Q_i}{T_i} + \frac{Q_j}{T_j} \right) - \left(\frac{Q_i}{T_i} R_j + \frac{Q_j}{T_j} R_i \right)$$

Then, $\frac{\partial}{\partial u} R(w, \sigma) \propto r(u, i, w, \sigma)$.

Proof. Proof.

$$\begin{aligned} \mu_i(\{\sigma_j, \sigma_2\}) &= \frac{Q_j T_i}{Q_j T_i + Q_i T_j} \\ R(w, \sigma) &= \mu_1(\sigma) R_1(w_1, \sigma_1) + \mu_2(\sigma) R_2(w_2, \sigma_2) \\ &= \left[\frac{1}{Q_2 T_1 + Q_1 T_2} \right] [Q_2 W_1 + Q_1 W_2] \\ R_i(\sigma_i) &= \frac{W_i}{T_i} \\ \frac{\partial}{\partial u} Q_i &= \frac{\partial}{\partial u} \left[\lambda_{i \rightarrow j} + \lambda_i \int_{\tau \in \sigma_i} q_{i \rightarrow j}(\tau) dF_i(\tau) \right] = \lambda_i q_{i \rightarrow j}(u) f_i(u) \\ \frac{\partial}{\partial u} W_i &= \frac{\partial}{\partial u} \left[\lambda_i \int_{\tau \in \sigma_i} w_i(\tau) dF_i(\tau) \right] = \lambda_i w_i(u) f_i(u) \\ \frac{\partial}{\partial u} T_i &= \lambda_i f_i(u) u \end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial u} R(w, \sigma) &= \left[\frac{\lambda_i f_i(u)}{Q_i T_j + Q_j T_i} \right] [[q_{i \rightarrow j}(u) W_j + Q_j w_i(u)] - R(w, \sigma)(u Q_j + q_{i \rightarrow j}(u) T_j)] \\
&\propto [q_{i \rightarrow j}(u) W_j + Q_j w_i(u)] - R(w, \sigma)(u Q_j + q_{i \rightarrow j}(u) T_j) \\
&\propto [q_{i \rightarrow j}(u) W_j + Q_j w_i(u)] (Q_i T_j \\
&\quad + Q_j T_i) - (Q_i W_j + Q_j W_i)(u Q_j + q_{i \rightarrow j}(u) T_j) \\
&= q_{i \rightarrow j}(u) W_j (Q_i T_j + Q_j T_i) + Q_j w_i(u) (Q_i T_j + Q_j T_i) \\
&\quad - u Q_j (Q_i W_j + Q_j W_i) - q_{i \rightarrow j}(u) T_j (Q_i W_j + Q_j W_i) \\
&\propto q_{i \rightarrow j}(u) W_j T_i + w_i(u) (Q_i T_j + Q_j T_i) - u (Q_i W_j + Q_j W_i) - q_{i \rightarrow j}(u) T_j W_i \\
&= q_{i \rightarrow j}(u) [W_j T_i - T_j W_i] + w_i(u) (Q_i T_j + Q_j T_i) - u (Q_i W_j + Q_j W_i) \\
&= u T_i T_j \left[\frac{q_{i \rightarrow j}(u)}{u} (R_j - R_i) + \frac{w_i(u)}{u} \left(\frac{Q_i}{T_i} + \frac{Q_j}{T_j} \right) - \left(\frac{Q_i}{T_i} R_j + \frac{Q_j}{T_j} R_i \right) \right] \\
&\propto \frac{q_{i \rightarrow j}(u)}{u} \Delta_{ji} + \frac{w_i(u)}{u} \left(\frac{Q_i}{T_i} + \frac{Q_j}{T_j} \right) - \left(\frac{Q_i}{T_i} R_j + \frac{Q_j}{T_j} R_i \right) \quad \Delta_{ji} = R_j - R_i \\
&\triangleq r(u, i, w, \sigma)
\end{aligned}$$

□

Lemma A.4.3. *Suppose $w_i(\tau) = m\tau + a$, where $m, a > 0$. Then, $r(u, i, w, \sigma)$ is strictly quasi-convex in u , for each fixed σ where $\Delta_{ji} < 0$.*

Proof. Proof.

$$r(u, i, w, \sigma) = \frac{c_1 a - c_2 q_{i \rightarrow j}(u)}{u} + c_3 \quad c_1, c_2 \geq 0; c_3 \text{ can be negative}$$

Then, $\frac{\partial}{\partial u} r(u, i, w, \sigma)$

$$\begin{aligned}
&= \frac{\partial}{\partial u} \left[\frac{c_1 - c_2 q_{i \rightarrow j}(u)}{u} + c_3 \right] \\
&= \frac{1}{u^2} \left[-uc_2 \frac{\partial}{\partial u} q_{i \rightarrow j}(u) - [c_1 - c_2 q_{i \rightarrow j}(u)] \right] \\
&= \frac{1}{u^2} \left[-uc_2 \frac{\partial}{\partial u} \left[\frac{\alpha}{\alpha + \beta} [1 - e^{-(\alpha + \beta)u}] \right] - \left[c_1 - c_2 \left[\frac{\alpha}{\alpha + \beta} [1 - e^{-(\alpha + \beta)u}] \right] \right] \right] \\
&= \frac{1}{u^2} \left[-uc_2 \left[\alpha e^{-(\alpha + \beta)u} \right] + c_2 \left[\frac{\alpha}{\alpha + \beta} [1 - e^{-(\alpha + \beta)u}] \right] - c_1 \right] \\
&= \frac{1}{u^2} \left[-uc_2 \left[\alpha \left[\sum_{n=0}^{\infty} \frac{u^n (-1)^n (\alpha + \beta)^n}{n!} \right] \right] + c_2 \left[\frac{\alpha}{\alpha + \beta} \left[1 - \left[\sum_{n=0}^{\infty} \frac{u^n (-1)^n (\alpha + \beta)^n}{n!} \right] \right] \right] - c_1 \right] \\
&= \frac{1}{u^2} \left[\frac{c_2 \alpha}{\alpha + \beta} \left[\sum_{n=0}^{\infty} \frac{(-1)^{n+1} u^{n+1} (\alpha + \beta)^{n+1}}{n!} + 1 + \sum_{n=0}^{\infty} \frac{u^n (-1)^{n+1} (\alpha + \beta)^n}{n!} \right] - c_1 \right] \\
&= \frac{1}{u^2} \left[\frac{c_2 \alpha}{\alpha + \beta} \left[\sum_{n'=1}^{\infty} \frac{(-1)^{n'} u^{n'} (\alpha + \beta)^{n'}}{(n' - 1)!} + \sum_{n=1}^{\infty} \frac{u^n (-1)^{n+1} (\alpha + \beta)^n}{n!} \right] - c_1 \right] \quad n' = n + 1 \\
&= \frac{1}{u^2} \left[\frac{c_2 \alpha}{\alpha + \beta} \left[\sum_{n=2}^{\infty} (-1)^n u^n (\alpha + \beta)^n \left[\frac{1}{(n-1)!} - \frac{1}{n!} \right] \right] - c_1 \right]
\end{aligned}$$

Where last line follows because first ($n = 1$) term of summation is zero.

Thus, $r(u, i, w, \sigma)$ is strictly quasi-convex if $\frac{c_2 \alpha}{\alpha + \beta} \left[\sum_{n=2}^{\infty} (-1)^n u^n (\alpha + \beta)^n \left[\frac{1}{(n-1)!} - \frac{1}{n!} \right] \right] - c_1$ is strictly increasing (derivative is strictly negative up to a point, and then strictly positive above that point u , for a fixed σ .)

$$\begin{aligned}
&\frac{\partial}{\partial u} \left[\frac{c_2 \alpha}{\alpha + \beta} \left[\sum_{n=2}^{\infty} (-1)^n u^n (\alpha + \beta)^n \left[\frac{1}{(n-1)!} - \frac{1}{n!} \right] \right] - c_1 \right] \\
&= \frac{c_2 \alpha}{\alpha + \beta} \left[\sum_{n=2}^{\infty} (-1)^n u^{n-1} (\alpha + \beta)^n \left[\frac{n}{(n-1)!} - \frac{n}{n!} \right] \right] \\
&= \frac{c_2 \alpha}{\alpha + \beta} \left[\sum_{n=2}^{\infty} (-1)^n u^{n-1} (\alpha + \beta)^n \frac{1}{(n-2)!} \right] = \frac{c_2 \alpha}{\alpha + \beta} \left[\sum_{n'=0}^{\infty} (-1)^{n'+2} u^{n'+1} (\alpha + \beta)^{n'+2} \frac{1}{n'!} \right] \quad n' = n - 2 \\
&= c_2 \alpha u (\alpha + \beta) \left[\sum_{n=0}^{\infty} (-1)^n u^n (\alpha + \beta)^n \frac{1}{n!} \right] = c_2 \alpha u (\alpha + \beta) e^{-(\alpha + \beta)u} > 0
\end{aligned}$$

□

Lemma A.4.4. *Suppose $w_i(\tau) = m\tau + a$, where $m > 0$ and $a < 0$. Then, $r(u, i, w, \sigma)$ is strictly quasi-concave in u , for each fixed σ where $\Delta_{ji} > 0$.*

Proof. Proof. Corollary of Lemma A.4.3. $r(u, i, w, \sigma)$ is the negative of the previous case, modulo

constants that do not affect quasi-concavity. \square

Lemma A.4.5. *Fix arbitrary σ_1 , and thus Q_1, T_1, R_1 . Let \bar{Q}_2, \bar{T}_2 be the respective values of Q_2, T_2 at $\sigma_2 = (0, \infty)$. Let $w_2(\tau) = m\tau + zq_{2 \rightarrow 1}(\tau)$, where $m > R_1$.*

If

$$\frac{T_1(\lambda_{2 \rightarrow 1}\bar{T}_2 - \bar{Q}_2) - (Q_1 + T_1\lambda_{2 \rightarrow 1})}{(Q_1(\lambda_{2 \rightarrow 1}\bar{T}_2 - \bar{Q}_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1}))} \leq \frac{z}{m - R_1} \leq \frac{\bar{Q}_2 T_1 + Q_1}{Q_1(\bar{Q}_2 - \lambda_{2 \rightarrow 1})}$$

Then $\frac{\partial}{\partial u} R(w, \sigma) \geq 0$, for all u, σ_2 . Furthermore, the constraint set is feasible regardless of the primitives.

Proof. Proof.

Suppose we have $w_2(u) = mu + zq_{2 \rightarrow 1}(u)$, for some $m > R_1, z \geq 0$.

From Remark A.4.2,

$$\begin{aligned} \frac{\partial}{\partial u} R(w, \sigma) \propto & u \left[\frac{q_{2 \rightarrow 1}(u)}{u} [(R_1 - m)T_1T_2 + mT_1 + zQ_1T_2 + zT_1\lambda_{2 \rightarrow 1}] \right] \\ & + u [Q_2T_1(m - R_1) + Q_1(m - zQ_2 + z\lambda_{2 \rightarrow 1})] \end{aligned}$$

T_2, Q_2 are functions of σ_2 .

As $u \rightarrow \infty$, the term in brackets in the first term goes to 0, and thus the first necessary condition is to have the second term always positive.

If the second term is always positive, then the first term may be negative as long as it has a smaller absolute value than the second term. As $u \rightarrow 0$, the ratio between (absolute value of) the first and second terms is maximized. Thus, the second necessary (and sufficient) condition is to have the entire value positive when we take the limit of $\frac{q_{2 \rightarrow 1}(u)}{u}$ as $u \rightarrow 0$.

These two conditions are sufficient for $\frac{\partial}{\partial u} R(w, \sigma) \geq 0$, for all u, σ_2 .

From the first condition, we need m, z such that:

$$\begin{aligned} & Q_2T_1(m - R_1) + Q_1(m - zQ_2 + z\lambda_{2 \rightarrow 1}) \geq 0 & \forall T_1, Q_1, Q_2, R_1 \\ \iff & \frac{z}{m - R_1} \leq \frac{Q_2T_1 + \frac{m}{m - R_1}Q_1}{Q_1(Q_2 - \lambda_{2 \rightarrow 1})} \end{aligned}$$

From the second condition, and using Remark A.4.3 we need:

$$\begin{aligned}
& \lambda_{2 \rightarrow 1} [(R_1 - m)T_1T_2 + mT_1 + zQ_1T_2 + zT_1\lambda_{2 \rightarrow 1}] + [Q_2T_1(m - R_1) + Q_1(m - zQ_2 + z\lambda_{2 \rightarrow 1})] \geq 0 \\
\iff & (m - R_1)T_1(Q_2 - \lambda_{2 \rightarrow 1}T_2) + m(Q_1 + \lambda_{2 \rightarrow 1}T_1) + zQ_1(\lambda_{2 \rightarrow 1}T_2 - Q_2 + \lambda_{2 \rightarrow 1}) + zT_1\lambda_{2 \rightarrow 1}^2 \geq 0 \\
\iff & z(Q_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1})) \geq (m - R_1)T_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) - m(Q_1 + T_1\lambda_{2 \rightarrow 1}) \\
\iff & \frac{z}{m - R_1} \geq \frac{T_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) - \frac{m}{m - R_1}(Q_1 + T_1\lambda_{2 \rightarrow 1})}{(Q_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1}))}
\end{aligned}$$

Putting the conditions together, we need, for all T_i, Q_i, R_i :

$$\frac{T_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) - \frac{m}{m - R_1}(Q_1 + T_1\lambda_{2 \rightarrow 1})}{(Q_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1}))} \leq \frac{z}{m - R_1} \leq \frac{Q_2T_1 + \frac{m}{m - R_1}Q_1}{Q_1(Q_2 - \lambda_{2 \rightarrow 1})}$$

$m > R_1$ by supposition, and so $\frac{m}{m - R_1} > 1$. Thus, the following is sufficient as the constraints become tighter:

$$\begin{aligned}
& \frac{T_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) - (Q_1 + T_1\lambda_{2 \rightarrow 1})}{(Q_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1}))} \leq \frac{z}{m - R_1} \leq \frac{Q_2T_1 + Q_1}{Q_1(Q_2 - \lambda_{2 \rightarrow 1})} \\
\iff & \frac{T_1 - \frac{Q_1 + T_1\lambda_{2 \rightarrow 1}}{(\lambda_{2 \rightarrow 1}T_2 - Q_2)}}{Q_1 + \frac{\lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1})}{(\lambda_{2 \rightarrow 1}T_2 - Q_2)}} \leq \frac{z}{m - R_1} \leq \frac{T_1 + \frac{Q_1}{Q_2}}{Q_1 \left(1 - \frac{\lambda_{2 \rightarrow 1}}{Q_2}\right)}
\end{aligned}$$

It turns out that both constraints are tightest when $\sigma_2 = (0, \infty)$. In the left constraint, the numerator is increasing and the denominator is decreasing with $\lambda_{2 \rightarrow 1}T_2 - Q_2$, and so the constraint becomes tighter as $\lambda_{2 \rightarrow 1}T_2 - Q_2$ increases. By Remark A.4.4, $\lambda_{2 \rightarrow 1}T_2 - Q_2$ is always positive, and maximized when $\sigma_2 = (0, \infty)$. Similarly, in the right constraint, the numerator decreases and the denominator increases with Q_2 .

Thus, it is sufficient for the two constraints to be feasible for $\sigma_2 = (0, \infty)$. Then, they are satisfied for all σ'_2 . For feasibility, we need

$$\begin{aligned}
& \frac{T_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) - (Q_1 + T_1\lambda_{2 \rightarrow 1})}{(Q_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1}))} \leq \frac{Q_2T_1 + Q_1}{Q_1(Q_2 - \lambda_{2 \rightarrow 1})} \\
\iff & (T_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) - (Q_1 + T_1\lambda_{2 \rightarrow 1}))Q_1(Q_2 - \lambda_{2 \rightarrow 1}) \\
& \leq (Q_2T_1 + Q_1)(Q_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1})) \\
\iff & Q_1Q_2(T_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) - (Q_1 + T_1\lambda_{2 \rightarrow 1})) - Q_1\lambda_{2 \rightarrow 1}(T_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) - (Q_1 + T_1\lambda_{2 \rightarrow 1})) \\
& \leq Q_2T_1(Q_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1})) + Q_1(Q_1(\lambda_{2 \rightarrow 1}T_2 - Q_2) + \lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1})) \\
\iff & Q_1Q_2(-(Q_1 + T_1\lambda_{2 \rightarrow 1})) - Q_1\lambda_{2 \rightarrow 1}(T_1(\lambda_{2 \rightarrow 1}T_2 - Q_2)) \\
& \leq Q_2T_1(\lambda_{2 \rightarrow 1}(Q_1 + T_1\lambda_{2 \rightarrow 1})) + Q_1(Q_1(\lambda_{2 \rightarrow 1}T_2 - Q_2))
\end{aligned}$$

For any valid Q_i, T_i , the left hand side of the final line is always non-positive, and the right hand

side is always non-negative, and thus there exists feasible ratios $\frac{z}{m-R_1}$.

□

Lemma A.4.6. *Fix arbitrary σ_2 , and thus Q_2, T_2, R_2 . Let \bar{Q}_1, \bar{T}_1 be the respective values of Q_1, T_1 at $\sigma_1 = (0, \infty)$. Let $w_1(\tau) = m\tau + zq_{1 \rightarrow 2}(\tau)$, where $m = R_2$.*

If

$$-\frac{(T_2\lambda_{1 \rightarrow 2} + Q_2)}{Q_2(\lambda_{1 \rightarrow 2}\bar{T}_1 - \bar{Q}_1) + \lambda_{1 \rightarrow 2}(T_2\lambda_{1 \rightarrow 2} + Q_2)} \leq \frac{z}{R_2} \leq \frac{1}{(\bar{Q}_1 - \lambda_{1 \rightarrow 2})}$$

Then $\frac{\partial}{\partial u}R(w, \sigma) \geq 0$, for all u, σ_1 . Furthermore, the constraint set is feasible regardless of the primitives.

Proof. Proof. Similar to previous proof. Suppose we have $w_1(u) = mu + zq_{1 \rightarrow 2}(u)$, for some $m = R_2, z \leq 0$.

From Remark A.4.2,

$$\begin{aligned} \frac{\partial}{\partial u}R(w, \sigma) &= u \left[\frac{q_{1 \rightarrow 2}(u)}{u} [(R_2 - m)T_1T_2 + mT_2 + zQ_2T_1 + zT_2\lambda_{1 \rightarrow 2}] \right] \\ &\quad + u [Q_1T_2(m - R_2) + Q_2(m - zQ_1 + z\lambda_{1 \rightarrow 2})] \\ &= u \left[\frac{q_{1 \rightarrow 2}(u)}{u} [R_2T_2 + zQ_2T_1 + zT_2\lambda_{1 \rightarrow 2}] + [Q_2(R_2 - zQ_1 + z\lambda_{1 \rightarrow 2})] \right] \end{aligned}$$

As before, we have two necessary and sufficient conditions for $\frac{\partial}{\partial u}R(w, \sigma) \geq 0$, for all u, σ_1 .

From the first condition, we need m, z such that:

$$\begin{aligned} Q_2(R_2 - z(Q_1 - \lambda_{1 \rightarrow 2})) &\geq 0 && \forall T_2, Q_2, Q_1, R_2 \\ \iff \frac{z}{R_2} &\leq \frac{1}{(Q_1 - \lambda_{1 \rightarrow 2})} \end{aligned}$$

This condition is trivially met when $z \leq 0$.

Similarly, the second condition becomes

$$\begin{aligned} \lambda_{1 \rightarrow 2} [R_2T_2 + zQ_2T_1 + zT_2\lambda_{1 \rightarrow 2}] + [Q_2(R_2 - zQ_1 + z\lambda_{1 \rightarrow 2})] &\geq 0 \\ \iff \lambda_{1 \rightarrow 2}R_2T_2 + Q_2R_2 &\geq -zQ_2(\lambda_{1 \rightarrow 2}T_1 - Q_1) - z\lambda_{1 \rightarrow 2}(T_2\lambda_{1 \rightarrow 2} + Q_2) \\ \iff \frac{z}{R_2} &\geq -\frac{(T_2\lambda_{1 \rightarrow 2} + Q_2)}{Q_2(\lambda_{1 \rightarrow 2}T_1 - Q_1) + \lambda_{1 \rightarrow 2}(T_2\lambda_{1 \rightarrow 2} + Q_2)} \end{aligned}$$

Both constraints are tightest when $\sigma_1 = (0, \infty)$. By Remark A.4.4, $\lambda_{1 \rightarrow 2}T_1 - Q_1$ is always positive, and maximized when $\sigma_1 = (0, \infty)$.

As one is non-negative and the other is non-positive, the constraints are feasible.

□

Appendix B

Designing Informative Rating Systems: Evidence from an Online Labor Market

B.1 Further analysis of the labor market test

In this section, we report more detail from the test on the online labor market. For much of this section, we analyze a subset of the jobs: some job covariate information is missing in what was given to us by the labor market. We have full covariate data for 100438 jobs (out of 184172).

B.1.1 Verifying randomization in allocation of clients

As noted in Section 3.3.3 of the main paper, there was a bug in the allocation code such that 1,086 clients were assigned to different treatment cells upon submissions of different jobs. Since this could potentially create contamination between our cells, we disregard these clients in our analysis. Here we make sure that neither this bug nor any other affected experimental validity by checking the distribution of client covariates across the treatment cells. We do so as follows.

We have a set of *job* level covariates for a subset of the jobs: *hourly rate of job* (if applicable), *total cost of project if not hourly* (if applicable), *previous number of closed jobs by client at time of job*, *previous spend by client at time of job*, *value of the job* (4 options), *Tier 1 category* (12 options), *Tier 2 category* (88 options), and *expertise level* (3 options). The first four are continuous covariates, and the last 4 are categorical covariates.

For each client, we sample one of that client's jobs and associate the client with that job's covariates. Then we run tests of independence for the samples of each covariate across the treatment

cells. Across a variety of tests and all covariates, the results are consistent with the randomization being valid.

- For each continuous covariate, using the Kruskal-Wallis H-test for independent samples on all the treatment groups together, the null hypothesis that the population median of all of the groups are equal is not rejected, with $p > .9$.
- Similarly, for each continuous covariate, using the one way ANOVA F test, the null hypothesis that all the treatment groups have the same population mean is not rejected, with $p > .2$.
- For each categorical covariate, we run the chi-squared test of independence of variables in a contingency table, which tests whether the observed frequencies of values is independent of the treatment group. The null hypothesis is not rejected with $p > .1$, for each covariate.

These tests are consistent with fact that the allocation of *valid* clients we used for analysis across treatment cells was truly random. Note that these tests do not check whether the *invalid* clients (which we threw out) are similar to the *valid* clients. Invalid clients are more likely to be higher volume clients, as those who submitted many jobs during the test period provided more chances for the bug to manifest.

B.1.2 Robustness against high volume clients and allocation bug

Recall that in the main text we further threw out the 7 clients who submitted more than 200 jobs during the test period (“heavy users”). However, the following may still be the case: idiosyncratic rating behavior of medium-volume clients (over 50 or 100 jobs submitted) may be driving the difference in behavior between treatment cells. Here we show that this is not the case, as well as the fact that throwing out the 7 heavy users was not consequential. We further show that including the clients who were thrown out due to the allocation bug does not materially affect results.

In Figure B.1, we plot the rating distributions when only sampling 1 job/client, including 7 clients excluded for submitting at least 200 jobs during the test period, and using all jobs and clients (even incorrectly allocated clients). The mean treatment responses are also included. Results are similar.

Data sampling policy:	From main text	One job per client	With outlier clients	All clients, even incorrectly allocated
<i>Expectations</i>	3.339	3.243	3.354	3.350
<i>Adjectives</i>	3.650	3.597	3.650	3.651
<i>Average</i>	3.763	3.687	3.788	3.774
<i>Average, not affect score</i>	3.777	3.693	3.777	3.771
<i>Average, Randomized</i>	3.465	3.438	3.463	3.458
<i>Numeric</i>	3.594	4.534	4.635	4.639

Table B.1: Average treatment responses under different data policies

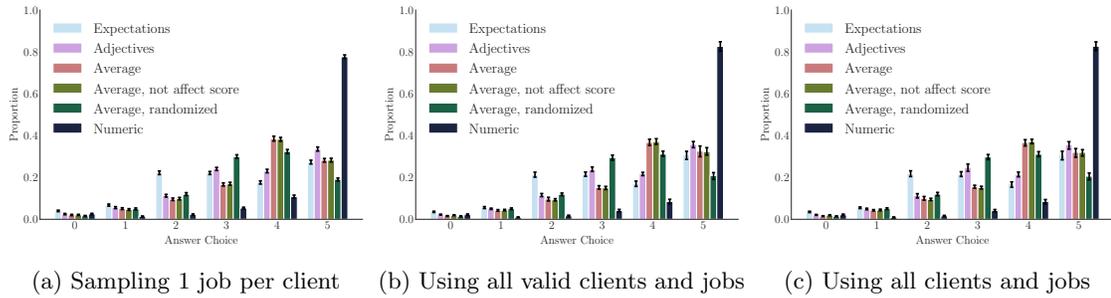


Figure B.1: Rating distributions for different client sampling techniques. As in the main text, the confidence intervals are 95% bootstrapped confidence intervals, with bootstrapped sampling at the client level.

B.1.3 Regressing treatment response with treatment cell and other covariates

We regress the treatment response with treatment cell and all of our job covariates (except tier 2 category, which had 88 unique values and is a more granular version of tier 1 category). (Note: to maintain full rank, each categorical covariate is encoded such that one of the levels is missing, except for treatment cell, and there is no intercept. As a result, the treatment cell coefficients cannot be interpreted as treatment means – they are the treatment means conditional on a specific value of each of the categorical covariates and of 0 for the continuous variables). Further note that for simplicity, we only include one set of interaction terms: treatment cell vs. the number of previous treatment responses. Finally, note that the displayed standard errors are cluster-robust standard errors where each client is a cluster, to take into account that ratings given by the same client are correlated. We learn several things from this regression, displayed in Table B.2:

- There is some heterogeneity in ratings across the job covariates, but on the order of .1 points on the average rating. This heterogeneity is dwarfed by the differences between the treatment cells, especially the numeric vs. non-numeric treatments. This relative lack of heterogeneity further supports that the differences between the mean treatment responses are not due to randomness caused by some types of jobs being more present in some treatment groups than others.
- We can directly measure the effect of the number of previous jobs during that testing period a given client has submitted, i.e., estimate the inflation that will result over time as clients submit additional jobs.

From the table below, each additional job a client has submitted raises the treatment response for the *Expectations* and the *Averages* treatments, on the order of .008 to .014 points per previous response. At this rate, these coefficients suggest that only after giving 100 ratings

would a client inflate ratings by an average of between .8 and 1.4 points. The *Numeric* treatment cell does not further inflate substantially.

Dep. Variable:		treatment-response	R-squared:		0.128		
Model:		OLS	Adj. R-squared:		0.128		
Method:		Least Squares	Log-Likelihood:		-1.6001e+05		
No. Observations:		100438	AIC:		3.201e+05		
Df Residuals:		100406	BIC:		3.204e+05		
Df Model:		31					
		coef	std err	z	P> z	[0.025	0.975]
treatment_cell[1]		3.0596	0.062	49.052	0.000	2.937	3.182
treatment_cell[2]		3.3965	0.063	53.862	0.000	3.273	3.520
treatment_cell[3]		3.4516	0.062	55.353	0.000	3.329	3.574
treatment_cell[4]		3.4414	0.062	55.796	0.000	3.321	3.562
treatment_cell[5]		3.1887	0.062	51.379	0.000	3.067	3.310
treatment_cell[6]		4.3745	0.062	70.044	0.000	4.252	4.497
value_group[T.lv]		0.1031	0.034	2.998	0.003	0.036	0.170
value_group[T.mv]		0.0206	0.034	0.601	0.548	-0.047	0.088
value_group[T.vlv]		0.2920	0.032	9.061	0.000	0.229	0.355
category_group[T.Admin Support]		-0.0591	0.046	-1.281	0.200	-0.150	0.031
category_group[T.Customer Service]		-0.1070	0.081	-1.320	0.187	-0.266	0.052
category_group[T.Data Science & Analytics]		0.1177	0.050	2.354	0.019	0.020	0.216
category_group[T.Design & Creative]		0.1077	0.042	2.581	0.010	0.026	0.189
category_group[T.Engineering & Architecture]		0.1235	0.058	2.122	0.034	0.009	0.238
category_group[T.IT & Networking]		0.1277	0.049	2.595	0.009	0.031	0.224
category_group[T.Legal]		0.0643	0.061	1.047	0.295	-0.056	0.185
category_group[T.Sales & Marketing]		-0.0869	0.045	-1.920	0.055	-0.176	0.002
category_group[T.Translation]		0.0405	0.060	0.676	0.499	-0.077	0.158
category_group[T.Web, Mobile & Software Dev]		0.0940	0.042	2.256	0.024	0.012	0.176
category_group[T.Writing]		-0.1158	0.044	-2.638	0.008	-0.202	-0.030
expertise_tier[T.Expert/Expensive]		0.1465	0.020	7.276	0.000	0.107	0.186
expertise_tier[T.Intermediate]		0.0582	0.018	3.306	0.001	0.024	0.093
hr_charge		1.376e-05	2.16e-06	6.376	0.000	9.53e-06	1.8e-05
fp_charge		3.64e-05	6.73e-06	5.409	0.000	2.32e-05	4.96e-05
log(1 +client_prev_spend)		-0.0069	0.006	-1.156	0.248	-0.018	0.005
log(1 +num_prev_asg)		-0.0177	0.010	-1.769	0.077	-0.037	0.002
treatment_cell[1]:# prev. treatment resps. by client		0.0080	0.004	2.042	0.041	0.000	0.016
treatment_cell[2]:# prev. treatment resps. by client		-0.0043	0.006	-0.675	0.500	-0.017	0.008
treatment_cell[3]:# prev. treatment resps. by client		0.0085	0.003	2.850	0.004	0.003	0.014
treatment_cell[4]:# prev. treatment resps. by client		0.0141	0.003	5.468	0.000	0.009	0.019
treatment_cell[5]:# prev. treatment resps. by client		0.0024	0.005	0.485	0.628	-0.007	0.012
treatment_cell[6]:# prev. treatment resps. by client		0.0010	0.004	0.246	0.806	-0.007	0.009
Omnibus:	11064.189	Durbin-Watson:	1.911				
Prob(Omnibus):	0.000	Jarque-Bera (JB):	15421.891				
Skew:	-0.876	Prob(JB):	0.00				
Kurtosis:	3.785	Cond. No.	9.60e+04				

Table B.2: OLS Regression Results with covariate for previous number of treatment responses

B.1.4 More on inflation over time

The interpretations above suffer from selection bias: the set of clients who submit 10 jobs in the test period are a different cohort than those who submit fewer. This effect is partially captured by the term containing the previous number of client assignments. To address this issue, we repeat the regression in Table B.2, limiting the analysis to those clients who have more than ten treatment responses during the test period (all of which have the job covariates). The table is omitted; the coefficients for inflation over time are largely the same.

To further help visualize (the relative lack of) inflation over the number of submitted ratings, Figure B.2 shows the mean ratings for each treatment cell by the number of previous treatment responses given during the test period. As the plot has no covariate data, we use the first ten responses for all 2145 clients who submitted at least 10 ratings during the test period. Clients are not substantially more likely to give more positive ratings on their 10th rating during the test than

they give on their first rating.

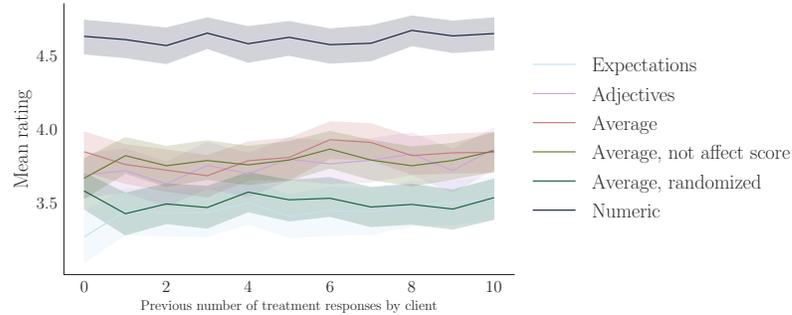


Figure B.2: Mean ratings for each treatment cell by the number of previous treatment responses given during the test period. Error bands are bootstrapped 95% confidence intervals.

B.1.5 Analysis of cell with randomized order of answer choices

The *Average, Randomized* contained the same question and answer choices as the *Average* condition, but the choices were presented in a random order. If the raters read all the answer choices and pick the most applicable one, then this condition would have returned a rating distribution identical to that of the *Average* condition. However, it does not. Furthermore, the *location* of the chosen choice would be distributed uniformly, i.e., the rater should pick the choice presented first as much as she picks other choices. We find this not to be the case: the first answer choice presented to the rater is picked $6806/26978 = 25.2\%$ of the time. The second through sixth answer choices are picked 17.3%, 14.7%, 14.3%, 13.9%, and 14.5% of the time each, respectively.

This phenomenon suggests that (a) a small percentage (up to 10 – 13%) of raters do not read the answer choices at all and simply select the first answer choice, and (b) many raters start reading from the first presented choice and select the first one that approximately describes their experience. Our test design cannot disambiguate between these (or other plausible) explanations. Nevertheless, this effect is second-order relative to the overall finding that more descriptive scales are substantially more informative than numeric scales, and the *Average, Randomized* treatment results are comparable to those of other verbal scales.

B.1.6 Design approach using labor market data

Table B.3 and Figures B.3 and B.5 contain supplementary information regarding our application of the design approach to the labor market data, as described in the main text.

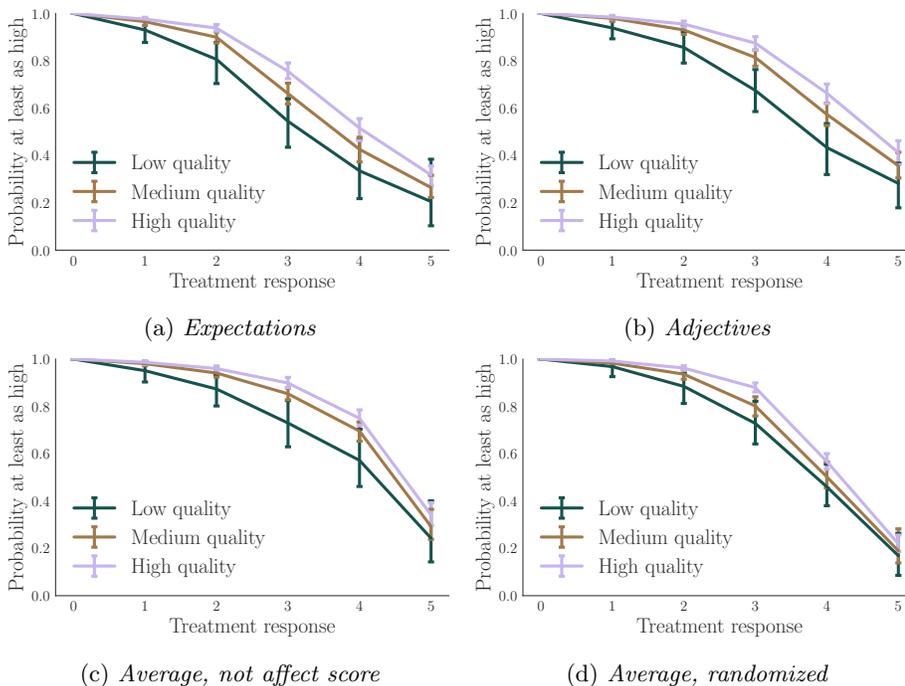


Figure B.3: Joint distributions of freelancer quality vs. ratings in the other treatment cells. Low, Medium, and High quality sellers refer to those with other cell average ratings in $[0, 2)$, $[2.5, 3.5)$ and $[4.5, 5]$, respectively.

B.2 Amazon Mechanical Turk synthetic experiment

In this section, we deploy an experiment on Amazon Mechanical Turk (“MTurk”) to repeat and analyze our design approach, in a synthetic setting where we have expert (external) quality information on items. We note that this section is not a replication of the behavioral components of our results, as the MTurk and online labor market settings are too different to meaningfully compare. Furthermore, one should be aware of limitations of using MTurk convenience samples in research (Landers and Behrend, 2015); such limitations mean that there will be behavioral biases that differ from those on other platforms. For these reasons, this section should be seen as a synthetic, example application of our overall comparison and design methodology to other domains, and in particular will show how our methods are useful not just to counter rating inflation but also other types of biases. This appendix section is organized as follows. In B.2.1 we describe the task, and in B.2.2 we repeat our analysis from the main text, including: showing the resulting marginal and joint distributions of ratings and quality, and testing designs on new, unseen data.

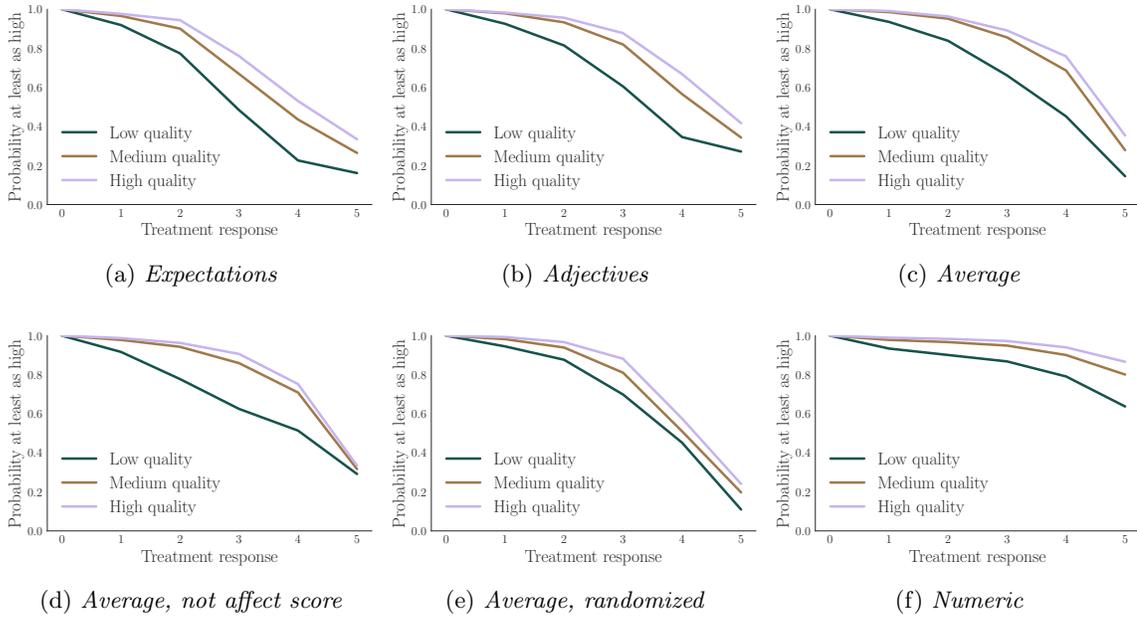


Figure B.4: Joint distributions, where Low, Medium, and High quality sellers refer to those with other cell average ratings in $[0, 2)$, $[2, 4)$ and $[4, 5]$, respectively.

B.2.1 Experiment description

Task Information

We asked subjects to rate the English proficiency of 10 paragraphs which are modified TOEFL (Test of English as a Foreign Language) essays with known scores as determined by experts and reported in a TOEFL study guide (Educational Testing Service, 2005); these are our true quality types for each essay. Expert scores range from 1 through 5, with two paragraphs with each score. Essays are shortened to a single paragraph of just a few sentences, and the top rated paragraphs are improved and the worst ones are made worse; this is largely to ensure the quality could be sufficiently distinguished between paragraphs despite having shortened them. In other words, for each topic, we

Condition	Response Score					
	0	1	2	3	4	5
Expectations	1.22	1.22	2.28	3.74	4.38	5.00
Adjectives	1.47	1.55	1.63	3.22	4.97	5.00
Average	1.80	1.84	1.88	2.53	3.83	5.00
Average, not affect score	0.89	1.57	1.59	3.32	4.04	5.00
Average, randomized	0.72	2.41	2.63	4.18	4.30	5.00
Numeric	0.50	1.20	1.98	2.88	3.45	5.00

Table B.3: Optimal scores ϕ for each treatment, where the score of the top position is normalized to 5.

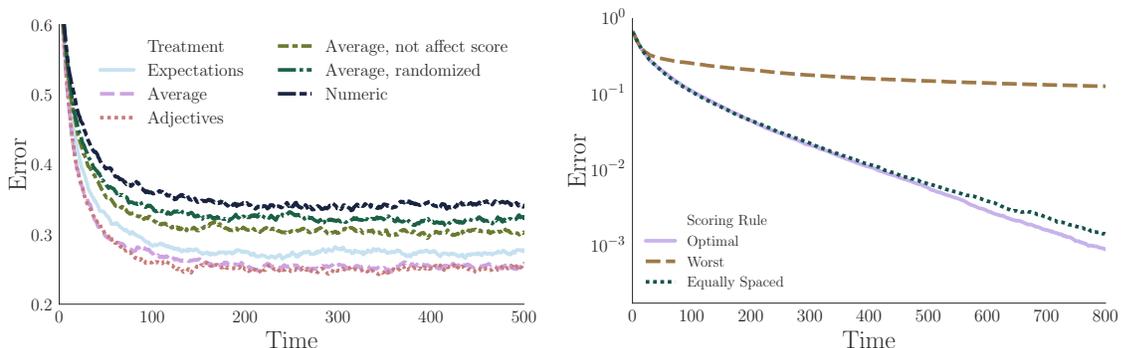
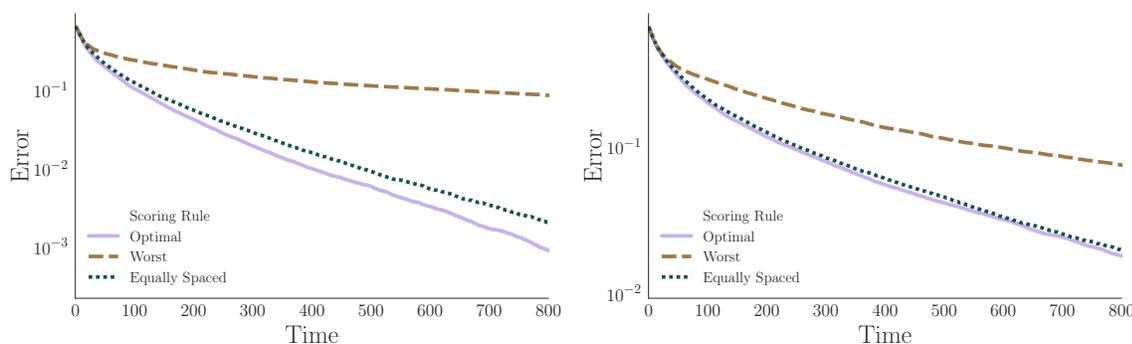
(a) With optimal ϕ and probability of exit of 0.01.(b) *Average* treatment with different scoring rules(c) *Adjectives* treatment with different scoring rules(d) *Numeric* treatment with different scoring rules

Figure B.5: Simulated performance over time with various other configurations. The “Worst” scoring rule corresponds to the rule ϕ with the smallest learning rate found for each treatment.

improved the language of the best rated paragraph and further degraded the language of the worst one. In principle, our editing of these paragraphs may remove the validity of the expert ratings. However, the estimated $R(\theta, y|Y)$ indicates that this does not substantially occur, suggesting our editing of the paragraphs preserved the quality ordering of the paragraphs per the expert ratings.

Subjects were given one of five possible verbal scales, where the scales were designed using a list of adjectives, $\{Abysmal, Awful, Bad, Poor, Mediocre, Fair, Good, Great, Excellent, Phenomenal\}$, compiled by Hicks et al. (2000). Each scale had five options. The scales are:

- **Every Other:** *Awful, Poor, Fair, Great, Phenomenal*
- **Close to Every Other:** *Abysmal, Poor, Mediocre, Good, Phenomenal*
- **Extremes:** *Abysmal, Awful, Bad, Excellent, Phenomenal*
- **Negative-skewed:** *Abysmal, Awful, Bad, Poor, Mediocre*
- **Positive-skewed:** *Fair, Good, Great, Excellent, Phenomenal*

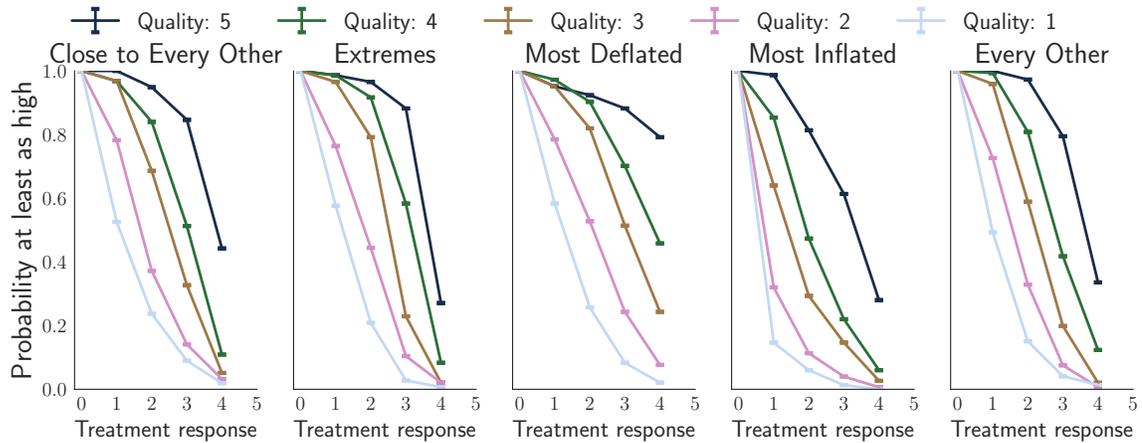


Figure B.6: Joint distributions of rating and expert score on the MTurk training set, by treatment condition.

We note that it is not a priori clear which of these scales will perform well in this setting, or what the optimal scoring mapping should be.

Raters (i.e., mTurk workers) were shown each of the ten paragraphs. The instructions were: “Please rate on English proficiency (grammar, spelling, sentence structure) and coherence of the argument, but not on whether you agree with the substance of the text.” The specific question then asked was: “How does the following rate on English proficiency and argument coherence?” One paragraph was shown per page; returning to modify a previous answer was not allowed; and paragraphs were presented in a random order. Each rater was shown one of the scales picked at random, and the same scale was used for all paragraphs for that rater. There were approximately 500 raters overall across the 5 treatment cells, with between 97 and 104 raters in each cell. For each cell, we divide the raters (randomly) into train (75%) and test (25%). We design optimal scoring rules using the training data, and then test performance on the test data.

Rater logistics

We did not exclude any data, and all raters were paid \$1.50. Instructions advised raters to spend no more than a minute per question, though this was not enforced. The median rater spent 325 seconds, corresponding to a median wage of \$16.61/hr. About 80% of raters spent 8 minutes or less.

B.2.2 Results

We now repeat the design and test procedure from the main text, for this setting. All plots, figures, and scoring rules are generated exactly as in the main text, with the following exceptions: (1) we have true expert scores for the paragraph qualities and so do not use the procedure where we

Condition	Learning rates	
	Equally spaced ϕ	Optimal ϕ
Every Other	0.058	0.069
Extremes	0.077	0.079
Negative-skewed	0.051	0.059
Positive-skewed	0.034	0.043
Close to Every Other	0.043	0.044

Table B.4: Large deviation learning rates for each treatment in the Mturk experiment, calculated using Equation (3.4) and the joint distributions generated using the training data plotted in Figure B.6. Optimal for each treatment corresponds to the highest learning rate among many random score functions tested.

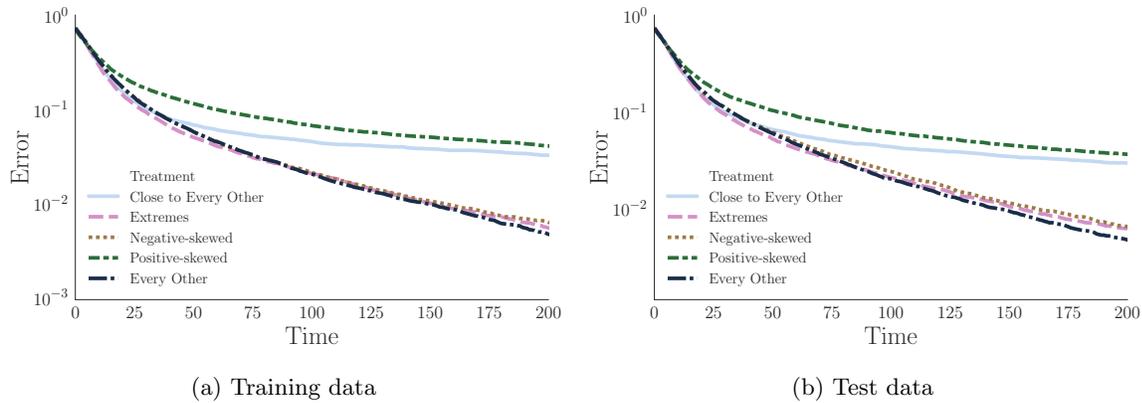


Figure B.7: Simulated performance of each rating scale with Equally Spaced scores.

estimate such qualities from the other treatment cells, and (2) we split the rater responses into test and training sets. We show the joint distributions $R(\cdot|Y)$ and optimal scores calculated from the training set, and then we evaluate performance on simulations using the test set.

Figure B.6 shows the joint distributions of rating and expert score on the MTurk training set, for each treatment condition. Table B.4 shows the training set learning rates for each treatment using equally spaced scores, as well as the best performing scores, respectively.

Finally, we simulate the performance of the designs (generated using the training data), following the same simulation technique as outlined in the main text. We also evaluate performance on the test data, in order to demonstrate how a platform would use our design approach. Figure B.7 shows the resulting errors over time with Equally Spaced scores. Errors with optimal scores are qualitatively similar.

B.3 Proofs

Lemma B.3.1.

$$\lim_{k \rightarrow \infty} -\frac{1}{k} \log [\mu((x_k(\theta_1) - x_k(\theta_2)) \leq 0 | \theta_1, \theta_2)] = \inf_{a \in \mathcal{R}} \{g(\theta_1)I(a|\theta_1) + g(\theta_2)I(a|\theta_2)\}$$

where $I(a|\ell) = \sup_z \{za - \Lambda(z|\theta)\}$, $\Lambda(z|\theta)$ is the log moment generating function of a single sample from $x(\theta_1)$, and $g(\theta)$ is the sampling rate.

Proof. $\lim_{k \rightarrow \infty} -\frac{1}{k} \log [\mu((x_k(\theta_1) - x_k(\theta_2)) \leq 0 | \theta_1, \theta_2)]$

$$= \lim_{k \rightarrow \infty} -\frac{1}{k} \log \left[\int_{a \in \mathcal{R}} \mu((x_k(\theta_1) = a | \theta_1) \mu(x_k(\theta_2) \geq a | \theta_2) da \right] \tag{B.1}$$

$$= \lim_{k \rightarrow \infty} -\frac{1}{k} \log \left[\int_{a \in \mathcal{R}} e^{-kg(\theta_1)I(a|\theta_1)} e^{-kg(\theta_2)I(a|\theta_2)} da \right] \tag{B.2}$$

$$= \inf_{a \in \mathcal{R}} \{g(\theta_1)I(a|\theta_1) + g(\theta_2)I(a|\theta_2)\} \tag{B.3}$$

Laplace principle

Where (B.2) is a basic result from large deviations, and $kg(\theta_i)$ is the number of samples item of quality θ_i has received. \square

This lemma also appears in Glynn and Juneja (2004), which uses the Gartner-Ellis Theorem in the proof. Our proof is conceptually similar but instead uses Laplace’s principle.

We can now establish the rate function for $P_k(\theta_1, \theta_2)$.

Recall $P_k(\theta_1, \theta_2) = \mu_k(x_k(\theta_1) > x_k(\theta_2) | \theta_1, \theta_2) - \mu_k(x_k(\theta_1) < x_k(\theta_2) | \theta_1, \theta_2)$. Then, we have

Lemma B.3.2. Given θ_1, θ_2 , let $\bar{P}_k(\theta_1, \theta_2) = 1 - P_k(\theta_1, \theta_2)$. Then:

$$-\lim_{k \rightarrow \infty} \frac{1}{k} \log \bar{P}_k(\theta_1, \theta_2) = \inf_{a \in \mathcal{R}} \{g(\theta_1)I(a|\theta_1) + g(\theta_2)I(a|\theta_2)\}, \quad (\text{B.4})$$

where $I(a|\theta) = \sup_z \{za - \Lambda(z|\theta)\}$, and $\Lambda(z|\theta)$ is the log moment generating function of a single rating given to seller of type θ :

$$\Lambda(z|\theta) = \log \sum_{y \in Y} \rho(\theta, y|Y) \exp(z\phi(y)).$$

Proof. Follows directly from Lemma B.3.1.

$$\begin{aligned} & -\lim_{k \rightarrow \infty} \frac{1}{k} \log \bar{P}_k(\theta_1, \theta_2|\beta) \\ &= \lim_{k \rightarrow \infty} -\frac{1}{k} \log [1 + \mu_k(x_k(\theta_1) - x_k(\theta_2) < 0|\theta_1, \theta_2) - \mu_k(x_k(\theta_1) - x_k(\theta_2) > 0|\theta_1, \theta_2)] \\ &= \lim_{k \rightarrow \infty} -\frac{1}{k} \log [2\mu_k(x_k(\theta_1) - x_k(\theta_2) < 0|\theta_1, \theta_2) + \mu_k(x_k(\theta_1) - x_k(\theta_2) = 0|\theta_1, \theta_2)] \\ &= \inf_{a \in \mathcal{R}} \{g(\theta_1)I(a|\theta_1) + g(\theta_2)I(a|\theta_2)\} \end{aligned} \quad \text{Lemma B.3.1}$$

□

Now we show that this rate function transfers to a rate function for W_k .

Proof of Theorem 1

$$r \triangleq -\lim_{k \rightarrow \infty} \frac{1}{k} \log(1 - W_k) = \min_{0 \leq i < M} \inf_{a \in \mathcal{R}} \{g(\theta_{i+1})I(a|\theta_{i+1}) + g(\theta_i)I(a|\theta_i)\} \quad (\text{B.5})$$

where $I(a|\theta) = \sup_z \{za - \Lambda(z|\theta)\}$, and $\Lambda(z|\theta) = \log \sum_{y \in Y} \rho(\theta, y|Y) \exp(z\phi(y))$ is the log moment generating function of a single rating given to seller of type θ .

Proof.

$$-\lim_{k \rightarrow \infty} \frac{1}{k} \log(1 - W_k) = -\lim_{k \rightarrow \infty} \frac{1}{k} \log \left(1 - \frac{2}{M(M-1)} \sum_{\theta_1 > \theta_2 \in \Theta} P_k(\theta_1, \theta_2) \right) \quad (\text{B.6})$$

$$= -\lim_{k \rightarrow \infty} \frac{1}{k} \log \frac{2}{M(M-1)} \sum_{0 \leq i < j \leq M} \bar{P}_k(\theta_j, \theta_i) \quad (\text{B.7})$$

$$= -\max_{0 \leq i < j \leq M} \left(\lim_{k \rightarrow \infty} \frac{1}{k} \log (\bar{P}_k(\theta_j, \theta_i)) \right) = \min_{0 \leq i < j \leq M} \left(-\lim_{k \rightarrow \infty} \frac{1}{k} [\log (\bar{P}_k(\theta_j, \theta_i))] \right) \quad (\text{B.8})$$

$$= \min_{0 \leq i < j \leq M} \inf_{a \in \mathcal{R}} \{g(\theta_j)I(a|\theta_j) + g(\theta_i)I(a|\theta_i)\} \quad (\text{B.9})$$

$$= \min_{0 \leq i < M} \inf_{a \in \mathcal{R}} \{g(\theta_{i+1})I(a|\theta_{i+1}) + g(\theta_i)I(a|\theta_i)\} \quad (\text{B.10})$$

Where the last line follows from adjacent θ_i, θ_{i+1} dominating the rate due to properties of R . Line (B.8) follows from: $\forall a_i^\epsilon \geq 0$, $\limsup_{\epsilon \rightarrow 0} \left[\epsilon \log \left(\sum_i^N a_i^\epsilon \right) \right] = \max_i^N \limsup_{\epsilon \rightarrow 0} \epsilon \log(a_i^\epsilon)$. See, e.g., Lemma 1.2.15 in Dembo and Zeitouni (2010) for a proof of this property.

□

Appendix C

Designing Optimal Binary Rating Systems

C.1 Mechanical Turk experiment, simulations, and results

In this section, we expand upon the results discussed in Section 4.5. We design and run an experiment that a real platform may run to design a rating system. We follow the general framework in Section 4.4. We first run an experiment to estimate a $\psi(\theta, y)$, the probability at which each item with quality θ receives a positive answer under different questions y . Then, we design $H(y)$, using our optimal β for various settings (different objectives w and matching rates g). Then, we simulate several markets (using the various matching rates g) and measure the performance of the different rating system designs H , as measured by various objective functions (4.2).

C.1.1 Experiment description

We now describe our Mechanical Turk experiment. We ask subjects to rate the English proficiency of ten paragraphs. These paragraphs are modified TOEFL (Test of English as a Foreign Language) essays with known scores as determined by experts (Educational Testing Service, 2005). Subjects were given six answer choices, drawn randomly from the following list: *Abysmal*, *Awful*, *Bad*, *Poor*, *Mediocre*, *Fair*, *Good*, *Great*, *Excellent*, *Phenomenal*, following the recommendation of Hicks et al. (2000). *Poor* and *Good* are always chosen, and the other four are sampled uniformly at random for each worker. One paragraph is shown per page; returning to modify a previous answer is not allowed; and paragraphs are presented in a random order. This data is used to calibrate a model of ψ for optimization, i.e. to simulate a system with a set of questions \mathcal{Y} , where each question y corresponds to a adjective, “Would you characterize the performance of this item as [adjective] or

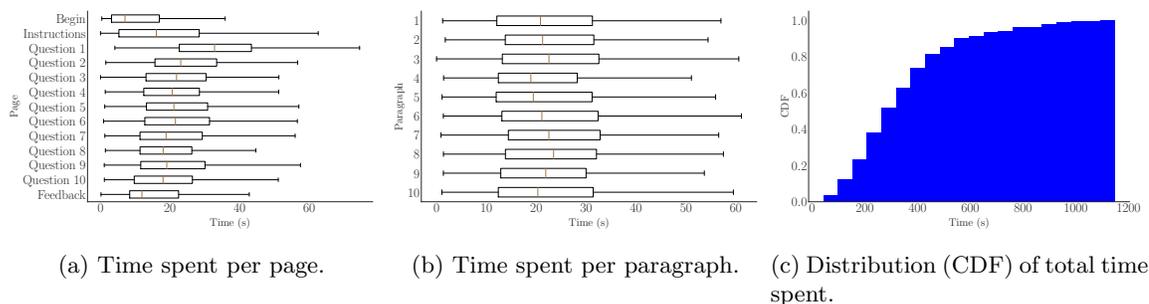


Figure C.1: Additional information for MTurk experiment

better?”¹

Different experiment trials are described below. Pilots were primarily used to garner feedback regarding the experiment from workers (fair pay, time needed to complete, website/UI comments, etc). All trials yield qualitatively similar results in terms of both paragraph ratings and feedback rating distributions for various scales.

Pilot 1 30 workers. Similar conditions as final experiment (6 words sampled for paragraph ratings, all uniformly at random, 5 point scale feedback rating), with identical question phrasing, “How does the following rate on English proficiency and argument coherence?”.

Pilot 2 30 workers. 7 words sampled for paragraph ratings, 6 point scale feedback rating, with the following question phrasing: “How does the following person rate on English proficiency and argument coherence?”.

Experiment 200 workers. 6 words sampled for paragraph ratings, with 2 fixed as described above, 5 point scale feedback rating. Question phrasing, “How does the following rate on English proficiency and argument coherence?”.

We use paragraphs modified from a set published by the Educational Testing Service (Educational Testing Service, 2005). There are 10 paragraphs, 5 each on 2 different topics. For each topic, the paragraphs have 5 distinct expert scores. Paragraphs are shortened to just a few sentences, and the top rated paragraphs are improved and the worst ones are made worse, preserving the ranking according to the expert scores.

Figure C.1a shows time spent on each page of the experiment, Figure C.1b shows the time spent per paragraph, and Figure C.1c shows the cumulative density function for time spent by workers. The paragraphs are presented to workers in a random order. No workers are excluded in our data and all workers were paid \$1.00, including the ones that spent 2-3 seconds per page. 7/60 workers

¹The data from the experiment is also used for a separate paper, Garg and Johari (2019b). In that work, we analyze the full multi-option question directly, but the main focus is reporting the results of a separate, live trial on a large online labor platform.

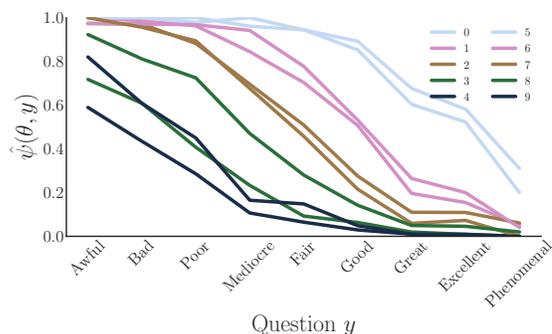


Figure C.2: Paragraph rating distribution – for paragraph θ and rating word y , the empirical $\hat{\psi}(\theta, y)$ is shown. Colors encode the true quality as rated by experts (light blue is best quality, dark blue is worst).

in the pilots received a bonus of \$0.20 for providing feedback. The instructions advised workers to spend no more than a minute per question, though this was not enforced.

The instructions for the main experiment were as follows: “Please rate on English proficiency (grammar, spelling, sentence structure) and coherence of the argument, but not on whether you agree with the substance of the text.” No additional context was provided.

C.1.2 Calculating optimal β and H

Figure C.2 shows the empirical $\hat{\psi}(\theta, y)$ as measured through our experiment. The colors encode the true quality as rated by experts (light blue is best quality, dark blue is worst); recall there are 10 paragraphs with 5 distinct expert ratings (paragraphs 0 and 5 are rated the best, paragraphs 4 and 9 are rated the worst).

With the β calculated and visualized using the methods in Section 4.3, we now find the optimal H for various settings using the methods in Section 4.4. We view our set of paragraphs as *representative items* Θ from a larger universe of paragraphs. In particular, we view our worst quality paragraphs as in the 10th percentile of paragraphs, and our best items as in the 90th percentile. In other words, from the empirical $\hat{\psi}$, we carry out the methods in Section 4.4 using a ψ s.t. $\psi(.1, y) = (\hat{\psi}(4, y) + \hat{\psi}(9, y))/2$ (and similarly for $\psi(.3, y), \psi(.5, y), \psi(.7, y), \psi(.9, y)$, where e.g. $\hat{\psi}(4, y)$ is the empirical rate at which paragraph 4 received a positive rating on question y).

Then, we solve the optimization problem for H stated in Section 4.4. From the above discussion, we want to find an H such that the worst rated paragraphs in our experiment have a probability of receiving a positive rating that is approximately $\beta(.1)$.

Figure C.3 shows the optimal H calculated for various platform settings. These distributions illustrate how often certain binary questions should be asked as it depends on the matching rates and platform objective. For example, as Figure C.3a shows, when there is uniform matching and

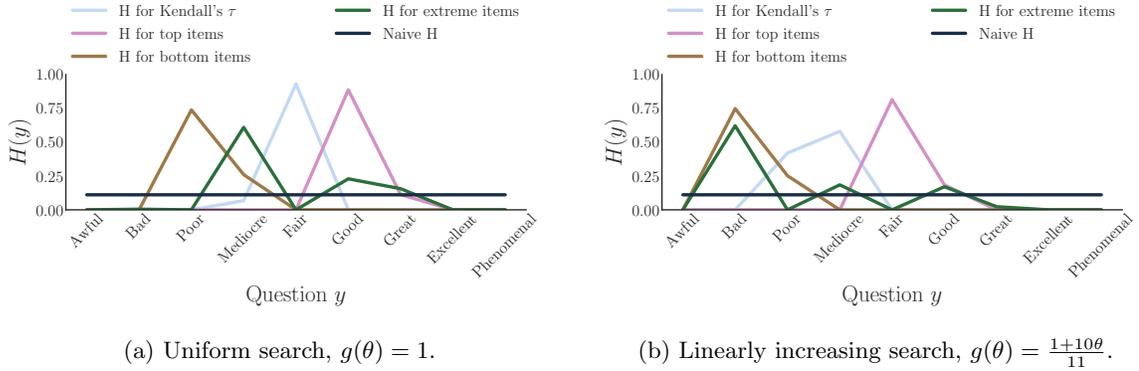


Figure C.3: Optimal $H(y)$ varying by $w(\theta_1, \theta_2)$ using Mechanical Turk data

the platform cares about the entire ranking (i.e. has Kendall’s τ or Spearman’s ρ objective), it should ask most buyers to answer the question, “Would you rate this item as having ‘Fair’ quality or better?”.

Several qualitative insights can be drawn from the optimal H . Most importantly, note that the optimal designs vary significantly with the platform objective and matching rates. In other words, given the same empirical data $\hat{\psi}$, the platform’s design changes substantially based on its goals and how skewed matches are on the platform. Further, note that the differences in H follow from the differences in β that are illustrated in Figure 4.1: when the platform wants to accurately rank the best items, the questions that distinguish amongst the best (e.g., “Would you rate this item as having ‘Good’ quality?”) are drawn more often.

C.1.3 Simulation description

Using the above data and subsequent designs, we simulate markets with a binary rating system as described in Section 4.3.1. Our simulations have the following characteristics.

- 500 items. Items have i.i.d. quality in $[0, 1]$. For item with quality θ , we model buyer rating data using the ψ collected from the experiment as follows. In particular, we presume the items are convex combinations of the representative items in our experiment – items with quality $\theta \in [.1, .3]$ are assumed to have rating probabilities $\psi(\theta, y) = \alpha\psi(.1, y) + (1 - \alpha)\psi(.3, y)$, where $\alpha = (\theta - .1)/.2$. Similarly for θ in other intervals. This process yields the $\tilde{\beta}$ shown in Figure 4.2b.
- In some simulations, all items enter the market at time $k = 0$ and do not leave. In the others, with *entry and exit*, each item independently leaves the market with probability .02 at the end of each time period, and a new item with quality drawn i.i.d. from $[0, 1]$ enters.

- There are 100 buyers, each of which matches to an item independently. In other words, matching is independent across items, and items can match more than once per time period.
- Matching is random with probability as a function of an item’s *estimated* rank $\hat{\theta}$ according to score, rather than actual rank. In other words, the optimal systems were designed assuming item θ would match at rate $g(\theta)$; instead it matches according to $g(\hat{\theta})$, where $\hat{\theta}$ is the item’s rank according to score. We use both $g = 1$ and *linear* search, $g(\hat{\theta}) = \frac{1+10\hat{\theta}}{11}$.
- \mathcal{Y} is the set of 9 adjectives from our MTurk experiments.
- We test several possible H : *naive* with $H(y) = \frac{1}{|\mathcal{Y}|}$, and then the various optimal H calculated for the different sections, illustrated in Figure C.3.

Simulation results

Figure C.4 contains plots from a simulated system that has binary ratings. Figures C.4a, C.4b are with uniform search ($g = 1$), Figures C.4c and C.4d plot the objective prioritizing the worst items, and Figures C.4e and C.4f are with linearly increasing search. For each setting, we include both plots with and without birth/death.

Together, the results suggest that the asymptotic and rate-wise optimality of our calculated β hold even under deviations of the model, and that the real-world design approach outlined in Section 4.4 would provide substantial information benefits to platforms.

Several specific qualitative insights can be drawn from the figures, alongside those discussed in the main text.

1. From all the plots with uniform search, the H designed using our methods for the given setting outperforms other H designs, as expected, and the optimal β (for the given setting) significantly outperforms other designs both *asymptotically* and *rate-wise*.
2. Qualitatively, again with uniform search, heterogeneous item age also does not affect the results. In fact, it seems as if the optimal β and best possible H (given the data) as calculated from our methods outperforms other designs both *asymptotically* and *rate-wise*. Note that this is true even though items entering and leaving the market means that the system may not enter the asymptotics under which our theoretical results hold.
3. Figures C.4c and C.4d show the same system parameters as Figures C.4a, C.4b, i.e. uniform search. However, while C.4a, C.4b show Kendall’s τ correlation over time, C.4c and C.4d show the objective prioritizing bottom items ($w = (1 - \theta_1)(1 - \theta_2)(\theta_1 - \theta_2)$). Note that the β calculated for the actual objective outperforms that calculated for Kendall’s τ , including asymptotically.

Similarly, complementing the fact that H design changes significantly with the weight function, these plots show the value of designing while taking into account one’s true objective value – the

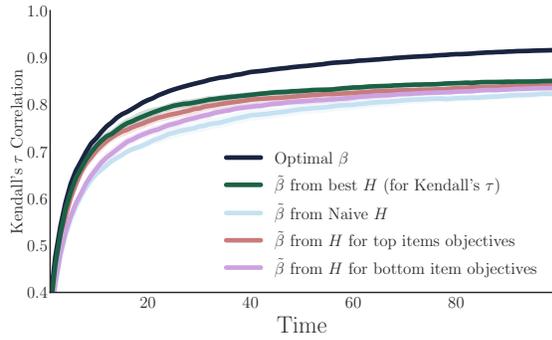
different designs perform differently. Mis-specifying one's objective (e.g. designing to differentiate the best items when one truly cares about the worst items) leads to a large gap in performance (e.g. see the gap between the dark green and red lines in C.4c and C.4d).

Note that comparing the performance of β for the misspecified objective and H for the true objective is not a fair comparison: the former differentiates between all items (though potentially not in a rate-optimal way), while H is constrained by reality, i.e. ψ and \mathcal{Y} .

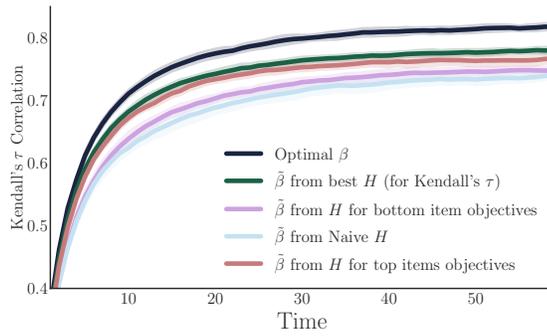
4. Now, consider Figures C.4e and C.4f, which plot the system with linearly increasing search. Note that, contrary to expectation, the optimal β for *uniform* search outperforms the β for the actual system simulated, with linear search! This pattern is especially true for small time k and with item birth/death.

This inversion can be explained as follows. *Uniformization* occurs with heterogeneous age and matching according to observed quality: new items of high type are likely to be mis-ranked lower, while new items of low type are more likely to be mis-ranked higher. (We note that this may not matter in practice, where the search function itself is fit through data, which already captures this effect.) These errors are prominent at low time k and with item birth/death, i.e., in the latter our system never reaches the asymptotics at which the linear β is the optimal design.

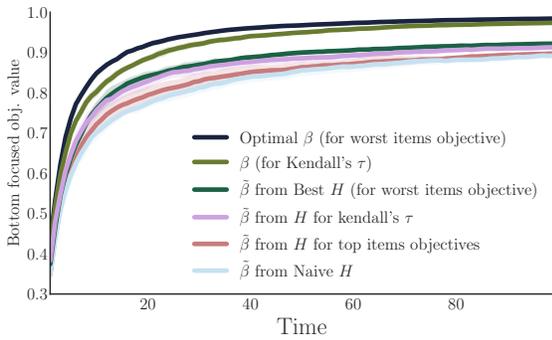
This pattern can be seen more clearly by comparing the two β curves in Figures C.4e, without item birth/death. At small k , when errors are common and so search is more effectively uniform, the β for uniform matching performs the best. However, as such errors subside over time, the performance of the β for linear search catches up and eventually surpasses that of uniform optimal β .



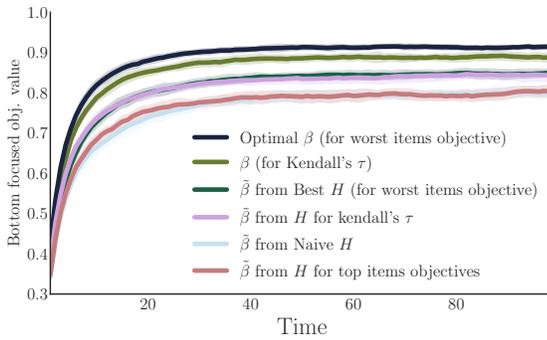
(a) Uniform matching, no birth/death



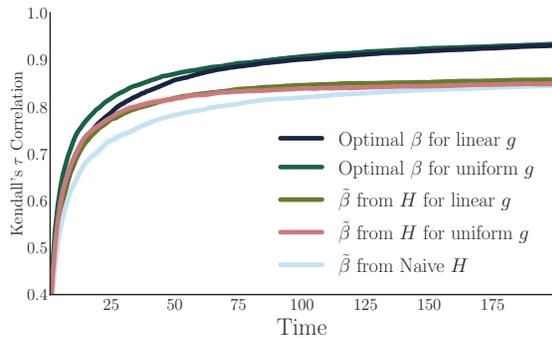
(b) Uniform matching, 2% probability of death per time



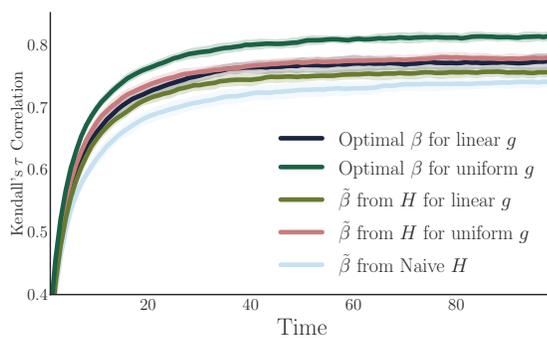
(c) Uniform matching, no birth/death, worst items weighted objective



(d) Uniform matching, 2% birth/death, worst items weighted objective



(e) Linear matching, no birth/death



(f) Linear matching, 2% probability of death per time

Figure C.4: Simulations from data from Mechanical Turk experiment – Binary rating system

C.2 Supplementary theoretical information and results

We now give some additional detail and develop additional results. Section C.2.1 contains the formal specification and update of our deterministic dynamical system. Section C.2.2 gives our algorithm, Nested Bisection, in far more detailed pseudo-code. Section C.2.3 formalizes our earlier qualitative discussion on how matching rates affects the function β . Section C.2.4 includes a convergence result for functions β_M as M increases. Finally, Section C.2.5 contains simple results on how one can learn $\psi(\theta, y)$ through experiments, even if one does not have a reference set of items Θ with known quality before one begins experiments.

C.2.1 Formal specification of system state update

Recall that $\mu_k(\Theta, X)$ is the mass of items with true quality $\theta \in \Theta \subseteq [0, 1]$ and a reputation score $x \in X \subseteq [0, 1]$ at time k . Let $E_k = \{\theta : n_k(\theta) = n_{k-1}(\theta) + 1\}$. These are the items who receive an additional rating at time k ; for all $\theta \in E_k^c$, $n_k(\theta) = n_{k-1}(\theta)$. Our system is completely deterministic, and evolves according to the distributions of the individual seller dynamics.

For each $\theta \in E_k, x, x'$, define $\omega(\theta, x, x')$ as follows:

$$\omega(\theta, x, x') = \beta(\theta)\mathbb{I}\{n_k(\theta)x - n_{k-1}(\theta)x' = 1\} + (1 - \beta(\theta))\mathbb{I}\{n_k(\theta)x - n_{k-1}(\theta)x' = 0\}.$$

Then ω gives the probability of transition from x' to x when an item receives a rating. We then have:

$$\mu_{k+1}(\Theta, X) = \int_{E_k} \int_{x'=0}^1 \int_{x \in X} \omega(\theta, x, x') dx \mu_k(dx', d\theta) + \int_{E_k^c} \int_{x \in X} \mu_k(dx, d\theta).$$

It is straightforward but tedious to check that the preceding dynamics are well defined, given our primitives.

C.2.2 Detailed algorithm

Here, we present the Nested Bisection algorithm, which is described at a high level and summarized in pseudo-code in the main text, in more detail.

ALGORITHM 3: Nested Bisection given in more detail

Data: Set size M , grid width δ , match function g */* Assume $\delta \ll \min_i t_i - t_{i-1}$ */*

Result: β_M levels $\{t_0 \dots t_{M-1}\}$

```

1 Function main ( $M, \delta, g$ )
2    $t_0 = 0, t_{M-1} = 1$ 
3    $\ell = 1 - \frac{1}{M-1}, u = 1 - \delta$ 
4   while  $u - \ell > \delta/2$  do
5      $j_{M-2} = \frac{r+\ell}{2}$ 
6      $\text{rate}_{\text{last}} = -g_{M-2} \log(t_{M-2})$ 
7      $\{j_1 \dots j_{M-3}\} = \text{CalculateOtherLevels}(j_{M-2}, \text{rate}_{\text{last}}, g)$ 
8      $\text{rate}_{\text{first}} = -g_1 \log(1 - t_1)$ 
9     if  $\text{rate}_{\text{first}} < \text{rate}_{\text{last}}$  then  $\ell = j_{M-2}$ 
10    else  $u = j_{M-2}$ 
11     $\{t_1 \dots t_{M-2}\} = \text{CalculateOtherLevels}(u, g)$ 
12     $t_{M-2} = u$ 
13    return  $\{t_i\}$ 
14 Function PairwiseRate ( $t_{m-1}, t_m, g_m, g_{m-1}$ )
15   return  $-(g_{m-1} + g_m) \log \left[ (1 - t_{m-1})^{\frac{g_{m-1}}{g_{m-1}+g_m}} (1 - t_m)^{\frac{g_m}{g_{m-1}+g_m}} + t_{m-1}^{\frac{g_{m-1}}{g_{m-1}+g_m}} t_m^{\frac{g_m}{g_{m-1}+g_m}} \right]$ 
16 Function CalculateOtherLevels ( $j_{M-2}, \text{rate}_{\text{target}}, g$ )
17   /* Given target rate from current guess  $j_{M-2}$ , sequentially fix other levels. */
18   foreach  $m \in M - 3 \dots 1$  do
19      $j_m = \text{BisectNextLevel}(j_{m+1}, \text{rate}_{\text{target}}, g_m, g_{m+1})$ 
20   return  $\{j_1 \dots j_{M-3}\}$ 
21 Function BisectNextLevel ( $j_m, \text{rate}_{\text{target}}, g_{m-1}, g_m$ )
22    $\ell = 0, r = j_m - \delta$ 
23   while  $r - \ell > \delta/2$  do
24      $j_{m-1} = \frac{r+\ell}{2}$ 
25      $\text{rate}_m = \text{PairwiseRate}(j_{m-1}, j_m, g_{m-1}, g_m)$ 
26     if  $\text{rate}_m \leq \text{rate}_{\text{target}}$  then  $r = j_{m-1}$ 
27     else  $\ell = j_{m-1}$ 
28   return  $r$ 

```

C.2.3 Formalization of effect of matching rates shifting

Matching concentrating at the top items moves mass of $\beta(\theta)$ away from high θ , and subsequently mass of $H(y)$ away from the questions that help distinguish the top items, as observed in Figures 4.1b and C.3b above. Informally, this occurs because when matching concentrates, top items are accumulating many ratings more ratings comparatively, and so the amount of information needed per rating is comparatively less. We formalize this intuition in Lemma C.2.1 below.

The lemma states that if matching rates shift such that there is an index k above which matching rates increase and below which they decrease, then correspondingly the levels of β , (i.e. t_i) become closer together above k .

Lemma C.2.1. *Suppose k, g, \tilde{g} such that $\forall j \in \{k+1 \dots M-1\}, g_j \geq \tilde{g}_j$, and $\forall j \in \{0 \dots k-1\}, g_j \leq \tilde{g}_j$, and $g_k = \tilde{g}_k$. Then, $t_k^* \geq \tilde{t}_k^*$.*

Proof. This proof is similar to that of Lemma 4.3.1, except that with the matching function changing the overall rate function can either increase, decrease, or stay the same. Suppose the overall rate function decreased or stayed the same when the matching function changed from \tilde{g} to g . Then $g_{M-2} > \tilde{g}_{M-2}$ and the target rate is no larger, and so $t_{M-2}^* > \tilde{t}_{M-2}^*$. Then, $t_{M-3}^* > \tilde{t}_{M-3}^*$ (a smaller width is needed because the matching rates are higher and the rate is no larger, and the next value also increased). This shifting continues until $t_{k+1}^* > \tilde{t}_{k+1}^*$. Then, $t_k^* > \tilde{t}_k^*$.

Suppose instead that the overall rate function increased when the matching function changed from \tilde{g} to g . Then $g_1 < \tilde{g}_1$ and the target rate is larger, and so $t_1^* > \tilde{t}_1^*$. Then, $t_2^* > \tilde{t}_2^*$ (a larger width is needed and the previous value also increased). This shifting continues until $t_{k-1}^* > \tilde{t}_{k-1}^*$. Then, $t_k^* > \tilde{t}_k^*$. \square

C.2.4 Limit of β as $M \rightarrow \infty$

Let β_M^w denote the optimal β with M intervals for weight function w , with intervals $\{S_i^{wM}\} = \{[s_i^{wM}, s_{i+1}^{wM})\}$ and levels \mathbf{t}^{wM} . Let $q_{wM}(\theta) = i/M$ when $\theta \in [s_i^{wM}, s_{i+1}^{wM})$, i.e. the *quantile* of interval item of type θ is in.

Then, we have the following convergence result for β_M .

Theorem C.2.1. *Let g be uniform. Suppose w such that q_{wM} converges uniformly. Then, $\forall C \in \mathbb{N}, \exists \beta^w$ s.t. $\beta_{C2^N+1}^w \rightarrow \beta^w$ uniformly as $N \rightarrow \infty$.*

The proof is technical and is below. We leverage the fact that, for g uniform, the levels of β_{2M} can be analytically written as a function of the levels of β_M . We believe (numerically observe) that this theorem holds for the entire sequence as opposed to the each such subsequence, and for general matching functions g . However, our proof technique does not carry over, and the proof would leverage more global properties of the optimal β_M .

Furthermore, the condition on w is light. For example, it holds for Kendall’s τ , Spearman’s ρ , and all other examples mentioned in this work.

This convergence result suggests that the choice of M when calculating a asymptotic and rate optimal β is not consequential. As M increases, the limiting value of W_k increases to 1 (i.e. the asymptotic value increases), but the optimal rate decreases to 0. As discussed above, with strictly increasing and continuous β , the asymptotic value is 1 but the large deviations rate does not exist, i.e. convergence is polynomial.

This result could potentially be strengthened as follows: first, show convergence on the entire sequence as opposed to these exponential subsequences, as conjectured; second, show desirable properties of the limiting function itself. It is conceivable but not necessarily true that the limiting function is “better” than other strictly continuous increasing functions in some rate sense, even though the comparison through large deviations rate is degenerate.

C.2.5 Learning $\psi(\theta, y)$ through experiments

Now, we show how a platform would run an experiment to decide to learn $\psi(\theta, y)$. In particular, one potential issue is that the platform does not have any items with know quality that it can use as representative items in its optimization. In this case, we show that it can use ratings within the experiment itself to identify these representative items. The results essentially follow from the law of large numbers.

We assume that $|\Theta| = L$ representative items $i \in \{1 \dots L\}$ are in the experiment, and each are matched N times. The experiment proceeds as follows: every time an item is matched, show the buyer a random question from \mathcal{Y} . For each word $y \in \mathcal{Y}$, track the empirical $\hat{\psi}(i, y)$, the proportion of times a positive response was given to question y . Alternatively, if \mathcal{Y} is totally ordered (i.e. a positive rating for a given y also implies positive ratings would be given to all “easier” y'), and can be phrased as a multiple choice question, data collection can be faster: e.g., as we do in our experiments: \mathcal{Y} consists of a set of totally ordered adjectives that can describe the item; the rater is asked to pick an adjective out of the set; this is interpreted as the item receiving a positive answer to the questions induced by the chosen answer and all worse adjectives, and a negative answer to all better adjectives.

First, suppose the platform approximately knows the quality θ_i of each item i , and θ_i are evenly distributed in $[0, 1]$. Suppose the items are ordered by index, i.e. $\theta_1 < \theta_2 < \dots < \theta_L$. Then let $\hat{\psi}(\theta, y) = \hat{\psi}(i, y)$ when $\theta \in [\theta_{i-1}, \theta_i]$. Call this procedure *KnownTypeExperiment*.

Lemma C.2.2. *Suppose $\psi(\theta, y)$ is Lipschitz continuous in θ . With *KnownTypeExperiment*, $\hat{\psi}(i, y) \rightarrow \psi(\theta_i, y) \forall y$ uniformly as $N \rightarrow \infty$. As $L \rightarrow \infty$, $\hat{\psi}(\theta, y) \rightarrow \psi(\theta, y) \forall \theta$ uniformly.*

Proof. The proof follows directly from the Strong Law of Large Numbers. As $N \rightarrow \infty$, $\forall i$, $\hat{\psi}(i, x) \rightarrow \psi(\theta_i, x)$ uniformly. Now, let $L \rightarrow \infty$. $\forall \epsilon, \exists L'$ s.t. $\forall L > L', \forall \theta, \exists i$ s.t. $|\theta - \theta_i| < \epsilon$. $\psi(\theta, x)$ is Lipschitz

in θ by assumption, and so $\hat{\psi}(\theta, x) \rightarrow \psi(\theta, x)$ uniformly. \square

We now relax the assumption that the platform has an existing set of items with known qualities. Suppose instead the platform has many items L of unknown quality who are expected to match N times each over the experiment time period. For each item, the platform would again ask questions from \mathcal{Y} , drawn according to any distribution (with positive mass on each question). Then generate $\hat{\psi}(\theta, y)$ as follows: first, rank the items according to their ratings during the experiment itself. Then, for each y , $\hat{\psi}(\theta, y)$ is the empirical performance of the θ th percentile item in the ranking, i.e. $\hat{\psi}(\theta, y) = \hat{\psi}(\theta_i, y)$ for $\theta \in [\frac{i-1}{L}, \frac{i}{L}]$. Call this procedure *UnknownTypeExperiment*.

Lemma C.2.3. *Suppose $\psi(\theta, y)$ is Lipschitz continuous in θ . With *UnknownTypeExperiment*, $\hat{\psi}(\theta, y) \rightarrow \psi(\theta, y) \forall y, \theta$ uniformly as $L, N \rightarrow \infty$.*

Proof. Fix L . Denote each item in the experiment as $i \in \{1 \dots L\}$ (with true quality $\theta_i \neq \theta_j$), and each item has N samples. Without loss of generality, assume the items are indexed according to their rank on the average of their scores on the samples, defined as the percentage of positive ratings received. $i = 1$ is then the worst item, and $i = L$ is the best item according to scores in the experiment.

For $\psi(\theta, x)$ increasing in θ , as $N \rightarrow \infty$, $Pr(\theta_i > \theta_j | i < j) \rightarrow 0$ almost surely by SLLN, and for a fixed L , $\{\theta_i\}$ this convergence is uniform. Furthermore, by SLLN, $\hat{\psi}(i, x) \rightarrow \psi(\theta_i, x)$ as $N \rightarrow \infty$. Recall $\hat{\psi}(\theta, x) = \hat{\psi}(i, x)$ for $\theta \in [\frac{i-1}{L}, \frac{i}{L}]$.

Now, let $L \rightarrow \infty$. $\forall \epsilon, \exists L'$ s.t. $\forall L > L', \forall \theta, \exists i$ s.t. $|\theta - \theta_i| < \epsilon$. $\psi(\theta, x)$ is Lipschitz in θ by assumption, and so $\hat{\psi}(\theta, x) \rightarrow \psi(\theta, x)$ uniformly. \square

C.3 Proofs

In this Appendix section, we prove our results.

Sections C.3.1-C.3.3 develop rate functions for P_k and W_k . While rates for P_k follow immediately from large deviation results, the rate function for W_k requires more effort as the quantity is an integral over a continuum of (θ_1, θ_2) , each of which has a rate corresponding to that of $P_k(\theta_1, \theta_2)$.

Then in Section C.3.4 we prove Theorem 4.3.1 and Lemma 4.3.1.

Section C.3.5 then contains additional necessary lemmas required for the proof of the algorithm and convergence result, Theorem C.2.1. The main difficulty for the former is showing a Lipschitz constant in the resulting rate if a level t_i is shifted, which requires lower and upper bounds for t_1 and t_{M-2} , respectively. For the former, we need to relate the solutions of the sequence of optimization problems used to find β_M as M increases. It turns out that both properties follow by relating the levels of β_M to those of β_{2M-1} .

These additional lemmas are used to prove the algorithm approximation bound (Theorem 4.3.2) and the convergence result (Theorem C.2.1) in Section C.3.6 and C.3.7, respectively.

Finally in Section C.3.8 we prove the comments we make in the main text about Kendall's τ and Spearman's ρ rank correlations belonging in our class of objective functions, with asymptotic values of W_k maximized when \mathbf{s} is equispaced in $[0, 1]$.

C.3.1 Rate functions for $P_k(\theta_1, \theta_2)$

Lemma C.3.1.

$$\lim_{k \rightarrow \infty} -\frac{1}{k} \log [\mu((x_k(\theta_1) - x_k(\theta_2)) \leq 0 | \theta_1, \theta_2)] = \inf_{a \in \mathbb{R}} \{g(\theta_1)I(a|\theta_1) + g(\theta_2)I(a|\theta_2)\}$$

where $I(a|\ell) = \sup_z \{za - \Lambda(z|\theta)\}$, and $\Lambda(z|\theta)$ is the log moment generating function of a single sample from $x(\theta_1)$ and $g(\theta)$ is the sampling rate.

Proof. $\lim_{k \rightarrow \infty} -\frac{1}{k} \log [\mu((x_k(\theta_1) - x_k(\theta_2)) \leq 0 | \theta_1, \theta_2)]$

$$= \lim_{k \rightarrow \infty} -\frac{1}{k} \log \left[\int_{a \in \mathbb{R}} \mu((x_k(\theta_1) = a|\theta_1) \mu(x_k(\theta_2) \geq a|\theta_2) da \right] \quad (\text{C.1})$$

$$= \lim_{k \rightarrow \infty} -\frac{1}{k} \log \left[\int_{a \in \mathbb{R}} e^{-kg(\theta_1)I(a|\theta_1)} e^{-kg(\theta_2)I(a|\theta_2)} da \right] \quad (\text{C.2})$$

$$= \inf_{a \in \mathbb{R}} \{g(\theta_1)I(a|\theta_1) + g(\theta_2)I(a|\theta_2)\} \quad \text{Laplace principle} \quad (\text{C.3})$$

□

Where (C.2) is a basic result from large deviations, where $kg(\theta_i)$ is the number of samples item of quality θ_i has received.

Note that this lemma also appears in Glynn and Juneja (2004), which uses the Gartner-Ellis Theorem in the proof. Our proof is conceptually similar but instead uses Laplace's principle.

Recall that $\text{KL}(a||b) = a \log \frac{b}{a} + (1-a) \log \frac{1-b}{1-a}$ is the Kullback-Leibler (KL) divergence between Bernoulli random variables with success probabilities a and b respectively. It is well known that for a Bernoulli random variable with success probability t ,

$$I(a|t) = \text{KL}(a||t)$$

Then, we have

Lemma C.3.2. *Let $\theta_1 > \theta_2$ and $I(a|\theta) = \text{KL}(a||\beta(\theta))$. Further, Let $\bar{P}_k(\theta_1, \theta_2) = 1 - P_k(\theta_1, \theta_2)$. Then,*

$$-\lim_{k \rightarrow \infty} \frac{1}{k} \log \bar{P}_k(\theta_1, \theta_2) = \inf_{a \in \mathbb{R}} \{g(\theta_1)I(a|\theta_1) + g(\theta_2)I(a|\theta_2)\}, \quad (\text{C.4})$$

Proof. Follows directly from Lemma C.3.1.

$$\begin{aligned}
& - \lim_{k \rightarrow \infty} \frac{1}{k} \log \bar{P}_k(\theta_1, \theta_2 | \beta) \\
&= \lim_{k \rightarrow \infty} -\frac{1}{k} \log [1 + \mu_k(x_k(\theta_1) - x_k(\theta_2) < 0 | \theta_1, \theta_2) - \mu_k(x_k(\theta_1) - x_k(\theta_2) > 0 | \theta_1, \theta_2)] \\
&= \lim_{k \rightarrow \infty} -\frac{1}{k} \log [2\mu_k(x_k(\theta_1) - x_k(\theta_2) < 0 | \theta_1, \theta_2) + \mu_k(x_k(\theta_1) - x_k(\theta_2) = 0 | \theta_1, \theta_2)] \\
&= \lim_{k \rightarrow \infty} -\frac{1}{k} \log [\mu_k(x_k(\theta_1) - x_k(\theta_2) \leq 0 | \theta_1, \theta_2)] \\
&= \inf_{a \in \mathbb{R}} \{g(\theta_1)I(a|\theta_1) + g(\theta_2)I(a|\theta_2)\} \tag{Lemma C.3.1}
\end{aligned}$$

□

C.3.2 Laplace's principle with sequence of rate functions

In order to derive a rate function for $\bar{W}_k = (\lim_k W_k) - W_k$, we need to be able to relate its rate to that of $\bar{P}_k(\theta_1, \theta_2)$. The following theorem, related to Laplace's principle for large deviations allows us to do so.

Theorem C.3.1. *Suppose that X is compact with finite Lebesgue measure $\mu(X) < \infty$. Suppose that $\varphi(x)$ has an essential infimum $\underline{\varphi}$ on X , that $\varphi_n(x)$ has an essential infimum $\underline{\varphi}_n$, that both φ and all φ_n are nonnegative, and that $\varphi_n \rightarrow \varphi$ uniformly:*

$$\lim_{n \rightarrow \infty} \sup_{x \in X} |\varphi_n(x) - \varphi(x)| = 0.$$

Then:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log \int_X e^{-n\varphi_n(x)} dx = -\underline{\varphi}. \tag{C.5}$$

Proof. First, we note that for all x and n , $e^{-n\varphi_n(x)} \leq e^{-n\underline{\varphi}_n}$. Therefore, letting $(*)$ denote the LHS of (C.5), we have:

$$(*) \leq \lim_{n \rightarrow \infty} \frac{1}{n} \log \int_X e^{-n\underline{\varphi}_n} dx = -\underline{\varphi},$$

where the last limit follows from the fact that φ_n converges uniformly to φ , so that $\underline{\varphi}_n \rightarrow \underline{\varphi}$.

Next, for $\epsilon > 0$ let $A_n(\epsilon) = \{x : \varphi_n(x) \leq \underline{\varphi}_n + \epsilon\}$ and let $A(\epsilon) = \{x : \varphi(x) \leq \underline{\varphi} + \epsilon\}$. It follows (again by uniform convergence) that for all sufficiently large n , $A(\epsilon/2) \subseteq A_n(\epsilon)$, so that $\mu(A(\epsilon/2)) \leq \mu(A_n(\epsilon))$ for all sufficiently large n . Further, $\mu(A(\epsilon/2)) > 0$, since $\underline{\varphi}$ is the essential infimum.

Since:

$$\int_X e^{-n\varphi_n(x)} dx \geq \mu(A_n(\epsilon)) e^{-n(\underline{\varphi}_n + \epsilon)},$$

it follows that:

$$(*) \geq -\underline{\varphi} - \epsilon + \lim_{n \rightarrow \infty} \frac{1}{n} \log \mu(A_n(\epsilon)).$$

To complete the proof, observe that since $\mu(A_n(\epsilon))$ is bounded below by a positive constant for all sufficiently large n , the last limit is zero. Therefore:

$$(*) \geq -\underline{\varphi} - \epsilon.$$

Since ϵ was arbitrary, this completes the proof. \square

Remark C.3.1. Let $X = [0, 1] \times [0, 1]$, $\varphi_n(\theta_1, \theta_2) = -\frac{1}{n} \log \bar{P}_n(\theta_1, \theta_2)$. Then, all the conditions for Theorem C.3.1 are met.

C.3.3 Rate function for W_k

Our next lemma shows that we can obtain a nontrivial large deviations rate for W_k when β is a step-wise increasing function.

$$\text{Recall } W_k = \int_{\theta_1 > \theta_2} w(\theta_1, \theta_2) P_k(\theta_1, \theta_2 | \beta) d(\theta_1, \theta_2).$$

$$\text{Let } \bar{P}_k(\theta_1, \theta_2) = 1 - P_k(\theta_1, \theta_2).$$

Further, let $\bar{W}_k = (\lim_k W_k) - W_k = \int_{\theta_1 > \theta_2} w(\theta_1, \theta_2) \bar{P}_k(\theta_1, \theta_2 | \beta) d(\theta_1, \theta_2)$. (recall we assumed w integrates to 1 without loss of generality).

Lemma C.3.3. Suppose β is piecewise constant with M levels $\{t_i\}$. Let $g_i \triangleq \inf_{\theta \in S_i} g(\theta) = g(s_i)$. Then,

$$-\lim_{k \rightarrow \infty} \frac{1}{k} \log \bar{W}_k = \min_{0 \leq i \leq M-2} \inf_{a \in \mathbb{R}} \{g_{i+1} I(a|t_{i+1}) + g_i I(a|t_i)\} \triangleq r, \quad (\text{C.6})$$

where $I(a|t) = KL(a||t)$ as defined in Lemma C.3.2.

Proof. When β is step-wise increasing with M levels $\{t_i\}$, then

$$\bar{W}_k = \sum_{0 \leq i < j < M} \int_{\theta_2 \in S_i, \theta_1 \in S_j} w(\theta_1, \theta_2) \bar{P}_k(\theta_1, \theta_2 | \beta) d(\theta_1, \theta_2)$$

as $\bar{P}_k(\theta_1, \theta_2) = 0$ when $\beta(\theta_1) = \beta(\theta_2)$.

$$\begin{aligned}
& -\lim_{k \rightarrow \infty} \frac{1}{k} \log \bar{W}_k \\
&= -\lim_{k \rightarrow \infty} \frac{1}{k} \log \int_{\theta_1 > \theta_2} w(\theta_1, \theta_2) \bar{P}_k(\theta_1, \theta_2 | \beta) d(\theta_1, \theta_2) \tag{C.7}
\end{aligned}$$

$$= -\lim_{k \rightarrow \infty} \frac{1}{k} \log \sum_{0 \leq i < j < M} \int_{\theta_2 \in S_i, \theta_1 \in S_j} w(\theta_1, \theta_2) \bar{P}_k(\theta_1, \theta_2 | \beta) d(\theta_1, \theta_2) \tag{C.8}$$

$$= -\max_{0 \leq i < j < M} \left(\lim_{k \rightarrow \infty} \frac{1}{k} \left[\log \int_{\theta_2 \in S_i, \theta_1 \in S_j} w(\theta_1, \theta_2) \bar{P}_k(\theta_j, \theta_i | \beta) d(\theta_1, \theta_2) \right] \right) \tag{C.9}$$

$$= -\max_{0 \leq i < j < M} \sup_{\theta_1 \in S_j, \theta_2 \in S_i} \left(\lim_{k \rightarrow \infty} \frac{1}{k} [\log w(\theta_1, \theta_2) \bar{P}_k(\theta_j, \theta_i | \beta)] \right) \tag{C.10}$$

$$= -\max_{0 \leq i < j < M} \sup_{\theta_1 \in S_j, \theta_2 \in S_i} \left(\lim_{k \rightarrow \infty} \frac{1}{k} \log \bar{P}_k(\theta_j, \theta_i | \beta) \right) \tag{C.11}$$

$$= \min_{0 \leq i < j < M} \inf_{\theta_1 \in S_j, \theta_2 \in S_i} \left(-\lim_{k \rightarrow \infty} \frac{1}{k} \log \bar{P}_k(\theta_j, \theta_i | \beta) \right) \tag{C.12}$$

$$= \min_{0 \leq i < j < M} \inf_{\theta_1 \in S_j, \theta_2 \in S_i} \inf_{a \in \mathbb{R}} \{g(\theta_1)I(a|t_j) + g(\theta_2)I(a|t_i)\} \tag{C.13}$$

$$= \min_{0 \leq i < j < M} \inf_{a \in \mathbb{R}} \{g_j I(a|t_j) + g_i I(a|t_i)\} \tag{C.14}$$

$$= \min_{0 \leq i < M-1} \inf_{a \in \mathbb{R}} \{g_{i+1} I(a|t_{i+1}) + g_i I(a|t_i)\} \tag{C.15}$$

The last line follows from adjacent t_i, t_{i+1} dominating the rate due to monotonicity properties.

Line (C.10) follows from Theorem C.3.1.

Line (C.9) follows from: $\forall a_i^\epsilon \geq 0, \limsup_{\epsilon \rightarrow 0} \left[\epsilon \log \left(\sum_i^N a_i^\epsilon \right) \right] = \max_i^N \limsup_{\epsilon \rightarrow 0} \epsilon \log(a_i^\epsilon)$, which is a finite case version (with fewer assumptions) of Theorem C.3.1. See, e.g., Lemma 1.2.15 in (Dembo and Zeitouni, 2010) for a proof of this property.

□

Lemma C.3.4. $\beta(\theta)$ is piecewise constant $\iff \exists c(\beta) > 0$ s.t. $-\lim_{k \rightarrow \infty} \frac{1}{k} \log(\bar{W}_k) = c(\beta)$.

Proof. \implies follows directly from Lemma C.3.3: $\inf_{a \in \mathbb{R}} \{g_{i+1} I(a|t_{i+1}) + g_i I(a|t_i)\} > 0$ when $t_i \neq t_{i+1}$, which holds when β is piece-wise constant with the appropriate number of levels.

\impliedby Consider β that is not piece-wise constant. Recall that we further assume that β is non-decreasing, and discontinuous only on a measure 0 set. Following algebra steps similar to those in Lemma C.3.3, but for general β :

$$-\lim_{k \rightarrow \infty} \frac{1}{k} \log \bar{W}_k = -\lim_{k \rightarrow \infty} \frac{1}{k} \log \int_{\theta_1 > \theta_2} w(\theta_1, \theta_2) \bar{P}_k(\theta_1, \theta_2 | \beta) d(\theta_1, \theta_2) \quad (\text{C.16})$$

$$= \inf_{\theta_1 > \theta_2} \left(-\lim_{k \rightarrow \infty} \frac{1}{k} \log \bar{P}_k(\theta_j, \theta_i | \beta) \right) \quad (\text{C.17})$$

$$= 0 \quad (\text{C.18})$$

Where the last line follows from β continuous at some θ_1 , and so $\lim_{\theta_2 \rightarrow \theta_1} \bar{P}_k(\theta_1, \theta_2 | \beta) = 1$.

Intuitively, what goes wrong with continuous β is that $\bar{P}_k(\theta_1, \theta_2 | \beta)$ does not converge uniformly:

$$\forall \epsilon, k, \exists \theta_2 \neq \theta_1 \quad \bar{P}_k(\theta_1, \theta_2) > \epsilon$$

i.e. close by items are very hard to distinguish from one another. Then, because the large deviations rate of \bar{W}_k is dominated by the worst rates under the integral, we don't get a positive rate.

□

C.3.4 Proofs of Lemma 4.3.1 and Theorem 4.3.1

Remark C.3.2. *The KL divergence for two Bernoulli random variables is continuous and strictly convex, with minima at $a = b$, when $a, b \in (0, 1)$. Note that $\inf_a \{g_i KL(a || t_i) + g(i+1) KL(a || t_{i+1})\}$, for all feasible g , is also continuous and strictly convex in t_i, t_{i+1} , with minima at $t_i = t_{i+1}$.*

One consequence of the above fact is that fixing either t_i or t_{i+1} and moving the other farther away monotonically increases KL, while moving it closer decreases KL.

Proof of Theorem 4.3.1

Proof. We use the same notation as the proof for Lemma C.3.3.

Part 1.

$$\lim_{k \rightarrow \infty} W_k = \lim_{k \rightarrow \infty} \sum_{0 \leq i < j < M} \left[\int_{\theta_2 \in S_i, \theta_1 \in S_j} w(\theta_1, \theta_2) P_k(\theta_1, \theta_2 | \beta) d(\theta_1, \theta_2) \right] \quad (\text{C.19})$$

$$= \sum_{0 \leq i < j < M} \int_{\theta_2 \in S_i, \theta_1 \in S_j} w(\theta_1, \theta_2) d(\theta_1, \theta_2) \quad (\text{C.20})$$

(C.20) follows from bounded convergence and $P_k(\theta_1, \theta_2 | \beta) \rightarrow 1$ for $\theta_1 \in S_j, \theta_2 \notin S_j$. Thus choosing s to maximize (C.20) maximizes the asymptotic value of W_k .

Part 2. Follows directly from Lemma C.3.3. □

Proof of Lemma 4.3.1

Proof. Recall $r(\mathbf{t}) \triangleq -\lim_{k \rightarrow \infty} \frac{1}{k} \log(W - W_k) = \min_{0 \leq i \leq M-2} \inf_{a \in \mathbb{R}} \{g_{i+1} \text{KL}(a||t_{i+1}) + g_i \text{KL}(a||t_i)\}$.

We show the following:

$r(\mathbf{t}) =$

$$\min \left(\log(1 - t_1)^{-g_1}, \right. \\ \left. \log \left[(1 - t_{i-1})^{\frac{g_{i-1}}{g_{i-1}+g_i}} (1 - t_i)^{\frac{g_i}{g_{i-1}+g_i}} + t_{i-1}^{\frac{g_{i-1}}{g_{i-1}+g_i}} t_i^{\frac{g_i}{g_{i-1}+g_i}} \right]^{-g_{i-1}-g_i} \text{ for } 1 < i < M - 1, \right. \\ \left. \log(t_{M-2})^{-g_{M-2}} \right)$$

and \mathbf{t}^* maximizes $r_w(\mathbf{t}) \iff$ all the terms inside the minimization $r_w(\mathbf{t}^*)$ are equal. Further, the optimal levels \mathbf{t}^* are unique. The result immediately follows, that $\{t_i\}$ is the unique solution that equalizes the rates inside the minimization, by noting that the optimal r has $t_0 = 0, t_{M-1} = 1$.

We first prove the alternative form for r . Note that $\{g_{i-1} \text{KL}(a||t_{i-1}) + g_i \text{KL}(a||t_i)\}$ is convex in a , and so we can find an analytic form for the infimum over a .

$$\text{Let } a_i = \arg \inf_{a \in [t_{i-1}, t_i]} \{g_{i-1} \text{KL}(a||t_{i-1}) + g_i \text{KL}(a||t_i)\}$$

$$\begin{aligned} &\implies \nabla_{a_i} [g_{i-1} \text{KL}(a_i||t_{i-1}) + g_i \text{KL}(a_i||t_i)] = 0 \\ &\implies \nabla_{a_i} \left[g_{i-1} \left(a_i \log \frac{a_i}{t_{i-1}} + (1 - a_i) \log \frac{1 - a_i}{1 - t_{i-1}} \right) + g_i \left(a_i \log \frac{a_i}{t_i} + (1 - a_i) \log \frac{1 - a_i}{1 - t_i} \right) \right] = 0 \\ &\implies g_{i-1} \left(\log \frac{a_i}{t_{i-1}} - \log \frac{1 - a_i}{1 - t_{i-1}} \right) + g_i \left(\log \frac{a_i}{t_i} - \log \frac{1 - a_i}{1 - t_i} \right) = 0 \\ &\implies \log \left(\frac{a_i}{1 - a_i} \right)^{g_{i-1}+g_i} = \log \left(\frac{t_{i-1}}{1 - t_{i-1}} \right)^{g_{i-1}} + \log \left(\frac{t_i}{1 - t_i} \right)^{g_i} \\ &\implies \frac{a_i}{1 - a_i} = \left[\left(\frac{t_{i-1}}{1 - t_{i-1}} \right)^{g_{i-1}} \left(\frac{t_i}{1 - t_i} \right)^{g_i} \right]^{\frac{1}{g_{i-1}+g_i}} \\ &\implies a_i = \frac{c}{1 + c}, \text{ where } c = \left[\left(\frac{t_{i-1}}{1 - t_{i-1}} \right)^{g_{i-1}} \left(\frac{t_i}{1 - t_i} \right)^{g_i} \right]^{\frac{1}{g_{i-1}+g_i}} \end{aligned}$$

Then,

$$g_{i-1}\text{KL}(a_i||t_{i-1}) + g_i\text{KL}(a_i||t_i)$$

$$\begin{aligned} &= g_{i-1}a \log \frac{a}{t_{i-1}} + g_i a \log \frac{a}{t_i} + g_{i-1}(1-a) \log \frac{1-a}{1-t_{i-1}} + g_i(1-a) \log \frac{1-a}{1-t_i} \\ &= a \left[(g_{i-1} + g_i) \log \frac{a}{1-a} + g_{i-1} \log \frac{1-t_{i-1}}{t_{i-1}} + g_i \log \frac{1-t_i}{t_i} \right] + \log(1-a)^{g_{i-1}+g_i} - \log(1-t_{i-1})^{g_{i-1}}(1-t_i)^{g_i} \\ &= (g_{i-1} + g_i) \log(1-a) - \log(1-t_{i-1})^{g_{i-1}}(1-t_i)^{g_i} \end{aligned} \quad (\text{C.21})$$

$$\begin{aligned} &= -(g_{i-1} + g_i) \log \left[1 + \left[\left(\frac{t_{i-1}}{1-t_{i-1}} \right)^{g_{i-1}} \left(\frac{t_i}{1-t_i} \right)^{g_i} \right]^{\frac{1}{g_{i-1}+g_i}} \right] (1-t_{i-1})^{\frac{g_{i-1}}{g_{i-1}+g_i}} (1-t_i)^{\frac{g_i}{g_{i-1}+g_i}} \\ &= -(g_{i-1} + g_i) \log \left[(1-t_{i-1})^{\frac{g_{i-1}}{g_{i-1}+g_i}} (1-t_i)^{\frac{g_i}{g_{i-1}+g_i}} + t_{i-1}^{\frac{g_{i-1}}{g_{i-1}+g_i}} t_i^{\frac{g_i}{g_{i-1}+g_i}} \right] \end{aligned} \quad (\text{C.22})$$

Where line (C.21) uses $\frac{a}{1-a} = c$ and $(g_{i-1} + g_i) \log c = \log \left[\left(\frac{t_{i-1}}{1-t_{i-1}} \right)^{g_{i-1}} \left(\frac{t_i}{1-t_i} \right)^{g_i} \right]$. Note that the first and last rates emerge, respectively, by plugging in $t_0 = 0, t_M = 1$, which holds trivially at the optimum from monotonicity.

We note that a similar derivation, of the large deviation rate for two binomial distributions with different probability of successes and match rates, appears in Glynn and Juneja (2004). In that work, the authors seek to optimize the g in order to identify the single best item out of a set of possible items, and a concave program emerges. In this work, because we optimize the probability of successes and care about retrieving a ranking of the items, no such concave or convex program emerges.

Now we show that \mathbf{t}^* maximizes $r_w(\mathbf{t}) \iff$ all the terms inside the minimization $r_w(\mathbf{t})$ are equal.

equalizes \implies optimal. Let $r(i)$ be the i th term in the minimization, starting at $i = 1$. Note that (holding the other fixed) increasing t_i increases the i th term monotonically and decreases the $(i + 1)$ th term monotonically. Suppose β s.t. $r(i) = r(j) \forall i, j$. To increase the minimization term, one must increase $r(i), \forall i$. To increase $r(1)$, t_1 must increase, regardless of what the other levels are. Then, to increase $r(2)$, t_2 must increase \dots to increase $r(M - 2)$, t_{M-2} must increase. However, to increase $r(M - 1)$, t_{M-2} must decrease, and we have a contradiction. Thus, one cannot increase all terms simultaneously.

equalizes \Leftarrow optimal. Suppose \mathbf{t} maximizes $r(\mathbf{t})$ but the terms inside the minimization are not equal. Then $\exists i$ s.t. $r(i) = \min_j r(j)$ and either $r(i) \neq r(i - 1)$ or $r(i) \neq r(i + 1)$. $r(i)$ can be increased without lowering the overall rate. This method can be repeated $\forall i : r(i) = \min_j r(j)$ and so \mathbf{t} would not be optimal, a contradiction.

Uniqueness follows from the overall rate unique determining t_1, t_{M-2} and so iteratively uniquely determining the rest. \square

C.3.5 Additional necessary lemmas

Now, we begin the set-up that will lead to a proof for Theorem 4.3.2. It turns out that proving the theorem requires, in the process, essentially proving our convergence result with $M \rightarrow \infty$, Theorem C.2.1. For Theorem 4.3.2, we need a lower bound for t_1 as a function of M . This seems hard to do in general. Luckily, in our case, there is a property for how \mathbf{t}^* changes when M is doubled. Using this property, we can derive that $t_1^* \geq \mathcal{O}(M^{-3})$.

Recall that step-wise increasing β with M intervals $S_i = [s_i, s_{i+1})$ has levels $\{t_i\}_{i=0}^{M-1}$, where $t_0 = 1, t_{M-1} = 1$, and $s_0 \triangleq 0, s_M \triangleq 1$.

Furthermore, we use the following notation for the large deviation rate

$$r_i = -(g_{i-1} + g_i) \log \left[(1 - t_{i-1})^{\frac{g_{i-1}}{g_{i-1}+g_i}} (1 - t_i)^{\frac{g_i}{g_{i-1}+g_i}} + t_{i-1}^{\frac{g_{i-1}}{g_{i-1}+g_i}} t_i^{\frac{g_i}{g_{i-1}+g_i}} \right] \quad (\text{C.23})$$

for $i \in \{1 \dots M-1\}$, which implies $r_1 = -g_1 \log(1 - t_1)$ and $r_{M-1} = -g_{M-2} \log(t_{M-2})$.

We further use r^{M-1} to be the rate achieved by the optimal β_M with M intervals.

Lemma C.3.5. *Suppose g uniform, i.e. $g_i = 1, \forall i$ and that β_M has values $\{t_i\}_{i=0}^{M-1}$. Then β_{2M-1} has values $\{t'_i\}_{i=0}^{2M-2}$, where $t'_{2i} = t_i, \forall i \in \{0 \dots M-1\}$, $t'_1 = \frac{1}{2}(1 - \sqrt{1 - t_1})$ and $t'_{2M-3} = \frac{1}{2}(1 + \sqrt{t_{M-2}})$.*

Proof. We first set the values $t'_{2i} = t_i$ and then optimally choose the remaining values t'_k, k odd. Then, we show that the resulting large deviation rates between all adjacent pairs are equal. Then, by the proof of Lemma 4.3.1, which showed that equalizing the rates between adjacent intervals is a sufficient condition for optimality, β_{2M-1} has the levels $\{t'_i\}_{i=0}^{2M-2}$.

Let r' denote rates between adjacent t' as r does for t . Supposing $t'_2 = t_1$, we find t'_1 such that $r'_1 = r'_2$ and $t'_1 < t'_2$.

$$\begin{aligned} -\log(1 - t'_1) &= -2 \log \left[\sqrt{(1 - t'_1)(1 - t'_2)} + \sqrt{t'_1 t'_2} \right] \\ \implies 1 - t'_1 &= (1 - t'_1)(1 - t'_2) + t'_1 t'_2 + 2\sqrt{(1 - t'_1)(1 - t'_2)t'_1 t'_2} \\ \implies t'_1 &= \frac{1}{2} \left(1 - \sqrt{1 - t'_2} \right) = \frac{1}{2} \left(1 - \sqrt{1 - t_1} \right) \end{aligned}$$

Similarly, $r'_{2M-3} = r'_{2M-2}$ when $t'_{2M-3} = \frac{1}{2}(1 + \sqrt{t_{M-2}})$. It follows that $r'_1 = r'_2 = r'_{2M-3} = r'_{2M-2}$ by choosing such t'_1, t'_{2M-3} .

Next, we find $t'_k \in (t'_{k-1}, t'_{k+1})$ for $k \in \{3, 5, \dots, 2M-5\}$ such that the rates $r'_k = r'_{k+1}$.

$$\begin{aligned} -2 \log \left[\sqrt{(1-t'_k)(1-t'_{k-1})} + \sqrt{t'_k t'_{k-1}} \right] &= -2 \log \left[\sqrt{(1-t'_k)(1-t'_{k+1})} + \sqrt{t'_k t'_{k+1}} \right] \\ \implies t'_k &= \frac{c}{1+c}, \text{ where } c = \left[\frac{\sqrt{1-t'_{k+1}} - \sqrt{1-t'_{k-1}}}{\sqrt{t'_{k-1}} - \sqrt{t'_{k+1}}} \right]^2 \end{aligned}$$

Now, we show that $r'_k = r'_j, \forall j, k$ by showing that the difference between each rate r_i and its analogous rate r'_{2i} is constant. $r_k = r_j, \forall j, k$ by assumption and so $r'_k = r'_j, \forall j, k$ follows.

$r_{M-1} = -\log t_{M-2}$ and $r'_{2M-2} = -\log \frac{1}{2} (1 + \sqrt{t_{M-2}})$. Thus if $r_i = -\log x$ for some x , then $r_{2i} = -\log \frac{1}{2} (1 + \sqrt{x})$ would imply that all the rates are equal. Thus, it is sufficient to show that

$$\left[\sqrt{(1-t'_{2i-1})(1-t'_{2i})} + \sqrt{t'_{2i-1} t'_{2i}} \right]^2 = \frac{1}{2} \left[1 + \sqrt{(1-t_{i-1})(1-t_i)} + \sqrt{t_{i-1} t_i} \right] \quad (\text{C.24})$$

$$\equiv \left[\sqrt{\left(1 - \frac{c}{1+c}\right)(1-t_i)} + \sqrt{\frac{c}{1+c} t_i} \right]^2 = \frac{1}{2} \left[1 + \sqrt{(1-t_{i-1})(1-t_i)} + \sqrt{t_{i-1} t_i} \right] \quad (\text{C.25})$$

$$\text{where } c = \left[\frac{\sqrt{1-t_i} - \sqrt{1-t_{i-1}}}{\sqrt{t_{i-1}} - \sqrt{t_i}} \right]^2$$

The proof for (C.25) is algebraically tedious and is shown in Remark C.3.3 below.

Then, by the proof of Lemma 4.3.1, which shows that equalizing the rates inside the minimization terms implies an optimal $\{t_i\}$, β_{2M-1} has the levels $\{t'_i\}_{i=0}^{2M-2}$.

□

Remark C.3.3.

$$\begin{aligned} \left[\sqrt{\left(1 - \frac{c}{1+c}\right)(1-t_i)} + \sqrt{\frac{c}{1+c} t_i} \right]^2 &= \frac{1}{2} \left[1 + \sqrt{(1-t_{i-1})(1-t_i)} + \sqrt{t_{i-1} t_i} \right] \\ \text{where } c &= \left[\frac{\sqrt{1-t_i} - \sqrt{1-t_{i-1}}}{\sqrt{t_{i-1}} - \sqrt{t_i}} \right]^2 \end{aligned}$$

Proof. Let $x = \sqrt{t_i}, y = \sqrt{1-t_i}, z = \sqrt{t_{i-1}}$, and $w = \sqrt{1-t_{i-1}}$. Note that $x > z, w > y, y = 1 - x^2, w = 1 - z^2$. Then,

$$\frac{c}{c+1} = \frac{(y-w)^2}{2-2xz-2yw}, \text{ and } \frac{1}{c+1} = \frac{(x-z)^2}{2-2xz-2yw}$$

(To show the above two equalities, factor out $\frac{1}{(x-z)^2}$ from numerator and denominator, and substitute $y = 1 - x^2, w = 1 - z^2$).

Now, the left hand side:

$$\begin{aligned}
& \left[\sqrt{\left(1 - \frac{c}{1+c}\right)(1-t_i)} + \sqrt{\frac{c}{1+c}t_i} \right]^2 \\
&= \frac{1}{2-2xz-2yw} \left[\sqrt{(x-z)^2y^2} + \sqrt{(y-w)^2x^2} \right]^2 \\
&= \frac{(x-z)^2y^2 + (y-w)^2x^2 + 2xy(x-z)(w-y)}{2-2xz-2yw} \quad \sqrt{(y-w)^2} = w-y, \sqrt{(x-z)^2} = x-z \\
&= \frac{z^2y^2 + w^2x^2 - 2wxyz}{2-2xz-2yw}
\end{aligned}$$

The right hand side:

$$\begin{aligned}
& \frac{1}{2} \left[1 + \sqrt{(1-t_{i-1})(1-t_i)} + \sqrt{t_{i-1}t_i} \right] \\
&= \frac{1}{2} [1 + (wy + xz)]
\end{aligned}$$

Multiplying both sides by $2 - 2xz - 2yw$, we have:

$$\begin{aligned}
& \left[\sqrt{\left(1 - \frac{c}{1+c}\right)(1-t_i)} + \sqrt{\frac{c}{1+c}t_i} \right]^2 = \frac{1}{2} \left[1 + \sqrt{(1-t_{i-1})(1-t_i)} + \sqrt{t_{i-1}t_i} \right] \\
&\quad \equiv z^2y^2 + w^2x^2 - 2wxyz = 1 - (wy + xz)^2 \\
&\quad \equiv z^2(1-x^2) + (1-z^2)x^2 - 2wxyz = 1 - w^2y^2 - x^2z^2 - 2wxyz \\
&\quad \equiv z^2 - 2x^2z^2 + x^2 = 1 - (1-z^2)(1-x^2) - x^2z^2 \\
&\quad \equiv 0 = 0
\end{aligned}$$

□

Corollary C.3.1. *Suppose g uniform, i.e. $g_i = 1, \forall i$. $\forall \epsilon > 0, \exists M$ s.t. $\forall M' \geq M, r^{M'} < \epsilon$.*

Proof. Let $M = 2^N, M' = 2^{N+1} - 1$, for some N . We show that $r^{M'} \leq \frac{1}{2}r^M$. The corollary follows by noting that $r^{K'} < r^K \forall K' > K$ and that $r^K < \infty, \forall K$.

$$\begin{aligned}
r^M - r^{M'} &= -\log t_{M-2}^M + \log t_{M'-2}^{M'} \\
&= -\log t_{M-2}^M + \log \left[\frac{1}{2} + \frac{1}{2} \sqrt{t_{M-2}^M} \right] && \text{Lemma C.3.5} \\
&= \log \left[\frac{1}{2} \frac{1}{t_{M-2}^M} + \frac{1}{2} \frac{1}{\sqrt{t_{M-2}^M}} \right] \\
&\geq -\frac{1}{2} \log t_{M-2}^M && \sqrt{t_{M-2}^M} \geq t_{M-2}^M \\
\implies r^{M'} &\leq \frac{1}{2} r^M
\end{aligned}$$

□

Corollary C.3.2. *Suppose g uniform, i.e. $g_i = 1, \forall i$. $\forall \delta > 0, \exists N$ s.t. $\forall M \geq N, \max_k t_k^M - t_{k-1}^M < \delta$.*

Proof. This corollary follows directly from Corollary C.3.1. If the rates are upper bounded, then so are the level differences.

We first find where the rate is minimized given a width between levels of δ

$$\begin{aligned}
x_m &= \arg \min_x -2 \log \left[\sqrt{(1-x-\delta)(1-x)} + \sqrt{x(x+\delta)} \right] \\
&= \frac{1}{2} - \frac{1}{2} \delta
\end{aligned}$$

Then given an upper bound of ϵ on the rate, there is a bound on δ determined by the largest possible difference at levels symmetric around $\frac{1}{2}$.

$$\begin{aligned}
r^L &= -2 \log \left[2 \sqrt{\left(\frac{1}{2} - \delta\right) \left(\frac{1}{2} + \delta\right)} \right] \\
&= -\log [1 - 4\delta^2] \\
&\geq \epsilon \text{ when } \delta > \frac{1}{2} \sqrt{1 - e^{-\epsilon}}
\end{aligned}$$

□

Lemma C.3.6. *Suppose g is non-decreasing in θ . Then, $t_{M-2} \geq 1 - \frac{1}{M-1}$.*

Proof. Note that, with uniform matching, $\forall x \in (0, 1], y \in [0, 1-x]$ the rate with values $t_{i-1} = y, t_i = y+x$ is no more than the last with $t_{M-2} = 1-x$. With width x , in other words, the extreme points

have a larger rates than the middle points. For $i \notin \{1, M-1\}$:

$$\begin{aligned}
r_i &= \inf_a \{g_{i-1} \text{KL}(a||t_{i-1}) + g_i \text{KL}(a||t_i)\} \\
&= \inf_a \{\text{KL}(a||y) + \text{KL}(a||y+x)\} && \text{uniform matching} \\
&= -2 \log \left[(1-y)^{\frac{1}{2}}(1-y-x)^{\frac{1}{2}} + y^{\frac{1}{2}}(y+x)^{\frac{1}{2}} \right] && \text{(C.26)} \\
&= -\log \left[(1-y)(1-y-x) + y(y+x) + 2[(1-y)(1-y-x)y(y+x)]^{1/2} \right] \\
&\leq -\log(1-x)
\end{aligned}$$

Where line (C.26) follows from line (C.22).

By the proof of Lemma 4.3.1, the optimal levels equalize the rates between each level. Then, when g is non-decreasing, $g_{M-2} \geq g_\ell, \forall \ell \in \{1 \dots M-3\}$. Then, at the same level differences, the rate corresponding to the last level is no smaller. Thus, to equalize the rates, the last width must be no larger than any other width. Thus, $t_{M-2} \geq 1 - \frac{1}{L}$. \square

Lemma C.3.7. *With uniform matching ($g_i = 1$), $r^{2^{N+1}-1} \geq \frac{1}{5}r^{2^N}$.*

Proof. Let $K = 2^N, K' = 2^{N+1} - 1$. Note that $t_{K-1}^K \geq \frac{1}{2}$ by Lemma C.3.6.

$$\begin{aligned}
r^K - r^{K'} &= -\log t_{K-2}^K + \log t_{K'-2}^{K'} \\
&= -\log t_{K-2}^K + \log \left[\frac{1}{2} + \frac{1}{2} \sqrt{t_{K-2}^K} \right] && \text{Lemma C.3.5} \\
&= \log \left[\frac{1}{2} \frac{1}{t_{K-2}^K} + \frac{1}{2} \frac{1}{\sqrt{t_{K-2}^K}} \right] \\
&\leq \log (t_{K-2}^K)^{-\frac{4}{5}} && \frac{1}{2} (t_{K-2}^K)^{-1} + \frac{1}{2} (t_{K-2}^K)^{-\frac{1}{2}} \leq (t_{K-2}^K)^{-\frac{4}{5}} \text{ when } t_{K-2}^K \in \left[\frac{1}{2}, 1 \right] \\
\implies r^{K'} &\geq \frac{1}{5} r^K
\end{aligned}$$

\square

Lemma C.3.8. *With uniform matching ($g_i = 1$), $\exists C > 0$ s.t. $\forall M, t_1^M \geq CM^{-3}$.*

Proof. By Lemma C.3.7, $\exists C_2 > 0$ s.t. $r^M \geq C_2 5^{-\lceil \log_2 M \rceil}$. Then

$$\begin{aligned}
-\log(1 - t_1^M) &= r^M \\
&\geq C_2 5^{-\lceil \log_2 M \rceil} \\
\implies t_1^M &\geq 1 - \exp\left[-C_2 5^{-\lceil \log_2 M \rceil}\right] \\
&\geq 1 - \exp\left[-C_3 M^{-\frac{1}{\log_5 2}}\right] \\
&\geq \frac{e-1}{e} C_3 M^{-\frac{1}{\log_5 2}} & e^{-x} \leq 1 - \frac{e-1}{e}x \text{ for } x \in [0, 1] \\
\implies \exists C > 0 \text{ s.t. } t_1^M &\geq CM^{-3}
\end{aligned}$$

□

Corollary C.3.3. *With monotonically non-decreasing g , $\exists C > 0$ s.t. $\forall M, t_1^M \geq CM^{-3}$.*

Proof. The result follows from noting that t_1^M with uniform matching lower bounds the first value with any other monotonically non-decreasing g , which is a direct application of Lemma C.2.1 – scale g such that $g_1 = 1$. Then, $g_j \geq 1, j > 1$ and $g_0 \leq 1$. Then, the condition of the lemma holds. □

Lemma C.3.9. *The run-time of `NestedBisection` is $O(M \log^2 \frac{1}{\delta})$, where δ is the bisection grid width and M is the number of intervals.*

Proof. The outer bisection, in *main*, runs at most $\log_2 \frac{2}{\delta} + 1$ iterations. Each outer iteration calls `BisectNextLevel` $M - 3$ times, and the inner bisection in each call runs for at most $\log_2 \frac{2}{\delta}$ iterations. Thus the run-time of algorithm is $O(M \log^2 \frac{1}{\delta})$. □

C.3.6 Proof for Theorem 4.3.2

Finally, we are ready to prove Theorem 4.3.2. It follows from formalizing the relationship between δ , the bisection grid width, and ϵ , the additive approximation error in the rate function.

Proof. Recall M is the number of intervals (levels) in β . We use j, t, t^* to denote the levels in a certain iteration, the returned levels, and the optimal levels, respectively. We use $r(\cdot)$ to denote the individual rates between returned levels, i.e. $r(1) = -g_1 \log(1 - t_1)$, $r(m) = \{g_{m-1} \text{KL}(a_m || t_{m-1}) + g_m \text{KL}(a_m || t_m)\}, m \in \{2 \dots M - 2\}, r(M - 1) = -g_{M-2} \log(t_{M-2})$, and use r^* to denote the optimal rate.

By Lemma C.3.6, $t_{M-2}^* \geq 1 - \frac{1}{M-1}$. By assumption, $t_{M-2}^* < 1 - \delta$. Thus, $t_{M-2}^* \in [1 - \frac{1}{M-1}, 1 - \delta]$, the starting interval for the outer bisection.

First, suppose the outer bisection terminates such that $t_{M-2} \leq t_{M-2}^* + \delta$. We prove that this case always occurs below.

In this case, $r^* - r(M-1)$ is at most $-g_{M-2} \log(t_{M-2}^*) + g_{M-2} \log(t_{M-2}^* + \delta) = g_{M-2} \log\left(\frac{t_{M-2}^* + \delta}{t_{M-2}^*}\right)$. For all $m \in \{M-2 \dots 2\}$, in the final *CalculateOtherLevels* call the algorithm will use bisection to match the corresponding rate with this last rate, $r(M-1) = -g_{M-2} \log(t_{M-2})$, setting t_{m-2} to the smallest value such that $r(m) \leq r(M-1)$ (i.e. the right end of the final interval is chosen).

Then, $\forall m \in \{M-2 \dots 2\}$, $r(m) \in [r(M-1) - \epsilon(\delta), r(M-1)]$, where $\epsilon(\delta)$ is an upper bound on the change in the rate functions with a shift of δ in one of the parameters.

For now, assume $r(1) = -g_1 \log(t_1) \geq r(M-1)$. We prove that this occurs below. Then,

$$\begin{aligned} r(m) &\geq r(M-1) - \epsilon(\delta) && \forall m \in \{1 \dots M\} \\ &\geq -g_{M-2} \log(t_{M-2}^* + \delta) - \epsilon(\delta) \end{aligned}$$

Now we characterize $\epsilon(\delta)$ in the region $[t_1^* + \delta, t_{M-2}^* + \delta]$. In particular, we want to bound the rate loss from the other levels $r(m)$, $m > 1$ after the $g_{M-2} \log\left(\frac{t_{M-2}^* + \delta}{t_{M-2}^*}\right)$ loss in $r(M-1)$. Note that the only source of error is a level shifting right by δ . $r_j(\cdot)$ denotes individual rates between levels j in an intermediary iteration. Let a'_i be the minimum point inside the rate infimum after the shift by δ .

$$\begin{aligned} \epsilon(\delta) &= \sup_{t_{i-1}, t_i} [g_{i-1} \text{KL}(a_i || t_{i-1}) + g_i \text{KL}(a_i || t_i) - g_{i-1} \text{KL}(a'_i || t_{i-1} + \delta) - g_i \text{KL}(a'_i || t_i)] \\ &\leq \sup_{t_{i-1}, t_i} [g_{i-1} \text{KL}(a'_i || t_{i-1}) + g_i \text{KL}(a'_i || t_i) - g_{i-1} \text{KL}(a'_i || t_{i-1} + \delta) + g_i \text{KL}(a'_i || t_i)] && a_i \text{ is inf point} \\ &= \sup_{t_{i-1}, t_i} g_{i-1} \left[a'_i \log \frac{t_{i-1} + \delta}{t_{i-1}} + (1 - a'_i) \log \frac{1 - t_{i-1} - \delta}{1 - t_{i-1}} \right] \\ &\leq \sup_{t_{i-1}, t_i} g_{i-1} \left[a'_i \log \frac{t_{i-1} + \delta}{t_{i-1}} \right] && \text{2nd term negative} \\ &\leq g_{M-2} \left[\log \frac{t_1^* + \delta}{t_1^*} \right] && t_j \geq t_1^*, g_j \leq g_{M-2} \\ \implies r(m) &\geq r^* - g_{M-2} \log\left(\frac{t_{M-2}^* + \delta}{t_{M-2}^*}\right) - g_{M-2} \left[\log \frac{t_1^* + \delta}{t_1^*} \right] \\ &\geq r^* - g_{M-2} \frac{\delta}{t_{M-2}^*} - g_{M-2} \frac{\delta}{t_1^*} && \log(1+x) \leq x \\ &\geq r^* - \delta g_{M-2} \left[\frac{M-1}{M-2} + \frac{1}{t_1^*} \right] && t_{M-2}^* \geq 1 - \frac{1}{M-1} \end{aligned}$$

By Corollary C.3.3, $\exists C > 0$ s.t. $t_1^* \geq CM^{-3} \implies r(m) \geq r^* - \delta g_{M-2} \left[\frac{M-1}{M-2} + CM^3 \right]$. Then, let $\delta = \frac{\epsilon}{g_{M-2} \left[\frac{M-1}{M-2} + CM^3 \right]}$. Supposing the algorithm terminates in such an iteration, it finds an ϵ -optimal β in time $O\left(M \log^2 \frac{g_{M-2} \left[\frac{M-1}{M-2} + CM^3 \right]}{\epsilon}\right) = O\left(M \log^2 \frac{M}{\epsilon}\right)$.

Next, we show that the algorithm only terminates the outer bisection when $u \leq t_{M-2}^* + \delta$. The claim follows from $\ell \leq t_{M-2}^*$ being an algorithm invariant. The initial $\ell = 1 - \frac{1}{M-1} \leq t_{M-2}^*$ by Lemma C.3.6. ℓ can only be set to be $> t_{M-2}^*$ if in the current iteration, $j_{M-2} > t_{M-2}^*$ and $r_j(1) < r_j(M-1)$. However, if $j_{M-2} \geq t_{M-2}^*$, then $r_j(1) \geq r_j(M-1)$ ($j_m \geq t_m^* \forall m$), following from a shifting argument like that given in Lemma 4.3.1 and that the inner bisection is such that $r_j(m) \leq r_j(M-1)$, $m \in \{2 \dots M-2\}$, i.e. all the values $t_m > t_m^*$. Thus, $\ell \leq t_{M-2}^*$ is an algorithm invariant and $u > t_{M-2}^* + \delta \implies u - \ell > \delta$.

Finally, we show that $r(1) \geq r(M-1)$ at the returned $\{t_i\}$. By assumption, in the initial iteration, $u \geq t_{M-2}^*$, and recall that the returned $\{t_i\}$ such that $t_{M-2} = u$ from the final iteration. As shown in the previous paragraph, $j_{M-2} \geq t_{M-2}^* \implies r_j(1) \geq r_j(M-1)$. Thus, if the algorithm terminates in the first iteration, then $r(1) \geq r(M-1)$. In any subsequent iteration, u is changed only if $r_j(1) \geq r_j(M-1)$ at its new value. Thus, $r_j(1) \geq r_j(M-1)$ is an algorithm invariant, and $r(1) \geq r(M-1)$.

The algorithm terminates in finite time. Thus, it terminates when $t_{M-2} = u \leq t_{M-2}^* + \delta$ and finds a (ϵ, M, g) -optimal β in time $O(M \log^2 \frac{M}{\epsilon})$.

□

In Theorem 4.3.2, there is an guarantee of an additive error away from the optimal rate. To instead have a multiplicative error bound for uniform matching, one can use the lower bound on the optimal rate from Lemma C.3.7, $\exists C > 0$ s.t. $r^* \geq CM^{-3}$. Then, for uniform matching, the algorithm returns a $(1 - \epsilon)$ multiplicative approximation in time $O(M \log^2 \frac{M}{\epsilon})$.

C.3.7 Proof of Theorem C.2.1

Let β_M^w denote the optimal β with M intervals for weight function w , with intervals \mathbf{s}^{wM} and levels \mathbf{t}^{wM} . Let $q_{wM}(\theta) = i/M$ when $\theta \in [s_i^{wM}, s_{i+1}^{wM})$, i.e. the *quantile* of interval item of type θ is in. Then we have the following convergence result for β_M .

Theorem C.2.1. *Let g be uniform. Suppose w such that q_{wM} converges uniformly. Then, $\forall C \in \mathbb{N}, \exists \beta^w$ s.t. $\beta_{C^2 N+1}^w \rightarrow \beta^w$ uniformly as $N \rightarrow \infty$.*

Proof. Note that the condition on q implies that $\exists \bar{M}$ s.t. $\forall M > \bar{M}, \forall \theta, \exists x_\theta$ such that $\theta \in [s_{\lfloor x_\theta M \rfloor}^M, s_{\lceil x_\theta M \rceil}^M)$.

Let $M' = 2M - 1, M'' = 4M - 3, M^q = 2^q M - 2^q + 1$. $\theta \in [s_{\lfloor x_\theta M \rfloor}^M, s_{\lceil x_\theta M \rceil}^M) \implies \beta_M(\theta) =$

$t_{[x_\theta M]}^M \in [t_{[x_\theta M]-1}^M, t_{[x_\theta M]+1}^M]$. Then,

$$\begin{aligned} \beta_{M'}(\theta) &= t_{[x_\theta M']}^{M'} \\ &= t_{[x_\theta(2M-1)]}^{M'} \\ &\in [t_{2[x_\theta M]-2}^{M'}, t_{2[x_\theta M]+2}^{M'}] \\ &\subset [t_{[x_\theta M]-1}^M, t_{[x_\theta M]+1}^M] \end{aligned} \quad \text{Lemma C.3.5}$$

And, for general q ,

$$\begin{aligned} \beta_{M^q}(\theta) &= t_{[x_\theta(2^q M - 2^q + 1)]}^{M^q} \\ &\in [t_{[x_\theta 2^q M] - 2^q}^{M^q}, t_{[x_\theta(2^q M)] + 1}^{M^q}] \\ &\subset [t_{2^q[x_\theta M] - 2^q}^{M^q}, t_{2^q[x_\theta M] + 1}^{M^q}] \\ &\subset [t_{[x_\theta M] - 1}^M, t_{[x_\theta M] + 1}^M] \end{aligned} \quad \text{Lemma C.3.5}$$

Then, $\forall N' > 1$, θ : $\beta_{2^{N'} M - 2^{N'} + 1}(\theta) \in [t_{[x_\theta M] - 1}^M, t_{[x_\theta M] + 1}^M]$ and

$$|\beta_{2^{N'} M - 2^{N'} + 1}(\theta) - \beta_M(\theta)| \leq t_{[x_\theta M] + 1}^M - t_{[x_\theta M] - 1}^M$$

By Corollary C.3.2, $\forall \delta > 0$, $\exists K$ s.t. $\forall K' > K$, $t_{[x_\theta K'] + 1}^{K'} - t_{[x_\theta K'] - 1}^{K'} < 2\delta$.

By the Cauchy criterion, $\exists \beta$ s.t. $\beta_{(C-1)2^N + 1} \rightarrow \beta$ uniformly.

By change of variables, $\exists \beta$ s.t. $\beta_{C2^N + 1} \rightarrow \beta$ uniformly. \square

Corollary C.3.4. *For Kendall's tau and Spearman's rho correlation measures, $\exists \beta$ s.t. $\beta_{2^N} \rightarrow \beta$ uniformly as $N \rightarrow \infty$.*

Proof. For Kendall's tau and Spearman's rho, $\{s_i\}$ is spaced such that $\forall i, j, s_i - s_{i-1} = s_j - s_{j-1}$. Thus, $x_\theta = \theta$ meets the criterion. \square

C.3.8 Kendall's tau and Spearman's rho related proofs

Definition C.3.1 (see e.g. Embrechts et al. (2003); Nelsen (2007)). *The population version of Kendall-tau correlation between item true quality and rating scores is proportional to*

$$W_k^\tau \triangleq 2 \int_{\theta_1 > \theta_2} P_k(\theta_1, \theta_2) d\theta_1 d\theta_2$$

Similarly, given items with qualities $\theta_1, \theta_2, \theta_3$, the population version of Spearman's rho correlation between item true quality and rating scores is

$$W_k^\rho \triangleq 6 \int_{\theta_1 > \theta_2, \theta_3} P_k(\theta_1, \theta_3) d\theta_1 d\theta_2 d\theta_3$$

Lemma C.3.10. Spearman's ρ can also be written as being proportional to $\int_{\theta_1 > \theta_2} (\theta_1 - \theta_2) P_k(\theta_1, \theta_2) d\theta_1 d\theta_2$, i.e. with $w(\theta_1, \theta_2) = (\theta_1 - \theta_2)$.

Proof. Recall $P_k(\theta_1, \theta_3) = \Pr((\theta_1 - \theta_2)(x_1^k - x_3^k) > 0)$

$$\begin{aligned} &= \int_{\theta_1 > \theta_2, \theta_3} \Pr(x_1^k - x_3^k > 0) d\theta_1 d\theta_2 d\theta_3 + \int_{\theta_1 < \theta_2, \theta_3} \Pr(x_1^k - x_3^k < 0) d\theta_1 d\theta_2 d\theta_3 \\ &= \int_{\theta_1, \theta_3} \Pr(x_1^k - x_3^k > 0) \left[\int_{\theta_2=0}^{\theta_1} d\theta_2 \right] d\theta_1 d\theta_3 + \int_{\theta_1, \theta_3} \Pr(x_1^k - x_3^k < 0) \left[\int_{\theta_2=\theta_1}^1 d\theta_2 \right] d\theta_1 d\theta_3 \\ &= \int_{\theta_1, \theta_3} [\Pr(x_1^k - x_3^k > 0)\theta_1] + \Pr(x_1^k - x_3^k < 0)(1 - \theta_1)] d\theta_1 d\theta_3 \\ &= \int_{\theta_1, \theta_3} [\Pr(x_1^k - x_3^k < 0) + \theta_1 [\Pr(x_1^k - x_3^k > 0) - \Pr(x_1^k - x_3^k < 0)]] d\theta_1 d\theta_3 \end{aligned}$$

Similarly,

$$\begin{aligned} \Pr((\theta_1 - \theta_2)(x_1^k - x_3^k) < 0) &= \\ &= \int_{\theta_1, \theta_3} [\Pr(x_1^k - x_3^k > 0) + \theta_1 [\Pr(x_1^k - x_3^k < 0) - \Pr(x_1^k - x_3^k > 0)]] d\theta_1 d\theta_3 \\ &= \int_{\theta_1, \theta_3} [\Pr(x_3^k - x_1^k > 0) + \theta_3 [\Pr(x_3^k - x_1^k < 0) - \Pr(x_3^k - x_1^k > 0)]] d\theta_1 d\theta_3 \end{aligned}$$

Where the second equality follows from θ_1, θ_3 interchangeable. Then

$$\begin{aligned} W_k^\rho &= 3 \int_{\theta_1, \theta_2} (\theta_1 - \theta_2) P_k(\theta_1, \theta_2) d\theta_1 d\theta_2 \\ &= \int_{\theta_1 > \theta_2} 6(\theta_1 - \theta_2) P_k(\theta_1, \theta_2) d\theta_1 d\theta_2 \end{aligned}$$

□

Note that Spearman's ρ is similar to Kendall's τ with an additional weighting for how far apart the two values that are flipped are.

Lemma C.3.11. *When w is constant, i.e. for Kendall's τ rank correlation, the intervals \mathbf{s} that maximize (C.20),*

$$\sum_{0 \leq i < j < M} \int_{\theta_2 \in S_i, \theta_1 \in S_j} w(\theta_1, \theta_2) d(\theta_1, \theta_2) = \sum_{0 \leq i < j < M} (s_{i+1} - s_i)(s_{j+1} - s_j) \quad (\text{C.27})$$

, are $\{s_i = \frac{i}{M}\}_{i=0}^M$.

Lemma C.3.12. *When w is $(\theta_1 - \theta_2)$, i.e. for Spearman's ρ rank correlation, the intervals \mathbf{s} that maximize (C.20),*

$$\sum_{0 \leq i < j < M} \int_{\theta_2 \in S_i, \theta_1 \in S_j} w(\theta_1, \theta_2) d(\theta_1, \theta_2) \quad (\text{C.28})$$

are $\{s_i = \frac{i}{M}\}_{i=0}^M$, i.e. the same as those for Kendall's τ .

Proof.

$$\begin{aligned} \sum_{0 \leq i < j < M} \int_{\theta_2 \in S_i, \theta_1 \in S_j} w(\theta_1, \theta_2) d(\theta_1, \theta_2) &= \sum_{0 \leq i < j < M} \int_{\theta_2 \in S_i, \theta_1 \in S_j} (\theta_1 - \theta_2) d(\theta_1, \theta_2) \\ &= \sum_{0 < i < j \leq M} \left(\frac{s_j + s_{j-1}}{2} - \frac{s_i + s_{i-1}}{2} \right) (s_i - s_{i-1})(s_j - s_{j-1}) \end{aligned}$$

Finding an asymptotically optimal $\{s_i\}$ then is a constrained third order polynomial maximization problem with M variables. The maximum is achieved at $\{s_i = \frac{i}{M}\}_{i=0}^M$, as for Kendall's tau correlation. \square

Appendix D

Iterative Local Voting

D.1 Mechanical Turk Experiment Additional Information

In this section, we provide additional information regarding our Amazon Mechanical Turk experiment, including a walk-through of the user experience. Furthermore, we have a live demo accessible at: <http://gargnikhil.com/projectdetails/IterativeLocalVoting/>. This demo will remain online for the foreseeable future.

Figures D.1 through D.5 show screenshots of the experiment. We now walkthrough the experiment:

- Figure D.1 – Welcome page. Arriving from Amazon Mechanical Turk, the workers read an introduction and the consent agreement.
- Figure D.2 – Instructions (shown are \mathcal{L}^2 instructions). The workers read the instructions, which are also provided on the mechanism page. There is a 5 minute limit for this page.
- Figures D.3, D.5 – Mechanism page for \mathcal{L}^2 and Full Elicitation, respectively. For the former, workers are asked to move to their favorite point within a constraint set, for 2 different budget points. The “Current Credit Allocation” encodes the constraint set – as workers move the budget bars, it shows how much of their movement budget they have spent, and on which items. The other constrained movement mechanisms are similar. For the Full Elicitation mechanism, voters are simply asked to indicate their favorite budget point and weights. The instructions are repeated on the mechanism page as well at the top. There is a 10 minute limit for this page.
- Second mechanism, 30% of workers. Some workers were asked to do both one of the $\mathcal{L}^1, \mathcal{L}^2$, or \mathcal{L}^∞ , and the Full Elicitation mechanism. For these workers, the Full Elicitation mechanism shows up after the constrained mechanism.

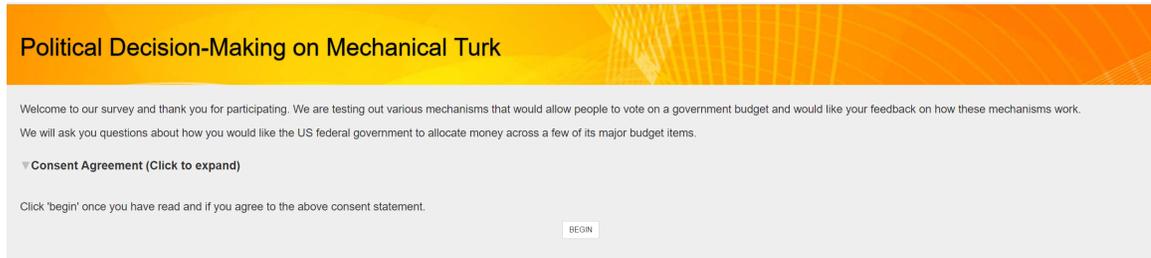
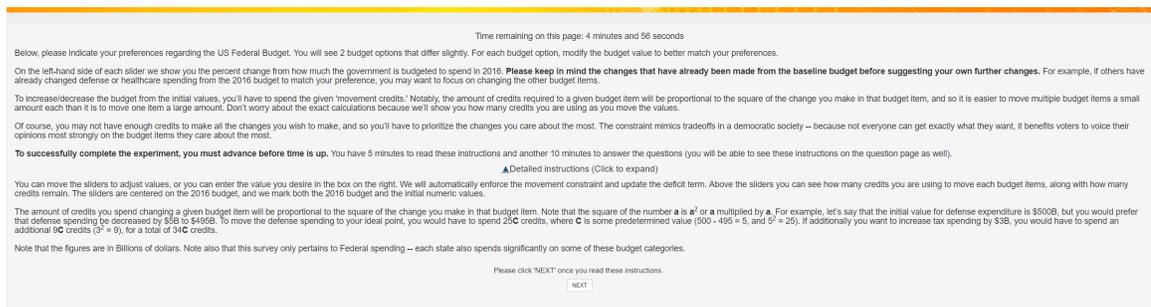


Figure D.1: Page 1 – Welcome Page for all mechanisms

Figure D.2: \mathcal{L}^2 Page 2 – Instructions

- Figure D.4 – Feedback page. Finally, workers are asked to provide feedback, after which they are shown a code and return to the Mechanical Turk website.

D.2 Indifference Regions Additional Information

We now present some additional data for the claim in Section 5.5.2, that voters have large indifference regions on the space. In particular, Figures D.6 and D.7 reproduce Figure 5.5 but with workers who provided explanations longer (and shorter) than the median response, respectively. This split can (roughly) correspond to workers who may have answered more or less sincerely to the budgeting question. We find that the response distribution, as measured by the fraction of possible movement one used when far away from one's ideal point on a given dimension, are similar.

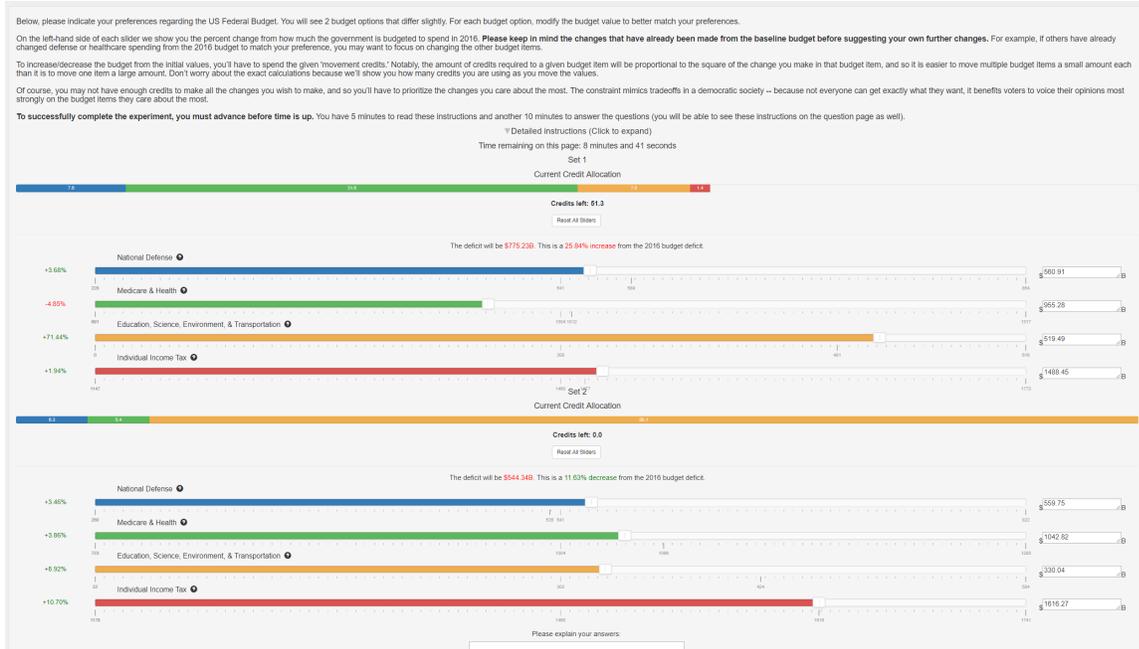


Figure D.3: \mathcal{L}^2 Page 3 – Mechanism

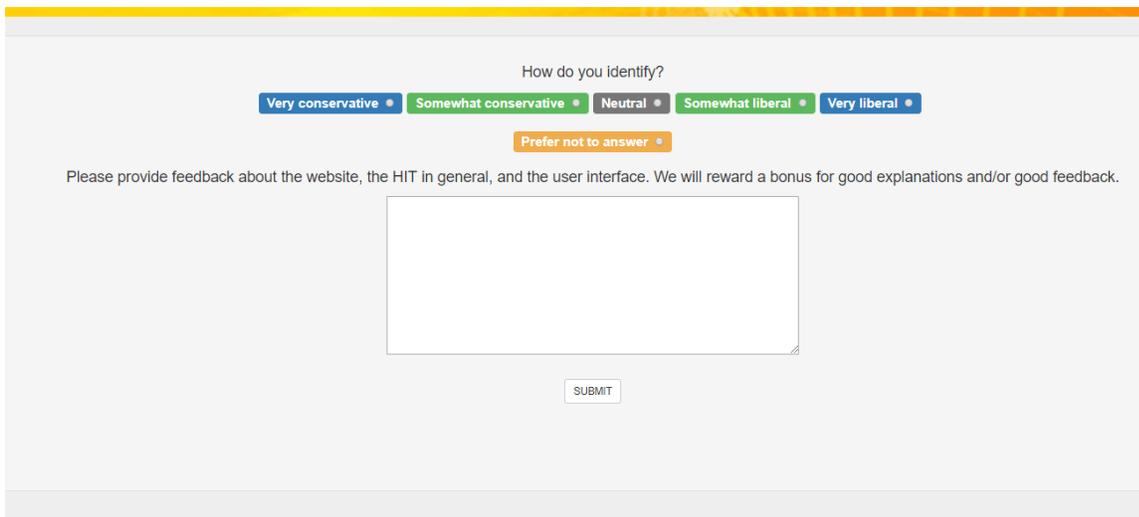


Figure D.4: Page 4 – Feedback for all mechanisms

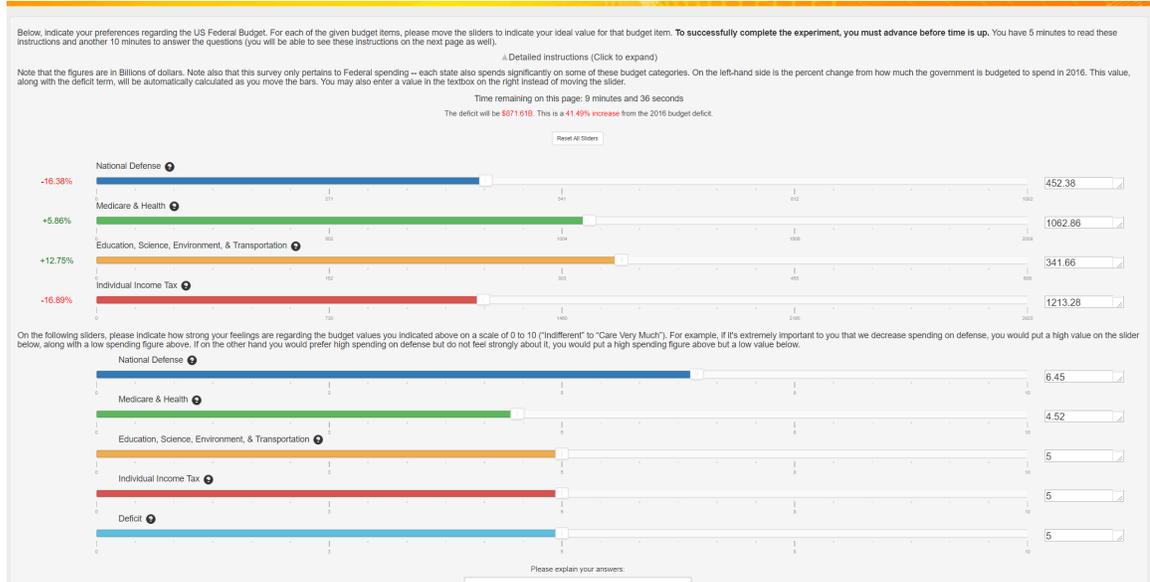


Figure D.5: Full Elicitation Page 3 – Mechanism

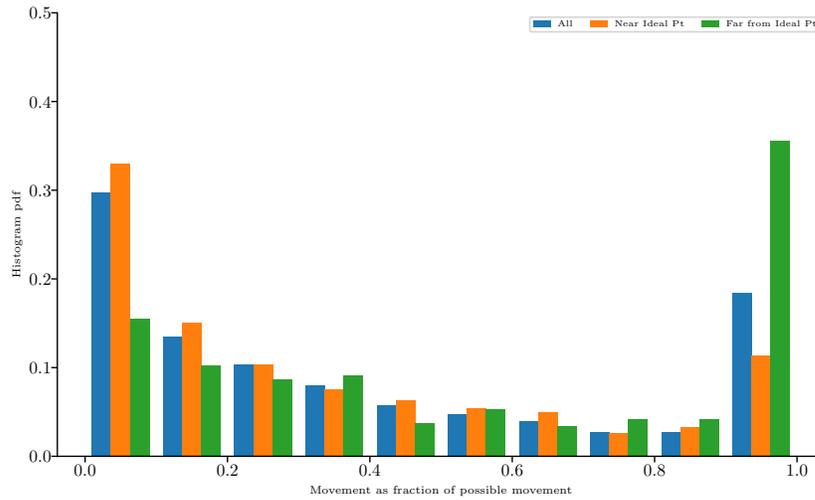


Figure D.6: Fraction of possible movement in each dimension in \mathcal{L}^∞ , conditioned on distance to ideal pt. The 'All' condition contains data from all three \mathcal{L}^∞ instances, whereas the others only from the instance that also did full elicitation. This plot only includes those people who provided an explanation as long or longer than the median explanation provided (197 characters).

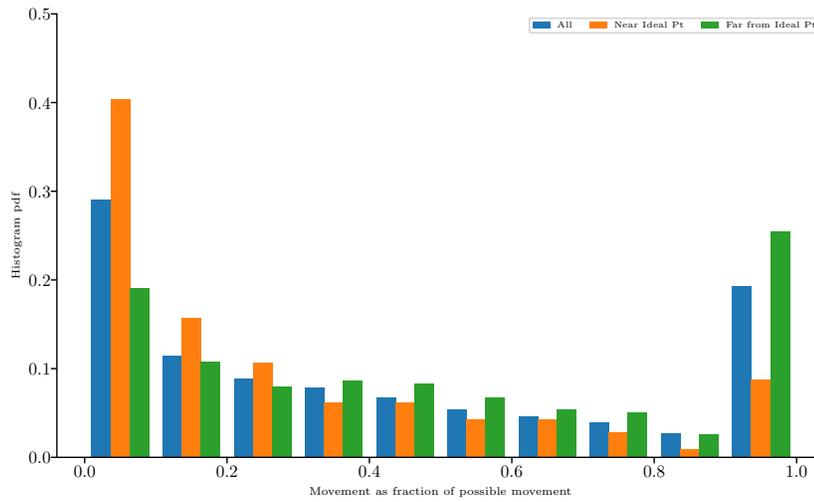


Figure D.7: Fraction of possible movement in each dimension in \mathcal{L}^∞ , conditioned on distance to ideal pt. The ‘All’ condition contains data from all three \mathcal{L}^∞ instances, whereas the others only from the instance that also did full elicitation. This plot only includes those people who provided an explanation shorter than the median explanation provided (197 characters).

D.3 Proofs

In this appendix, we include proofs for all the theorems in the paper.

D.3.1 Known SSGM Results

Theorem D.3.1. (Nemirovski et al., 2009; Strassen, 1965) Let $\theta \in \Theta$ be a random vector with distribution P . Let $\bar{f}(x) = \mathbb{E}[f(x, \theta)] = \int_{\Theta} f(x, \theta) dP(\theta)$, for $x \in \mathcal{X}$, a non-empty bounded closed convex set, and assume the expectation is well-defined and finite valued. Suppose that $f(\cdot, \theta), \theta \in \Theta$ is convex and $\bar{f}(\cdot)$ is continuous and finite valued in a neighborhood of point x .

For each θ , choose any $g(x, \theta) \in \partial f(x, \theta)$. Then, there exists $\bar{g}(x) \in \partial \bar{f}(x)$ s.t. $\bar{g}(x) = \mathbb{E}_{\theta}[g(x, \theta)]$.

This theorem says that the expected value of the sub-gradient of the utility at any point x across voters is a subgradient of the societal utility at x , irrespective of how the voters choose the subgradient when there are multiple subgradients, i.e., when the utility function is not differentiable. This key result allows us to use the subgradient of utility function of a sampled voter as an unbiased estimate of the societal subgradient.

Now, consider a convex function f on a non-empty bounded closed convex set $\mathcal{X} \subset \mathbb{R}^M$, and use $[\cdot]_{\mathcal{X}}$ to designate the projection operator. Starting with some $x_0 \in \mathcal{X}$, consider the SSGM update rule $x_t = [x_{t-1} - r_t(\bar{g}_t + z_t + b_t)]_{\mathcal{X}}$, where z_t is a zero-mean random variable and b_t is a constant, and $\bar{g}_t \in \partial f(x_t)$. Let $\mathbb{E}_t[\cdot]$ be the conditional expectation given \mathcal{F}_t , the σ -field generated by x_0, x_1, \dots, x_t . Then we have the following convergence result.

Theorem D.3.2. (Jiang and Walrand, 2010) Consider the above update rule. If

$$\begin{aligned}
 & f(\cdot) \text{ has a unique minimizer } x^* \in \mathcal{X} \\
 & r_t > 0, \sum_t r_t = \infty, \sum_t r_t^2 < \infty \\
 & \exists C_1 \in \mathbb{R} < \infty \text{ s.t. } \|\partial f(x)\|_2 \leq C_1, \forall x \in \mathcal{X} \\
 & \exists C_2 \in \mathbb{R} < \infty \text{ s.t. } \mathbb{E}_t[\|z_t\|^2] \leq C_2, \forall t \\
 & \exists C_3 \in \mathbb{R} < \infty \text{ s.t. } \|b_t\|_2 \leq C_3, \forall t \\
 & \sum_t r_t \|b_t\| < \infty \text{ w.p. } 1
 \end{aligned}$$

Then $x_t \rightarrow x^*$ w.p. 1 as $t \rightarrow \infty$.

Note: Jiang and Walrand (2010) prove the result for gradients, though the same proof follows for subgradients. Only the inequality $[x^* - x_t]^T g_t \leq f(x^*) - f(x_t)$ for gradient g_t at iteration t is used, which holds for subgradients. Boyd and Mutapcic provide a general discussion of subgradient

methods, along with similar results. Shor (1998), in Theorem 46, provide a convergence proof for the stochastic subgradient method without projections and the extra noise terms.

D.3.2 Mapping ILV to SSGM

As described in Section 5.3, suppose that $h_{\mathcal{X}}$ is the induced probability distribution on the ideal values of the voters. In the following discussion, we will refer to voters and their ideal solutions interchangeably.

Next, we restate ILV without the stopping condition so that it looks like the stochastic subgradient method. Consider Algorithm 4.

Algorithm 4: ILV

- 1 Start at some $x_0 \in \mathcal{X}$. For $t \geq 1$,
 - Sample voter $v_t \in \mathcal{V}$ from $h_{\mathcal{V}}$.
 - Compute $x_t = [x_{t-1} - r_t \tilde{g}_{v_t}(x_t)]_{\mathcal{X}}$, where $r_t = \frac{r_0}{t}$ and $r_t \tilde{g}_{v_t}(x_t)$ is movement given by voter v_t .
-

We want to minimize the societal cost, $\bar{f}(x) = \mathbb{E}[f_v(x)]$. From Theorem D.3.1, it immediately follows that if each voter v articulates a subgradient of her utility function for all x , i.e. $\tilde{g}_v(x) \in \partial f_v(x)$, then from Theorem D.3.2, we can conclude that the algorithm converges. However, users may not be able to articulate such a subgradient. Instead, when the voters respond correctly to query (1) (i.e. move to their favorite point in the given \mathcal{L}^q neighborhood), we have

$$\tilde{g}_{v_t}(x_t) = \frac{x_t - \arg \min_x [f_{v_t}(x) : \|x - x_t\|_q \leq r_t]}{r_t} \quad (2)$$

Furthermore, for all the proofs, we assume the following.

1. The solution space $\mathcal{X} \subset \mathbb{R}^M$ is non-empty, bounded, closed, and convex.
2. Each voter v has a unique ideal solution $x_v \in \mathcal{X}$.
3. The ideal point x_v of each voter is drawn independently from a probability distribution with a bounded and measurable density function $h_{\mathcal{X}}$ on M dimensions:
there exists C s.t. $\forall x$ we have $h_{\mathcal{X}}(x) \leq C$. This assumption allows us to bound the probability of errors that occur in small regions of the space.

D.3.3 Proof of Theorem 5.3.1

Let the disutility, or cost to voter $v \in \mathcal{V}$ be $f_v(x) = \|x - x_v\|_p$ for all $x \in \mathcal{X}$. We use the following technical lemma:

Lemma D.3.1. *For $q \in \{1, 2, \infty\}$, there exists $K_2 \in \mathbb{R}^+$ s.t. $\|\tilde{g}_v(x) - g_t\|_2 \leq K_2, \forall g_t \in \partial f_v(x)$ for any v and x .*

The lemma bounds the error in the movement direction from the gradient direction, by noting that both the movement direction and the gradient direction have bounded norms.

We also need the following lemma, which is proved separately for each case in the following sections.

Lemma D.3.2. *Suppose that $f_v(x) \triangleq \|x_v - x\|_p$ and define the function*

$$A_t \triangleq \mathbb{I}\{\tilde{g}_{v_t}(x_t) \notin \partial f_{v_t}(x_t)\},$$

where $\tilde{g}_{v_t}(x_t)$ is as defined in (2). Then there exists $C \in \mathbb{R}$ s.t. $\forall n, \mathbb{P}(A_t = 1 | \mathcal{F}_t) \leq Cr_t$, when $(p = 2, q = 2)$, $(p = 1, q = \infty)$, or $(p = \infty, q = 1)$.

The lemma can be interpreted as follows: A_t indicates a ‘bad’ event, when a voter may not be providing a true subgradient of her utility function. However, the probability of the event occurring vanishes with r_t , which, as we will see below, is the right rate for the algorithm to converge.

Theorem 5.3.1. *Suppose that conditions \mathbf{C}_1 , \mathbf{C}_2 , and \mathbf{C}_3 are satisfied, the voter utilities are \mathcal{L}^p normed, and voters respond to query (1) according to either **Model A** or **Model B**. Then, ILV with \mathcal{L}^q neighborhoods converges to the societal optimal point w.p. 1 when $(p, q) = (2, 2)$, $(1, \infty)$, or $(\infty, 1)$.*

Proof. We will show that Algorithm 4 meets the conditions in Theorem D.3.2. Let $b_t \triangleq \mathbb{E}_t[\tilde{g}_{v_t}(x_t)] - \bar{g}_t$ and $z_t \triangleq \tilde{g}_{v_t}(x_t) - \mathbb{E}_t[\tilde{g}_t]$, for some $\bar{g}_t \in \partial \bar{f}(x_t)$. Then, $\tilde{g}_{v_t}(x_t)$ can be written as $\tilde{g}_{v_t}(x_t) = \bar{g}_t + z_t + b_t$. We show that b_t, z_t meet the conditions in the theorem, and so the algorithm converges.

Let A_t be the indicator function described in Lemma D.3.2. Then, for some $\bar{g}_t \in \partial \bar{f}(x_t)$,

$$\begin{aligned} b_t &= \mathbb{E}_t[\tilde{g}_{v_t}(x_t)] - \bar{g}_t \\ &= \mathbb{E}_t[\tilde{g}_{v_t}(x_t)] - \mathbb{E}_t[g_t] && \text{Theorem D.3.1, i.i.d sampling of } v \\ &= \mathbb{P}(A_t = 1 | \mathcal{F}_t)(\mathbb{E}_t[\tilde{g}_{v_t}(x_t) | A_t = 1] - \mathbb{E}_t[g_t | A_t = 1]) \\ &\quad + \mathbb{P}(A_t = 0 | \mathcal{F}_t)(\mathbb{E}_t[\tilde{g}_{v_t}(x_t) | A_t = 0] - \mathbb{E}_t[g_t | A_t = 0]) \\ &= \mathbb{P}(A_t = 1 | \mathcal{F}_t)(\mathbb{E}_t[\tilde{g}_{v_t}(x_t) | A_t = 1] - \mathbb{E}_t[g_t | A_t = 1]) \\ &\leq Cr_t(\mathbb{E}_t[\tilde{g}_{v_t}(x_t) | A_t = 1] - \mathbb{E}_t[g_t | A_t = 1]). && \text{Lemma D.3.2} \end{aligned}$$

Combining with Lemma D.3.1, and the fact that $r_t = r_0/t$, we have

$$\sum r_t \|b_t\| \leq \infty \text{ and there exists } C_1 \in \mathbb{R} < \infty \text{ s.t. } \|b_t\|_2 \leq C_1, \forall t.$$

Finally, note that $\|z_t\| \triangleq \|\tilde{g}_{v_t}(x_t) - \text{E}_t[\tilde{g}_{v_t}(x_t)]\|$ is bounded for each t because the $\|\tilde{g}_{v_t}(x_t)\|$ is bounded as defined. Thus, all the conditions in Theorem D.3.2 are met for both b_t and z_t , and the algorithm converges. \square

D.3.4 Proof of Theorem 5.3.2

Instead of moving to their favorite point on the ball, voters now instead move in the direction of the gradient of their utility function to the boundary of the given neighborhood. In this case, we have:

$$\tilde{g}_{v_t}(x_t) = \frac{g_{v_t}}{\|g_{v_t}\|_q}; \text{ for } g_{v_t} \in \partial f_{v_t}(x_t). \quad (\text{D.1})$$

The key to the proof is the following observation, that the q norm of the gradient of the p norm, except at the ideal points on each dimension, is constant. This observation is formalized in the following lemma:

Lemma D.3.3. $\forall (p, q)$ s.t. $p > 0, q > 0$, and $1/p + 1/q = 1$, $\|\nabla \|x - x_v\|_p\|_q = 1, \forall x$ s.t. $x^m \neq x_v^m$ for any m .

Theorem 5.3.2. Suppose that conditions $\mathbf{C}_1, \mathbf{C}_2$, and \mathbf{C}_3 are satisfied, the voter utilities are \mathcal{L}^p normed, and voters respond to query (1) according to **Model B**. Then, ILV with \mathcal{L}^q neighborhoods converges to the societal optimal point w.p. 1 for any $p > 0$ and $q > 0$ such that $1/p + 1/q = 1$.

Proof. Since the probability of picking a voter v such that $x_t^m = x_v^m$ for some dimension m is 0, we have $\tilde{g}_{v_t}(x_t) = g_{v_t}$ for $g_{v_t} = \nabla f_{v_t}(x_t)$. Thus we obtain the gradient exactly, and hence Theorem D.3.2 applies with $b_t = 0$ for all t . \square

D.3.5 Proof of Propositions

We now turn our attention to the case of Weighted Euclidean utilities and show that Algorithm 4 converges to the societal optimum. The analogue to Lemma D.3.2 for this case is (proved in the following subsection):

Lemma D.3.4. Suppose that $f_v(x) \triangleq \sum_{k=1}^K \frac{w_v^k}{\|w_v\|_2} \|x^k - x_v^k\|_2$, and define the function

$$A_t \triangleq \mathbb{I}\{\tilde{g}_{v_t}(x_t) \notin \partial f_{v_t}(x_t)\},$$

where $\tilde{g}_{v_t}(x_t)$ is as defined in (2) for $q = 2$. Then there exists $C \in \mathbb{R}$ s.t. $\forall n, \text{P}(A_t = 1 | \mathcal{F}_t) \leq Cr_t$.

Suppose that conditions \mathbf{C}_1 , \mathbf{C}_2 , and \mathbf{C}_3 are satisfied, the voter utilities are Weighted Euclidean, and voters correctly respond to query (1) according to either **Model A** or **Model B**. Then, ILV with \mathcal{L}^2 neighborhoods converges with probability 1 to the societal optimal point.

Proof. The proof is then similar to that of Theorem 5.3.1, and the algorithm converges to $x^* = \arg \min \mathbb{E} \left[\frac{\sum_{k=1}^K w_v^k \|x^k - x_v^k\|_2}{\|w_v\|_2} \right]$. \square

Now, we sketch the proof for fully decomposable utility functions and \mathcal{L}^∞ neighborhoods.

Suppose that conditions \mathbf{C}_1 , \mathbf{C}_2 , and \mathbf{C}_3 are satisfied, the voter utilities are decomposable, and voters respond to query (1) according to either **Model A** or **Model B**. Then, ILV with \mathcal{L}^∞ neighborhoods converges with probability 1 to a point in the set of medians \bar{x} .

Proof. Consider each dimension separately. If $x_{t-1}^m < x_v^m$, then the sampled voter increases x_{t-1}^m by r_t as long as $x_{t-1}^m + r_t \leq x_v^m$. On the other hand if $x_{t-1}^m > x_v^m$, then the sampled voter decreases x_{t-1}^m by r_t as long as $x_{t-1}^m - r_t \geq x_v^m$. Thus except for when a voter's ideal solution is too close to the current point, the algorithm can be seen as performing SSGM on each dimension separately as if the utility function was \mathcal{L}^1 (the absolute value) on each dimension. Thus a proof akin to that of Theorem 5.3.1 with $p = 1, q = \infty$ holds. \square

D.3.6 Proof of Theorem 5.3.3

We now show that the algorithm finds directional equilibria in the following sense: if under a few conditions a trajectory of the algorithm converges to a point, then that point is a directional equilibrium.

Theorem 5.3.3. *Suppose that \mathbf{C}_1 , \mathbf{C}_2 , and \mathbf{C}_3 are satisfied, and let $G(x) \triangleq \mathbb{E}_v \left[\frac{\nabla f_v(x)}{\|\nabla f_v(x)\|_2} \right]$. Suppose, $G(x)$ is uniformly continuous, \mathcal{L}^2 movement norm constraints are used, and voters move according to **Model B**. If a trajectory $\{x\}_{t=1}^\infty$ of the algorithm converges to x^* , i.e. $x_t \rightarrow x^*$, then x^* is a directional equilibrium, i.e. $G(x^*) = 0$.*

Proof. Suppose x^* is not a directional equilibrium, i.e. $\exists \epsilon > 0$ s.t. $\|G(x^*)\|_2 = \epsilon$. Consider a δ -ball around x^* , $B_\delta \triangleq \{x : \|x^* - x\|_2 < \delta\}$, with $\delta, \epsilon_2 > 0$ chosen such that $\exists m \in \{1 \dots M\}$ s.t. $\forall x \in B_\delta$, $\text{sign}(G_m(x)) = \text{sign}(G_m(x^*))$ and $|G_m(x)| > \epsilon_2$, i.e. the gradient in the m th dimension does not change sign and has magnitude bounded below. Such a δ, ϵ_2 exists by the continuity assumption (if x^* is not a directional equilibrium, at least 1 dimension of $G(x^*)$ is non-zero and thus one can construct a ball around x^* such that $G(x), x \in B_\delta$ in that dimension satisfies the conditions).

Now, one can show that the probability of leaving neighborhoods around x^* goes to 1: $\forall t > 0, 0 < \delta_2 < \delta$, w.p. 1 $\exists \tau \geq t$ s.t. $\|x_\tau - x^*\|_2 > \delta_2$.

Suppose $x_t \in B_{\delta_2}$ (otherwise $\tau = t$ satisfies), $r_k = \frac{1}{k}$.

$$\begin{aligned}
x_\tau &= x_t + \sum_{k=t}^{\tau} \Delta x_k & \Delta x_k &\triangleq -r_k \frac{\nabla f^{v_k}(x_k)}{\|\nabla f^{v_k}(x_k)\|_2} \\
\|x_\tau - x^*\|_2 &= \|x_t - x^* + \sum_{k=t}^{\tau} \Delta x_k\|_2 \\
&\geq \left\| \sum_{k=t}^{\tau} \Delta x_k \right\|_2 - \|x_t - x^*\|_2 \\
&\geq \left\| \sum_{k=t}^{\tau} \Delta x_k \right\|_2 - \delta_2 \\
\left\| \sum_{k=t}^{\tau} \Delta x_k \right\|_2 &\geq \left| \sum_{k=t}^{\tau} \Delta x_{k,m} \right| && \text{defn of } \|\cdot\|_2 \\
&= \left| \mathbb{E}_v \left[\sum_{k=t}^{\tau} \Delta x_{k,m} \right] + \sum_{k=t}^{\tau} \Delta x_{k,m} - \mathbb{E}_v \left[\sum_{k=t}^{\tau} \Delta x_{k,m} \right] \right| \\
&\geq \left| \mathbb{E}_v \left[\sum_{k=t}^{\tau} \Delta x_{k,m} \right] \right| - \left| \sum_{k=t}^{\tau} \Delta x_{k,m} - \mathbb{E}_v \left[\sum_{k=t}^{\tau} \Delta x_{k,m} \right] \right|
\end{aligned}$$

By Hoeffding's inequality,

$$\begin{aligned}
Pr \left(\sum_{k=t}^{\tau} \Delta x_{k,m} - \mathbb{E}_v \left[\sum_{k=t}^{\tau} \Delta x_{k,m} \right] \geq \epsilon_3 \right) &\leq \exp \left[-\frac{2(\tau-t)^2 \epsilon_3^2}{2 \sum_{k=t}^{\tau} \frac{1}{k}} \right] \\
&\rightarrow 0 \text{ as } \tau \rightarrow \infty
\end{aligned}$$

Furthermore, by the continuity assumption,

$$\begin{aligned}
\left| \mathbb{E}_v \left[\sum_{k=t}^{\tau} \Delta x_{k,m} \right] \right| &\triangleq \left| \sum_{k=t}^{\tau} r_k G_m(x_k) \right| \\
&\rightarrow \infty \text{ as } \tau \rightarrow \infty \text{ while } x_k \in B_{\delta_2}
\end{aligned}$$

Thus, $Pr(\|x_\tau - x^*\|_2 > \delta_2) \rightarrow 1$ as $\tau \rightarrow \infty$. Thus, if an infinite trajectory converges to x^* , then w.p. 1, then x^* is a directional equilibrium. \square

D.3.7 Proofs of Lemmas

Lemma D.3.1 For $q \in \{1, 2, \infty\}$, $\exists K_2 \in \mathbb{R}^+ < \infty$ s.t. $\|\tilde{g}_{v_t} - g_t\|_2 \leq K_2, \forall g_t \in \partial f_{v_t}(x_t), v_t, x_t$.

Proof.

$$\begin{aligned}
\|\tilde{g}_{v_t}(x_t) - g_t\|_2 &\leq \|\tilde{g}_{v_t}(x_t)\|_2 + \|g_t\|_2 \\
&= \frac{\|x_t - \arg \min_x [\|x - x_{v_t}\|_p : \|x - x_t\|_q \leq r_t]\|_2}{r_t} + \|g_t\|_2 \\
&\leq K_1 + \|g_t\|_2 \\
&\leq K_2
\end{aligned}$$

for some $K_1, K_2 \in \mathbb{R}^+$. The second inequality follows from the fact that for finite M-dimensional vector spaces, $\|y\|_2 \leq \|y\|_1$ and $\|y\|_2 \leq \sqrt{M}\|y\|_\infty$. The third follows from the norm of the subgradients of the p norm being bounded. \square

Lemma D.3.2, case $(p = 2, q = 2)$.

Proof. Remember that $A_t \triangleq \mathbb{I}\{\tilde{g}_{v_t}(x_t) \notin \partial f_{v_t}(x_t)\}$. Let $B_t = \mathbb{I}\{\|x_{v_t} - x_t\|_2 \leq r_t\}$. We show that A) $B_t = 0 \implies A_t = 0$, and B) $\exists C \in \mathbb{R}$ s.t. $\mathbb{P}(B_t = 1 | \mathcal{F}_t) \leq Cr_t$. Then, $\exists C \in \mathbb{R}$ s.t. $\mathbb{P}(A_t = 1 | \mathcal{F}_t) \leq Cr_t$.

Part A, $B_t = 0 \implies \tilde{g}_{v_t}(x_t) = g_t$, for some $g_t \in \partial f_{v_t}(x_t)$:

First, note that

$$\begin{aligned}
\partial f_{v_t}(x) &= \partial \|x - x_{v_t}\|_2 \\
&= \begin{cases} \left\{ \frac{x - x_{v_t}}{\|x_{v_t} - x\|_2} \right\} & x \neq x_{v_t} \\ \{g : \|g\|_2 \leq 1\} & x = x_{v_t} \end{cases}
\end{aligned}$$

If $\|x_{v_t} - x_t\|_2 > r_t$, then

$$\arg \min_x [\|x - x_{v_t}\|_2 : \|x - x_t\|_2 \leq r_t] = x_t + r_t \frac{x_{v_t} - x_t}{\|x_{v_t} - x_t\|_2}$$

Then,

$$\begin{aligned}
\tilde{g}_{v_t}(x_t) &= \frac{x_t - \arg \min_x [\|x - x_{v_t}\|_2 : \|x - x_t\|_2 \leq r_t]}{r_t} && \text{Definition} \\
&= \frac{x_t - \left(x_t + r_t \frac{x_{v_t} - x_t}{\|x_{v_t} - x_t\|_2} \right)}{r_t} \\
&= \frac{x_t - x_{v_t}}{\|x_{v_t} - x_t\|_2} \\
&\in \partial f_{v_t}(x_t)
\end{aligned}$$

Part B, $\exists C \in \mathbb{R}$ s.t. $\mathbb{P}(B_t = 1 | \mathcal{F}_t) \leq Cr_t$:

$$\begin{aligned}
\mathbb{P}(B_t = 1 | \mathcal{F}_t) &= \mathbb{P}(\|x_{v_t} - x_t\|_2 \leq r_t | \mathcal{F}_t) \\
&= \int_{x \in \{x: \|x - x_t\|_2 \leq r_t\}} h_{\mathcal{X} | \mathcal{F}_t}(x) dx \\
&= \int_{x \in \{x: \|x - x_t\|_2 \leq r_t\}} h_{\mathcal{X}}(x) dx && v \text{ drawn independent of history} \\
&\leq C r_t^2 && \text{bounded } h_{\mathcal{X}} \\
&\leq C r_t && r_t \leq 1 \text{ eventually}
\end{aligned}$$

for some $C \in \mathbb{R} < \infty$. Note that C depends on the volume of a sphere in M dimensions. \square

Lemma D.3.2, case $(p = 1, q = \infty)$.

Proof. Let $h_{v_t}(x_t) \triangleq \left[\text{sign}(x_{v_t}^1 - x_t^1), \dots, \text{sign}(x_{v_t}^m - x_t^m), \dots, \text{sign}(x_{v_t}^M - x_t^M) \right]^T$

Let $B_t = \mathbb{I}\{\exists m, |x_{v_t}^m - x_t^m| \leq r_t\}$. We show the the same two parts as in the above proof.

Part A, $B_t = 0 \implies \tilde{g}_{v_t}(x_t) = g_t$, for some $g_t \in \partial f_{v_t}(x_t)$:

First, note that the subgradients are

$$\begin{aligned}
\partial f_{v_t}(x) &= \partial \|x - x_{v_t}\|_1 \\
&= \{g : \|g\|_\infty \leq 1, g^T(x - x_{v_t}) = \|x - x_{v_t}\|_1\}
\end{aligned}$$

If $\forall m, |x_{v_t}^m - x_t^m| > r_t$, then

$$\arg \min_x [\|x - x_{v_t}\|_1 : \|x - x_t\|_\infty \leq r_t] = x_t + r_t h_{v_t}(x_t)$$

Then,

$$\begin{aligned}
\tilde{g}_{v_t}(x_t) &= \frac{x_t - \arg \min_x [\|x - x_{v_t}\|_1 : \|x - x_t\|_\infty \leq r_t]}{r_t} && \text{Definition} \\
&= \frac{x_t - (x_t + r_t h_{v_t}(x_t))}{r_t} \\
&= -h_{v_t}(x_t) \\
&\in \partial f_{v_t}(x_t)
\end{aligned}$$

Part B, $\exists C \in \mathbb{R}$ s.t. $\mathbb{P}(B_t = 1 | \mathcal{F}_t) \leq C r_t$:

$$\begin{aligned}
\mathbb{P}(B_t = 1 | \mathcal{F}_t) &= \mathbb{P}(\exists m : |x_{v_t}^m - x_t^m| \leq r_t | \mathcal{F}_t) \\
&= \int_{x \in \{x : \exists m, |x^m - x_t^m| \leq r_t\}} h_{\mathcal{X} | \mathcal{F}_t}(x) dx \\
&= \int_{x \in \{x : \exists m, |x^m - x_t^m| \leq r_t\}} h_{\mathcal{X}}(x) dx && v \text{ drawn independent of history} \\
&\leq Cr_t && \text{bounded } h_{\mathcal{X}}, \text{ fixed } M, \text{ bounded } \mathcal{X}
\end{aligned}$$

for some $C \in \mathbb{R} < \infty$. In the last line, $C \cong 2M(\text{diameter}(\mathcal{X}))$, based on the volume of the slices around the ideal points on each dimension. \square

Lemma D.3.2, case $(p = 1, q = \infty)$.

Proof. Let $\bar{m}_t \in \arg \max_m |x_{v_t}^m - x_t^m|$,

Let $h_{v_t}(x_t) \triangleq [0, 0, \dots, 0, \text{sign}(x_t^{\bar{m}_t} - x_{v_t}^{\bar{m}_t}), 0, \dots, 0, 0]^T$,

Let $B_t \triangleq \mathbb{I}\{\exists m \neq \bar{m}_t : |x_{v_t}^{\bar{m}_t} - x_t^{\bar{m}_t}| < |x_{v_t}^m - x_t^m| + r_t\}$. We show the the same two parts as in the above proofs.

Part A, $B_t = 0 \implies \tilde{g}_{v_t}(x_t) = g_t$, for some $g_t \in \partial f_{v_t}(x_t)$:

First, note that when $B_t = 0$, the set of subgradients is

$$\begin{aligned}
\partial f_{v_t}(x) &= \partial \|x - x_{v_t}\|_{\infty} \\
&= \{h_{v_t}(x_t)\}
\end{aligned}$$

Also when $B_t = 0$,

$$\arg \min_x [\|x - x_{v_t}\|_{\infty} : \|x - x_t\|_1 \leq r_t] = x_t - r_t h_{v_t}(x_t)$$

Then,

$$\begin{aligned}
\tilde{g}_{v_t}(x_t) &= \frac{x_t - \arg \min_x [\|x - x_{v_t}\|_1 : \|x - x_t\|_{\infty} \leq r_t]}{r_t} && \text{Definition} \\
&= \frac{x_t - (x_t - r_t h_{v_t}(x_t))}{r_t} \\
&= h_{v_t}(x_t) \\
&\in \partial f_{v_t}(x_t)
\end{aligned}$$

Part B, $\exists C \in \mathbb{R}$ s.t. $\mathbb{P}(B_t = 1 | \mathcal{F}_t) \leq Cr_t$:

$$\begin{aligned}
& \mathbb{P}(B_t = 1 | \mathcal{F}_t) \\
&= \mathbb{P}(\mathbb{I}\{\exists m \neq \bar{m}_t : |x_{v_t}^{\bar{m}_t} - x_t^{\bar{m}_t}| < |x_{v_t}^m - x_t^m| + r_t\} | \mathcal{F}_t) \\
&= \int_{x \in \{x: \exists m \neq \bar{m}_t \text{ s.t. } |x_{v_t}^{\bar{m}_t} - x_t^{\bar{m}_t}| < |x_{v_t}^m - x_t^m| + r_t\}} h_{\mathcal{X} | \mathcal{F}_t}(x) dx \\
&= \int_{x \in \{x: \exists m \neq \bar{m}_t \text{ s.t. } |x_{v_t}^{\bar{m}_t} - x_t^{\bar{m}_t}| < |x_{v_t}^m - x_t^m| + r_t\}} h_{\mathcal{X}}(x) dx && v \text{ ind. of history} \\
&\leq Cr_t && \text{bounded } h_{\mathcal{X}}, \mathcal{X}, \text{ fixed } M
\end{aligned}$$

for some $C \in \mathbb{R} < \infty$. Note that $C \approx 2M^2(\text{diameter}(\mathcal{X}))$, based on the volume of the slices around each dimension. \square

Lemma D.3.3 $\forall (p, q)$ s.t. $p > 0, q > 0, 1/p + 1/q = 1, \|\nabla\|x - x_v\|_p\|_q = 1, \forall x$ s.t. $\forall m, x^m \neq x_v^m$.

Proof. If $x^m \neq x_v^m, \forall m$:

$$\begin{aligned}
\nabla_m \|x - x_v\|_p &= \nabla_m \left(\sum_m |x^m - x_v^m|^p \right)^{1/p} \\
&= \frac{1}{p} \frac{\nabla_m |x^m - x_v^m|^p}{\left(\sum_m |x^m - x_v^m|^p \right)^{1-1/p}} \\
&= \frac{|x^m - x_v^m|^{p-1} (\nabla_m |x^m - x_v^m|)}{\|x - x_v\|_p^{p-1}}
\end{aligned}$$

Then

$$\begin{aligned}
\|\nabla\|x - x_v\|_p\|_q &= \left\| \frac{|x^m - x_v^m|^{p-1} (\nabla_m |x^m - x_v^m|)}{\|x - x_v\|_p^{p-1}} \right\|_q \\
&= \frac{1}{\|x - x_v\|_p^{p-1}} \left(\sum_m \left| |x^m - x_v^m|^{p-1} (\nabla_m |x^m - x_v^m|) \right|^q \right)^{1/q} \\
&= \frac{1}{\|x - x_v\|_p^{p-1}} \left(\sum_m |x^m - x_v^m|^{(p-1)q} \right)^{1/q} \\
&= \frac{1}{\|x - x_v\|_p^{p-1}} \|x - x_v\|_p^{p/q} && (p-1)q = p \\
&= 1
\end{aligned}$$

\square

Lemma D.3.4 Suppose that $f_v(x) \triangleq \sum_{k=1}^K \frac{w_v^k}{\|w_v\|_2} \|x^k - x_v^k\|_2$, and define the function

$$A_t \triangleq \mathbb{I}\{\tilde{g}_{v_t}(x_t) \notin \partial f_{v_t}(x_t)\},$$

where $\tilde{g}_{v_t}(x_t)$ is as defined in (2) for $q = 2$. Then there exists $C \in \mathbb{R}$ s.t. $\forall n, \mathbb{P}(A_t = 1 | \mathcal{F}_t) \leq Cr_t$.

Proof. Let $B_t = \mathbb{I}\{\exists k \text{ s.t. } \|x_{v_t}^k - x_t^k\|_2 \leq r_t\}$. We show the same two parts for B_t as for the proofs for Lemma D.3.2.

Part A, $B_t = 0 \implies \tilde{g}_{v_t}(x_t) = g_t$, for some $g_t \in \partial f_{v_t}(x_t)$:

First, note that, when $B_t = 0$,

$$\begin{aligned} \partial_m f_{v_t}(x_t) &= \partial_m \sum_{k=1}^K \frac{w_v^k}{\|w_v\|_2} \|x^k - x_v^k\|_2 \\ &= \frac{w^{k_m}}{\|w_v\|_2} \frac{x^m - x_{v_t}^m}{\|x_{v_t}^{k_m} - x_t^{k_m}\|_2} \quad k_m \text{ is subspace containing the } m\text{th dimension} \end{aligned}$$

Also if $B_t = 0$, then

$$\arg \min_x \left[\sum_{k=1}^K \frac{w^k}{\|w_v\|_2} \|x^k - x_v^k\|_2 : \|x - x_t\|_2 \leq r_t \right] = x_t + r_t \left[\dots, \frac{w^{k_m}}{\|w_v\|_2} \frac{x_{v_t}^m - x^m}{\|x_{v_t}^{k_m} - x_t^{k_m}\|_2}, \dots \right]$$

Then,

$$\begin{aligned} \tilde{g}_{v_t}(x_t) &= \frac{x_t - \arg \min_x [\|x - x_{v_t}\|_2 : \|x - x_t\|_2 \leq r_t]}{r_t} && \text{Definition} \\ &\in \partial f_{v_t}(x_t) \end{aligned}$$

Part B, $\exists C \in \mathbb{R}$ s.t. $\mathbb{P}(B_t = 1 | \mathcal{F}_t) \leq Cr_t$:

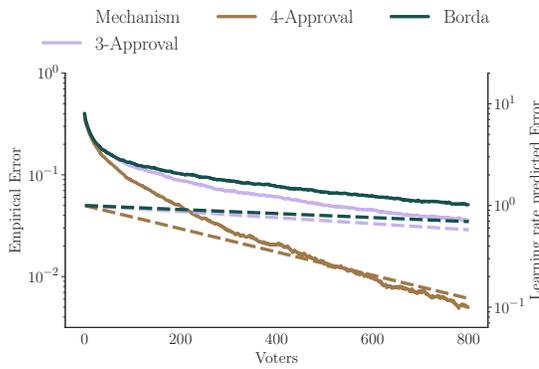
$$\begin{aligned} \mathbb{P}(B_t = 1 | \mathcal{F}_t) &= \mathbb{P}(\|x_{v_t} - x_t\|_2 \leq r_t | \mathcal{F}_t) \\ &= \int_{x \in \{x: \exists k \text{ s.t. } \|x_{v_t}^k - x_t^k\|_2 \leq r_t\}} h_{\mathcal{X} | \mathcal{F}_t}(x) dx \\ &= \int_{x \in \{x: \exists k \text{ s.t. } \|x_{v_t}^k - x_t^k\|_2 \leq r_t\}} h_{\mathcal{X}}(x) dx && v \text{ drawn independent of history} \\ &\leq Cr_t^2 && \text{bounded } h_{\mathcal{X}} \\ &\leq Cr_t && r_t \leq 1 \text{ eventually} \end{aligned}$$

for some $C \in \mathbb{R} < \infty$. Note that C depends on K and M . \square

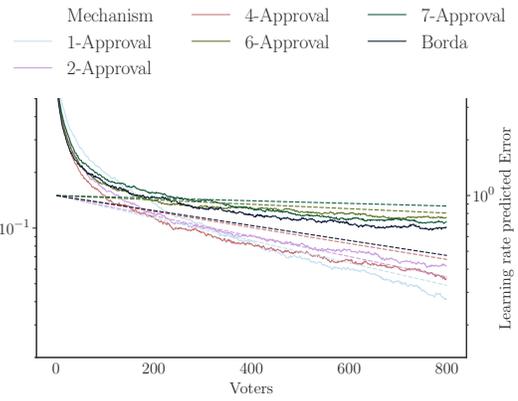
Appendix E

Who is in Your Top Three? Optimizing Learning in Elections with Many Candidates

E.1 Empirics additional information



(a) Boston 2016, selecting 4 winners.



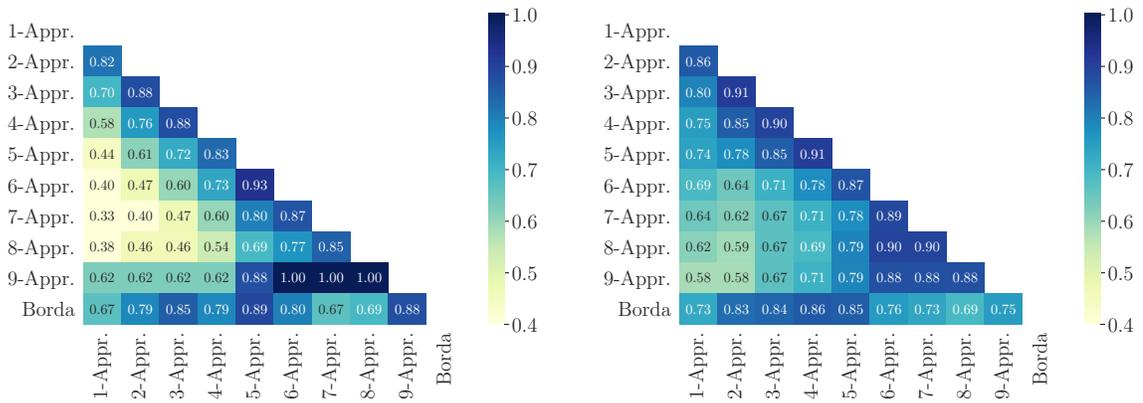
(b) Durham Ward 1, selecting 4 winners. K -Approval for $K \in \{1, \dots, 7\}$ and the Borda rule all have the same asymptotic winners, but we omit several mechanisms from the plot for visualization ease.

Figure E.1: Average bootstrapped error (fraction of winning subset not identified) by the number of voters, compared to the errors implied by the (empirically calculated) learning rates. All mechanisms plotted have the same asymptotic winners.

Name	Candidates	Votes with complete rankings	K-Ranking available
Participatory Budgeting			
Boston, 2016	8	4173	4
Durham Ward 1, 2019	21	1637	10
Durham Ward 2, 2019	10	329	10
Durham Ward 3, 2019	12	694	10
Rochester, 2019	22	649	5
PrefLib:			
Irish01	12	4259	12
Irish02	9	4810	9
Irish03	14	3166	14
ElectorialReformSociety77	12	1312	12
ElectorialReformSociety13	5	1809	5
Sushi10	10	5000	10
Glasgow05	10	718	10
Glasgow17	9	962	9
Glasgow10	9	818	9
Glasgow18	9	767	9
Glasgow20	9	726	9
Glasgow14	8	1071	8
Glasgow12	8	1040	8
Burlington01	6	2603	6
Burlington02	6	2853	6
APA03	5	11539	5
APA01	5	10978	5
APA11	5	10791	5
APA05	5	10655	5
APA02	5	10623	5
APA04	5	10519	5
APA09	5	10211	5
APA06	5	10177	5
APA07	5	9747	5
APA12	5	9091	5
APA08	5	8532	5
APA10	5	8467	5
Aspen02	5	1183	5

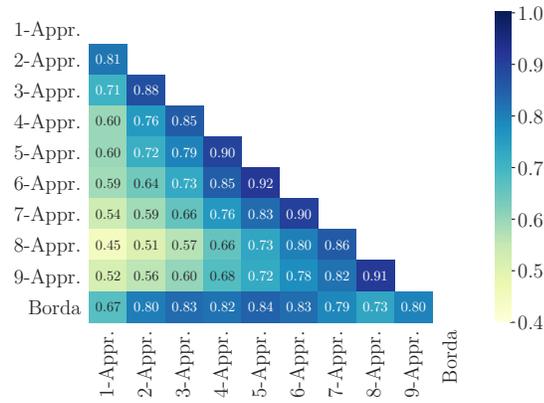
Table E.1: List of election data that we use in Section 6.6. From PrefLib, we use all elections where full rankings are available and there are at least 5 candidates and 700 voters. Throughout, we ignore voters who did not submit full rankings (especially with high K -Ranking requested, this might only be a fraction of the total number of actual votes). Additionally, for the PB elections, we limit the data to those who submitted votes online rather than through paper ballots.

Sources for the PrefLib datasets are: Mattei and Walsh (2013); O’Neill (2013); Popov et al. (2014); Regenwetter et al. (2007, 2008).



(a) Task of selecting $W = 1$ winners.

(b) Task of selecting $W = 3$ winners.



(c) Task of ranking all candidates. The values plotted are the average Kendall's τ rank correlation between resulting rankings.

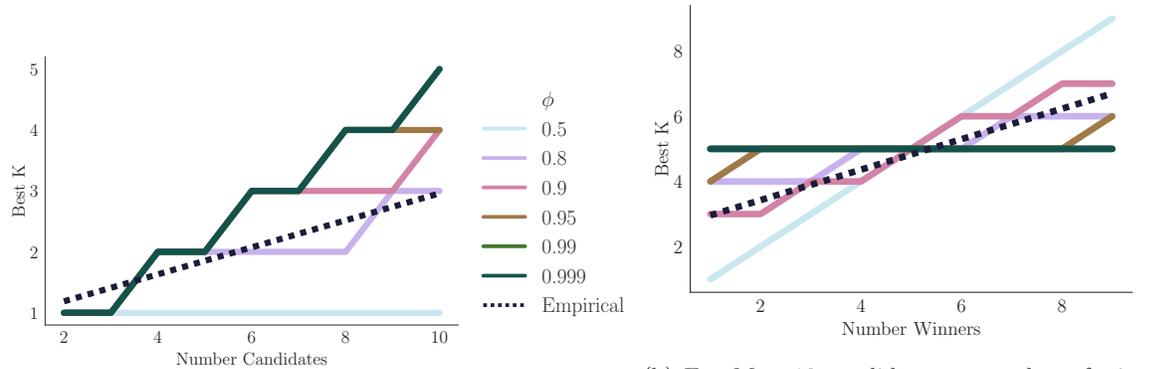
Figure E.2: More approximate design invariance plots

Dep. Variable:	Best Mechanism	R-squared:	0.273
Model:	OLS	Adj. R-squared:	0.264
Method:	Least Squares	F-statistic:	42.68
Date:	Wed, 12 Jun 2019	Prob (F-statistic):	2.72e-11
Time:	16:17:32	Log-Likelihood:	-531.30
No. Observations:	241	AIC:	1071.
Df Residuals:	237	BIC:	1085.
Df Model:	3		

	coef	std err	z	P> z	[0.025	0.975]
Intercept	-0.1687	0.411	-0.411	0.681	-0.973	0.636
Number Winners	0.9133	0.126	7.229	0.000	0.666	1.161
Number Candidates	0.2662	0.057	4.630	0.000	0.154	0.379
Number Winners:Number Candidates	-0.0446	0.008	-5.786	0.000	-0.060	-0.030

Omnibus:	8.524	Durbin-Watson:	1.693
Prob(Omnibus):	0.014	Jarque-Bera (JB):	8.414
Skew:	0.417	Prob(JB):	0.0149
Kurtosis:	2.624	Cond. No.	463.

Table E.2: OLS Regression on the best K to use in K-Approval, by the number of candidates and desired winners. Standard errors are cluster standard errors, where each cluster is an election in our dataset.



(a) For selecting $W = 1$ winner as number of candidates vary. (b) For $M = 10$ candidates as number of winners vary.

Figure E.3: K -Approval rate optimal mechanism for the Mallows model as ϕ , number of candidates, and number of winners vary. This plot contains an empirical line, which is calculated using the coefficients in the regression contained in Table E.2.

Election	Number Winners	Mechanism 1	Mechanism 2	Beats Approval rate optim
Durham Ward 1, 2019	2	3	4	True
Durham Ward 1, 2019	13	8	9	True
Durham Ward 1, 2019	17	6	7	True
Irish03	1	1	3	False
Irish03	10	8	9	False
Irish03	10	8	10	False
Irish01	5	3	4	True
Irish01	5	3	5	False
Irish01	5	3	11	False
Irish01	7	1	5	False
Glasgow05	2	4	5	True
Glasgow05	2	5	6	True
Glasgow10	2	3	4	False
Glasgow10	2	4	6	False
Glasgow10	2	5	6	True
APA08	2	2	3	True

Table E.3: Elections and goals where randomizing between two K -Approval mechanisms produces leads to faster learning than using either of the mechanisms separately. For several of these cases, randomization also beats the Approval rate optimal mechanism.

E.2 Proofs

E.2.1 Asymptotic design-invariance

A setting (M, F) for goal G is asymptotically design-invariant if and only if there exist candidate tiers $O^* = \{C_1^* \dots C_T^*\}$ (corresponding to G) s.t. $\forall s < t: i \in C_s^*, j \in C_t^* \implies \Pr_F(\sigma_v(i) \leq k) > \Pr_F(\sigma_v(j) \leq k), \forall k \in \{1 \dots M-1\}$.

Proof.

$$\begin{aligned}
\forall i \in C : \mathbb{E}[s_{iv}] &= \sum_{m=1}^M \beta(m) \Pr_F(\sigma_v(i) = m) \\
&= \sum_{m=1}^M \beta(m) \Pr_F(\sigma_v(i) \leq m) - \sum_{m=2}^M \beta(m) \Pr_F(\sigma_v(i) < m) \\
&= \sum_{m=1}^M \beta(m) \Pr_F(\sigma_v(i) \leq m) - \sum_{m=1}^{M-1} \beta(m+1) \Pr_F(\sigma_v(i) \leq m) \\
&= \beta(M) + \sum_{m=1}^{M-1} [\beta(m) - \beta(m+1)] \Pr_F(\sigma_v(i) \leq m)
\end{aligned}$$

\implies . By the definition of asymptotically design-invariant,

$$\exists O^* : \forall \beta \in \mathcal{B}, \lim_{N \rightarrow \infty} \mathcal{O}(M, N, F, \beta, G) = O^*, \text{ with probability } 1$$

For this $O^* = \{C_1^*, \dots, C_T^*\}$, we show by contradiction that $\forall s < t: i \in C_s^*, j \in C_t^* \implies \Pr_F(\sigma_v(i) \leq k) > \Pr_F(\sigma_v(j) \leq k), \forall k \in \{1 \dots M-1\}$: Suppose $\exists i \in C_s^*, j \in C_t^*, s < t, k \in \{1 \dots M-1\}$ such that $\Pr_F(\sigma_v(i) \leq k) \leq \Pr_F(\sigma_v(j) \leq k)$. Then, let

$$\beta(m) = \begin{cases} 1 & m \leq k \\ 0 & m > k \end{cases}$$

Then, $\mathbb{E}[s_{iv}] = \beta(M) + \beta(k) \Pr_F(\sigma_v(i) \leq k) \leq \mathbb{E}[s_{jv}]$. Then, with positive probability,

$$\lim_{N \rightarrow \infty} \mathcal{O}(M, N, F, \beta, G) \neq O^*$$

.

\Leftarrow . Suppose there exists such a O^* . Then, $\forall \beta \in \mathcal{B} = \{\beta : \forall k < \ell \in 1 \dots M, \beta(k) \geq \beta(\ell)\}$, and $\exists k <$

$\ell, \beta(k) > \beta(\ell)$: Suppose $i \in C_s^*, j \in C_t^*, s < t$:

$$\begin{aligned} \mathbb{E}[s_{iv}] &= \beta(M) + \sum_{m=1}^{M-1} [\beta(m) - \beta(m+1)] \Pr_F(\sigma_v(i) \leq m) \\ &> \mathbb{E}[s_{jv}] \end{aligned}$$

Where the strict inequality follows as $\exists m : \beta(m) - \beta(m+1) > 0$. Then, for all candidates $i \in C_s^*, j \in C_t^*, s < t$, by the strong law of large numbers $\lim_{N \rightarrow \infty} s_i^N > \lim_{N \rightarrow \infty} s_j^N$ w.p. 1. Thus, $\lim_{N \rightarrow \infty} \mathcal{O}(M, N, F, \beta, G) = O^*$ w.p. 1. \square

Remark E.2.1. *The following example, with candidates A, B, C, D leads to a disjoint set of 2 winners with 1-Approval and 2 approval, respectively*

	Voter 1	Voter 2	Voter 3	Voter 4	Voter 5
Rank 1	A	A	D	D	B
Rank 2	B	B	C	C	C
Rank 3	C	C	B	B	D
Rank 4	D	D	A	A	A

With 1-Approval, candidates A, D are selected. With 2-Approval, B, C are selected.

E.2.2 Learning rates

Notation $t_{ij}^i(K)$ is the probability that i is approved but j is not, using K -Approval.

For convenience, we overload the rate function $r(\cdot)$:

- $r_{ij}(\beta)$ is as defined in Proposition 6.4.1, the large deviation rate to learn a pair of candidates i, j given scoring rule β , for a fixed F that should be clear from context. When a goal G is clear from context, $r(\beta)$ is as defined in Proposition 6.4.1, the minimum over $r_{ij}(\beta)$ for candidate pairs that are in different asymptotic tiers.
- $r_{ij}(K)$ is as defined in Proposition 6.4.1, the large deviation rate to learn a pair of candidates i, j using K -Approval. $r(K)$ is analogous to the previous item when using K approval.
- $r(a, b)$ is the large deviation rate to learn a pair of candidates i, j using approval voting when the probability that i is approved but j is not is a ; and b is the probability that j is approved but i is not is not.

When which rate function we mean is clear from context, we may drop the argument (\cdot) and just write r_{ij} or r .

Remark E.2.2. $r(a, b) > r(c, d)$ when $a > c, b \leq d$, OR $a \geq c, b < d$.

Proof. $\gamma(a, b) = \sqrt{ab} + 1 - a - b$ is strictly concave in a, b , with maximum at $a = b$. Thus, holding either a or b constant and moving the other farther away strictly decreases γ , and thus strictly increases r . \square

Fix scoring rule $\beta \in \mathcal{B}$, voter distribution F , and consider candidates i, j such that $s_i > s_j$. Then, the probability of making a mistake in ranking these two candidates after N voters, $\Pr(\sigma^N(i) > \sigma^N(j))$, goes to zero with large deviation rate

$$r_{ij}(\beta) = - \inf_{z \in \mathbb{R}} \log \mathbb{E}_F [\exp(z(\beta(\sigma_v(i)) - \beta(\sigma_v(j))))]$$

Further, the following upper bound holds for any N .

$$\Pr(\sigma^N(i) > \sigma^N(j)) \leq \exp(-r_{ij}(\beta)N)$$

Proof. Define the following random variable for each voter $v \sim F$:

$$A_v = \beta(\sigma_v(i)) - \beta(\sigma_v(j))$$

Then, $\sigma^N(i) < \sigma^N(j)$ when $A^N = \sum_{v=1}^N A_v > 0$, and $\mathbb{E}[A_v] > 0$ by supposition. Let

$$r_{ij}(\beta) = - \inf_{z \in \mathbb{R}} \Lambda(z)$$

$$\Lambda(z) = \log \left[\sum_{m=1}^M \sum_{\ell \neq m} \Pr(\sigma_v(i) = m, \sigma_v(j) = \ell) \exp[z(\beta(\sigma_v(i)) - \beta(\sigma_v(j)))] \right]$$

Then, by basic large deviation bounds (see, e.g. Dembo and Zeitouni (2010)):

$$- \lim_{N \rightarrow \infty} \frac{1}{N} \log \Pr(A^N \leq 0) = r_{ij}(\beta)$$

And, applying Chernoff bounds, we get the standard relationship to the large deviation rate, giving an upper bound for the probability of error directly, including any polynomial factors out

front:

$$\begin{aligned}
\Pr(\sigma^N(i) > \sigma^N(j)) &\leq \Pr(A^N \leq 0) \\
&< \left[\inf_{z>0} \mathbb{E}[\exp[-zA_v]] \right]^N \\
&= \left[\inf_{z<0} \left[\sum_{m=1}^M \sum_{\ell \neq m} \Pr(\sigma_v(i) = m, \sigma_v(j) = \ell) \exp[z(\beta(\sigma_v(i)) - \beta(\sigma_v(j)))] \right] \right]^N \\
&= \left[\inf_{z<0} \exp[\Lambda(z)] \right]^N \\
&= \exp[-r_{ij}(\beta)N]
\end{aligned}$$

Then, $\Pr(\sigma^N(i) < \sigma^N(j)) > 1 - \epsilon$ when

$$\begin{aligned}
\exp[-r_{ij}N] &< \epsilon \\
\iff N &> \frac{1}{r_{ij}} \log\left(\frac{1}{\epsilon}\right)
\end{aligned}$$

□

Consider β consistent with K -Approval voting for some fixed K , and candidates i, j such that $s_i > s_j$. Then the large deviation rate $r_{ij}(\beta)$ in Proposition 6.4.1 is

$$r_{ij}(K) = -\log\left(2\sqrt{t_{ij}^i(K)t_{ij}^j(K)} + 1 - t_{ij}^i(K) - t_{ij}^j(K)\right)$$

Where $t_{ij}^i(K) \triangleq \Pr_F(\sigma_v(i) \leq K, \sigma_v(j) > K)$, i.e., the probability that a voter approves i but not j .

Proof. With K -approval voting, A_v^{ij} becomes

$$A_v = \begin{cases} 1 & \text{w.p. } t_{ij}^i, \text{ i.e., when candidate } i \text{ approved but } j \text{ not approved} \\ 0 & \text{w.p. } 1 - t_{ij}^i - t_{ij}^j, \text{ i.e., when both approved, or neither approved} \\ -1 & \text{w.p. } t_{ij}^j, \text{ i.e., when candidate } j \text{ approved but } i \text{ not approved} \end{cases}$$

Then,

$$r_{ij} = - \inf_{z \in \mathbb{R}} \Lambda(z)$$

$$\Lambda(z) = \log \left[\sum_{m=1}^M \sum_{\ell \neq m} \Pr(\sigma_v(i) = m, \sigma_v(j) = \ell) \exp[z(\beta(\sigma_v(i)) - \beta(\sigma_v(j)))] \right]$$

$$= \log \left[t_{ij}^i \exp(z) + t_{ij}^j \exp(-z) + (1 - t_{ij}^i - t_{ij}^j) \right]$$

The $\inf(\Lambda)$ is attained at $z = \frac{1}{2} \log \frac{t_{ij}^j}{t_{ij}^i}$ (Λ is convex in z , and so setting the first derivative to zero finds the \inf). And so

$$r_{ij} = - \log \left[t_{ij}^i \exp \left(\frac{1}{2} \log \frac{t_{ij}^j}{t_{ij}^i} \right) + t_{ij}^j \exp \left(-\frac{1}{2} \log \frac{t_{ij}^j}{t_{ij}^i} \right) + (1 - t_{ij}^i - t_{ij}^j) \right]$$

$$= - \log \left[2 \sqrt{t_{ij}^i t_{ij}^j} + 1 - t_{ij}^i - t_{ij}^j \right]$$

□

Consider goal G and $\beta \in \mathcal{B}$ such that $O^N \rightarrow O^*$. Let Q^N be the expected number of errors in the outcome after N voters, $\sum_{i \in C_s^*, j \in C_t^*, s < t} \Pr(\sigma^N(i) > \sigma^N(j))$. Then Q^N goes to zero with large deviation rate

$$r(\beta) = \min_{i \in C_s^*, j \in C_t^*, s < t} r_{ij}(\beta)$$

Further, the following upper bound holds for any N .

$$Q^N \leq M^2 \exp(-rN)$$

Proof. By the Union bound

$$Q^N = \sum_{i \in C_s^*, j \in C_t^*, s < t} \Pr(\sigma^N(i) > \sigma^N(j))$$

$$\leq \sum_{i \in C_s^*, j \in C_t^*, s < t} \exp[-r_{ij}N] \quad \text{Proposition 6.4.1}$$

$$\leq M^2 \exp[-rN]$$

Now, using large deviation properties:

By supposition, $Q^N \rightarrow 0$, and so $-Q^N$ approaches 0 from below. Then,

$$\begin{aligned}
-\lim_{N \rightarrow \infty} \frac{1}{N} \log(Q^N) &= -\lim_{N \rightarrow \infty} \frac{1}{N} \log \sum_{i \in C_s^*, j \in C_t^*, s < t} \Pr(\sigma^N(i) > \sigma^N(j)) \\
&= -\max_{i \in C_s^*, j \in C_t^*, s < t} \left(\lim_{N \rightarrow \infty} \frac{1}{N} \log \Pr(\sigma^N(i) > \sigma^N(j)) \right) \\
&= \min_{i \in C_s^*, j \in C_t^*, s < t} - \left(\lim_{N \rightarrow \infty} \frac{1}{N} \log \Pr(\sigma^N(i) > \sigma^N(j)) \right) \\
&= \min_{i \in C_s^*, j \in C_t^*, s < t} r_{ij} = r
\end{aligned} \tag{E.1}$$

Line (E.1) follows from: $\forall a_i^\epsilon \geq 0$, $\limsup_{\epsilon \rightarrow 0} \left[\epsilon \log \left(\sum_i^N a_i^\epsilon \right) \right] = \max_i^N \limsup_{\epsilon \rightarrow 0} \epsilon \log(a_i^\epsilon)$. See, e.g., Lemma 1.2.15 in Dembo and Zeitouni (2010) for a proof of this property.

Thus r is the large deviation rate for Q^N .

□

E.2.3 Design insights

Theorem 6.5.1. *Randomization does not improve the outcome learning rate for any asymptotically design-invariant noise model F or goal G . For any randomized scoring rule mechanism (B, D) , where $B \subset \mathcal{B}$, for any F, G , the scoring rule $\beta^*(k) = \sum_p d_p \beta_p(k)$ satisfies $r(\beta^*) \geq r(B, D)$.*

Proof. From Proposition 6.4.1, for a given scoring rule β and pair of candidates i, j , the learning rate is

$$\begin{aligned}
r_{ij}(\beta) &= -\inf_{z \in \mathbb{R}} \log \mathbb{E}_F [\exp [z [\beta(\sigma_v(i)) - \beta(\sigma_v(j))]]] \\
&= -\log \inf_{z \in \mathbb{R}} \mathbb{E}_F [\exp [z [\beta(\sigma_v(i)) - \beta(\sigma_v(j))]]]
\end{aligned}$$

Similarly, if we use scoring rules $\{\beta^u\}_{u=1}^P$, each with probability d^u , then,

$$\begin{aligned}
r_{ij}(\{\beta^u\}_{u=1}^P) &= -\log \inf_{z \in \mathbb{R}} \mathbb{E}_{F, \{d^u, \beta^u\}} [\exp [z [\beta^u(\sigma_v(i)) - \beta^u(\sigma_v(j))]]] \\
&= -\log \inf_{z \in \mathbb{R}} \sum_u d^u \mathbb{E}_F [\exp [z [\beta^u(\sigma_v(i)) - \beta^u(\sigma_v(j))]]]
\end{aligned}$$

Now, for a single scoring rule $\beta(\cdot)$, let

$$\gamma(\beta(1), \dots, \beta(M)) \triangleq \mathbb{E}_F [\exp [z [\beta(\sigma_v(i)) - \beta(\sigma_v(j))]]]$$

Below, we show that $\gamma(\beta(1), \dots, \beta(M))$ is convex in $\beta(k), \forall k, z$. Then, by convexity, $\forall z$

$$\sum_{u=1}^P d^u \gamma(\beta^u(1), \dots, \beta^u(M)) \geq \gamma\left(\sum_{u=1}^P d^u \beta^u(1), \dots, \sum_{u=1}^P d^u \beta^u(M)\right)$$

and so

$$\inf_z \left[\sum_{u=1}^P d^u \gamma(\beta^u(1), \dots, \beta^u(M)) \right] \geq \inf_z \gamma\left(\sum_{u=1}^P d^u \beta^u(1), \dots, \sum_{u=1}^P d^u \beta^u(M)\right)$$

The left hand side is equal to the argument inside the $-\log(\cdot)$ for the rate function for randomizing between scoring rules $\{\beta^u\}_{u=1}^P$, each with probability d^u , and the right hand side is the argument inside for the rate function for instead using the single scoring rule β^* defined as the convex combination of $\{\beta^u\}_{u=1}^P$. Then, as $-\log(x)$ is decreasing in x , we have that

$$r_{ij}(\beta^*) \geq r_{ij}(\{\beta^u\}_{u=1}^P)$$

As this holds for each pair of candidates i, j simultaneously, we are done.

Proof that $\gamma(\beta(1), \dots, \beta(M)) \triangleq \mathbb{E}_F[\exp[z[\beta(\sigma_v(i)) - \beta(\sigma_v(j))]]]$ is convex in $\beta(k), \forall k$ We directly calculate the Hessian of γ and note that it is diagonally dominant and thus positive semidefinite. For notational convenience, we let $\beta_k = \beta(k)$, and $\sigma(k, \ell) = \Pr_F(\sigma_v(i) = k, \sigma_v(j) = \ell)$. Of course, $\sigma(k, k) = 0$, as we assume each voter has a strict ranking as her preference.

$$\begin{aligned} \gamma(\beta_1, \dots, \beta_M) &= \mathbb{E}_F[\exp[z[\beta(\sigma_v(i)) - \beta(\sigma_v(j))]]] \\ &= \sum_{k=1}^M \sum_{\ell=1}^M \sigma(k, \ell) \exp[z[\beta_k - \beta_\ell]] \\ \frac{\partial}{\partial \beta_k} \gamma(\beta_1, \dots, \beta_M) &= z \exp[z\beta_k] \sum_{\ell \neq k} \exp[-z\beta_\ell] \sigma(k, \ell) - z \exp[-z\beta_k] \sum_{\ell \neq k} \exp[z\beta_\ell] \sigma(\ell, k) \\ \frac{\partial^2}{\partial (\beta_k)^2} \gamma(\beta_1, \dots, \beta_M) &= z^2 \exp[z\beta_k] \sum_{\ell \neq k} \exp[-z\beta_\ell] \sigma(k, \ell) + z^2 \exp[-z\beta_k] \sum_{\ell \neq k} \exp[z\beta_\ell] \sigma(\ell, k) \\ &= z^2 \left[\left[\exp[z\beta_k] \sum_{\ell \neq k} \exp[-z\beta_\ell] \sigma(k, \ell) \right] + \left[\exp[-z\beta_k] \sum_{\ell \neq k} \exp[z\beta_\ell] \sigma(\ell, k) \right] \right] \\ \frac{\partial^2}{\partial \beta_k \partial \beta_\ell} \gamma(\beta_1, \dots, \beta_M) &= -z^2 [\exp[z\beta_k] \exp[-z\beta_\ell] \sigma(k, \ell) + \exp[-z\beta_k] \exp[z\beta_\ell] \sigma(\ell, k)] \end{aligned}$$

Thus, the Hessian of γ is diagonally dominant with non-negative diagonal elements: $\forall k$,

$$\left| \frac{\partial^2 \gamma}{\partial (\beta_k)^2} \right| \geq \sum_{\ell \neq k} \left| \frac{\partial^2 \gamma}{\partial \beta_k \partial \beta_\ell} \right|$$

and so the Hessian is positive semi-definite. Thus, γ is convex in β_k .

□

Theorem 6.5.2. *Randomization amongst K -Approval mechanisms does not improve the learning rate for separating a given pair of candidates i, j for any asymptotically design-invariant noise model F or goal G . For any randomized K -Approval mechanism (B, D) , where $\beta_p \in B$ corresponds to p -Approval, for any F, G , there exists a mechanism K_{ij}^* -Approval such that $r_{ij}(K_{ij}^*) \geq r_{ij}(B, D)$.*

Proof. From Proposition 6.4.1, for k -Approval,

$$r_{ij}(t_{ij}^i, t_{ij}^j) = -\log \left[2\sqrt{t_{ij}^i t_{ij}^j} + 1 - t_{ij}^i - t_{ij}^j \right]$$

Where t_{ij}^i is the probability that i is approved but j is not.

This rate function is convex in t_{ij}^i, t_{ij}^j :

- $r_{ij}(a, b) = h(g(a, b))$, where $h(x) = -\log(x)$, $g(a, b) = 2\sqrt{ab} + 1 - a - b$.
- $g(a, b)$ is concave in a, b
- h is convex, and \tilde{h} is non-increasing, where $\tilde{h}(x) = \begin{cases} h(x) & x > 0 \\ \infty & x \leq 0 \end{cases}$ is the extended value function of h .
- By convex composition rules, $r_{ij}(a, b)$ is convex (see, e.g., page 84 of Boyd and Vandenberghe (2004)).

The result follows by convexity. Consider a randomization of K -Approval mechanisms for $K \in \{1, \dots, M-1\}$, where K -Approval is used with probability d^K .

The resulting approval probabilities are: $t_{ij}^i = \sum_{K=1}^{M-1} d^K t_{ij}^i(K)$, $t_{ij}^j = \sum_{K=1}^{M-1} d^K t_{ij}^j(K)$. By convexity:

$$\begin{aligned} r \left(\sum_{K=1}^{M-1} d^K t_{ij}^i(K), \sum_{K=1}^{M-1} d^K t_{ij}^j(K) \right) &\leq \sum_{K=1}^{M-1} d^K r \left(t_{ij}^i(K), t_{ij}^j(K) \right) \\ &= \sum_{K=1}^{M-1} d^K r_{ij}(K) \\ &\leq \max_K r_{ij}(K) \end{aligned}$$

We note that, unlike the previous proof, we cannot conclude in general that randomization cannot improve the rates at which the outcome is learned (in fact, Theorem 6.5.3 establishes otherwise). That is because while the same β^* could be said to be rate optimal (compared to the randomized mechanism) for every pair of candidates simultaneously in that proof, in this proof $\arg \max_K r_{ij}(K)$ may change based on the pair i, j .

□

Corollary 6.5.1. *Randomization among K -Approval mechanisms does not improve the learning rate for selecting W winners from the Mallows model. For any randomized K -Approval mechanism (B, D) , where $\beta_p \in B$ corresponds to p -Approval, for selecting W winners from the Mallows model, there exists an Approval rate optimal mechanism K^* -Approval such that $r(K^*) \geq r(B, D)$.*

Proof. When selecting W winners out of M candidates, we need to separate candidates $1 \dots W$ from candidates $W + 1 \dots M$. It is easy to show that the pivotal pair, regardless of which K is used in K -Approval, is $W, W + 1$. Applying Theorem 6.5.2, then, randomization cannot help the overall rate. □

Theorem 6.5.4. *Under the Mallows model and the goal of selecting W winners, W -Approval may not be Approval rate optimal.*

Proof. We prove the result by providing an example where it is not optimal. Suppose there are 4 candidates, and we wish to select 3 winners, i.e., separate the first three items from the last item.

Let the items in the reference ranking be, in order, $i = 1, 2, 3, 4$, respectively.

A Mallows model (with parameter $\phi = \frac{p}{1-p}$, where p is the probability of flipping a given pair of candidates) can be sampled by repeated insertion (Diaconis, 1988; Lu and Boutilier, 2011): starting from the first item in the reference ranking, there exists probability $\tilde{p}_{ij} = \frac{\phi^{i-j}}{1+\phi^1+\dots+\phi^{i-1}}$ at which item i can be inserted into position $j \leq i$, *independently* of how items above it were inserted, such that the resulting ranking distribution matches the Mallows model.

Using this repeated insertion property for our example, we can derive $p_{\ell k}$, the probability at which item 3 is in position ℓ and item 4 is in position k after sampling from a Mallows model with parameter ϕ .

In particular, if $\ell < k$, $p_{\ell k}$ is exactly the probability that item 3 is inserted in position ℓ and item 4 is inserted in position k . If $k > \ell$, however, it is the probability that item 3 is inserted in position $\ell - 1$ and then pushed down when item 4 is inserted in position k . (More generally, it turns out, the exact probability for an item appearing in a given position in the Mallows model can be calculated using a simple dynamic program, a fact that does not appear to be documented elsewhere but may be independently useful. We used this dynamic program to find this given example).

Then, for our example

$$N_i \triangleq 1 + \phi^1 + \dots + \phi^{i-1}$$

$$p_{\ell k} = \begin{bmatrix} 0 & \frac{\phi^2 \phi^2}{N_3 N_4} & \frac{\phi^2 \phi^1}{N_3 N_4} & \frac{\phi^2 \phi^0}{N_3 N_4} \\ \frac{\phi^2 \phi^3}{N_3 N_4} & 0 & \frac{\phi^1 \phi^1}{N_3 N_4} & \frac{\phi^1 \phi^0}{N_3 N_4} \\ \frac{\phi^1 \phi^3}{N_3 N_4} & \frac{\phi^1 \phi^2}{N_3 N_4} & 0 & \frac{\phi^0 \phi^0}{N_3 N_4} \\ \frac{\phi^0 \phi^3}{N_3 N_4} & \frac{\phi^0 \phi^2}{N_3 N_4} & \frac{\phi^0 \phi^1}{N_3 N_4} & 0 \end{bmatrix}_{\ell k} = \frac{1}{N_3 N_4} \begin{bmatrix} 0 & \phi^4 & \phi^3 & \phi^2 \\ \phi^5 & 0 & \phi^2 & \phi^1 \\ \phi^4 & \phi^3 & 0 & 1 \\ \phi^3 & \phi^2 & \phi^1 & 0 \end{bmatrix}_{\ell k}$$

Recall that $t_{ij}^i(K)$ is the probability that i is approved but j is not, using K -Approval. Then, if we use 3-approval and 2-approval, respectively:

$$N_3 = 1 + \phi + \phi^2$$

$$N_4 = 1 + \phi + \phi^2 + \phi^3$$

$$t_{34}^3(3) = p_{14} + p_{24} + p_{34} = \frac{\phi^2 + \phi^1 + 1}{N_3 N_4} = \frac{1}{N_4}$$

$$t_{34}^4(3) = p_{41} + p_{42} + p_{43} = \frac{\phi^3 + \phi^2 + \phi^1}{N_3 N_4} = \frac{\phi}{N_4}$$

$$t_{34}^3(2) = p_{14} + p_{24} + p_{13} + p_{23} = \frac{\phi^2 + \phi^1 + \phi^3 + \phi^2}{N_3 N_4}$$

$$t_{34}^4(2) = p_{41} + p_{42} + p_{31} + p_{32} = \frac{\phi^3 + \phi^2 + \phi^4 + \phi^3}{N_3 N_4}$$

Then, recall the rate between items i, j using K approval is

$$r_{ij}(K) = -\log \left[2\sqrt{t_{ij}^i(K)t_{ij}^j(K)} + 1 - t_{ij}^i(K) - t_{ij}^j(K) \right]$$

$$r_{34}(3) = -\log \left[2\sqrt{\frac{1}{N_4} \frac{\phi}{N_4}} + 1 - \frac{1}{N_4} - \frac{\phi}{N_4} \right]$$

$$r_{34}(2) = -\log \left[2\sqrt{\frac{\phi^2 + \phi^1 + \phi^3 + \phi^2}{N_3 N_4} \frac{\phi^3 + \phi^2 + \phi^4 + \phi^3}{N_3 N_4}} + 1 - \frac{\phi^2 + \phi^1 + \phi^3 + \phi^2}{N_3 N_4} - \frac{\phi^3 + \phi^2 + \phi^4 + \phi^3}{N_3 N_4} \right]$$

When there is low noise, e.g., $\phi = .1$ ($p = .091$):

$$r_{34}(3) = .5462$$

$$r_{34}(2) = .04696 < r_{34}(3)$$

But when there is high noise, e.g., $\phi = .8$ ($p = .44$):

$$\begin{aligned} r_{34}(3) &= .00378 \\ r_{34}(2) &= .00402 > r_{34}(3) \end{aligned}$$

Note that the same example works for selecting 1 winner out of the 4 candidates, as the repeated insertion model can be run in reverse.

□

Theorem 6.5.3. *Randomization among K -Approval mechanisms may improve the learning rate for the goal of selecting W winners. There exist asymptotically design-invariant settings (M, F) for the goal of selecting W winners such that a randomized K -Approval mechanism (B, D) , where $\beta_p \in B$ corresponds to p -Approval, satisfies*

$$r(B, D) > \max_K r(K)$$

Proof. We provide two proofs: a numeric example from a real-world election, and a contrived, constructed example.

Numeric example found in a real election In Durham Ward 1, to select 2 winners, randomizing between 3 and 4-Approval is better than either individually, even though asymptotically the mechanisms pick the same set of winners. The critical pair with 2-Approval is with the candidate asymptotically ranked 1st, and the best item not selected. With 3-Approval, it is with the candidate asymptotically ranked 2nd, and the same best item not selected.

We will call these items h, i, j (the one not selected) respectively. The respective probabilities of being selected alone:

$$t_{hj}^h(3) = 0.277$$

$$t_{hj}^h(4) = 0.266$$

$$t_{hj}^j(3) = 0.200$$

$$t_{hj}^j(4) = 0.188$$

$$t_{ij}^i(3) = 0.255$$

$$t_{ij}^i(4) = 0.295$$

$$t_{ij}^j(3) = 0.160$$

$$t_{ij}^j(4) = 0.217$$

$$t_{hj}^h(\{3, 4\}) = 0.271$$

$$t_{hj}^j(\{3, 4\}) = 0.194$$

$$t_{ij}^i(\{3, 4\}) = 0.275$$

$$t_{ij}^j(\{3, 4\}) = 0.189$$

And the resulting rates (using the formula in Proposition 6.4.1) are:

$$r_{hj}(3) = .00616932$$

$$r_{ij}(3) = .01114061$$

$$r_{hj}(4) = .00677352$$

$$r_{ij}(4) = .00592327$$

$$r_{hj}(\{3, 4\}) = .00642839$$

$$r_{ij}(\{3, 4\}) = .00815633$$

$$r(3) = \min(r_{hj}(3), r_{ij}(3)) = .00616932$$

$$r(4) = \min(r_{hj}(4), r_{ij}(4)) = .00592327$$

$$r(\{3, 4\}) = \min(r_{hj}(\{3, 4\}), r_{ij}(\{3, 4\})) = .00642839 > \max(r(3), r(4))$$

Thus randomization improves learning.

Constructed example with design invariance We now construct a fully design-invariant example with the same flavor as the numeric example, where which pair is critical changes with the

mechanism.

Consider three candidates h, i, j , such that h is asymptotically in the set of W winners and i, j are not. Thus, we need the rates at which h is separated from both i, j . Let $W = K < L = K + 1$.

We prove the result by giving an example where: it is easier to separate h from i using K -Approval, and easier to separate h from j using L -Approval. Using K -Approval, r_{hj} asymptotically dominates the rate at which the overall outcome is learned, and using L -Approval, r_{hi} does. Further, randomizing between the two mechanisms improves the two rates that dominate enough such that the overall rate is improved.

We need to show the following hold for our example: one of the rates between candidates h and i, j are smaller than other rates, i.e., dominate the overall learning rate when K and/or L approval is used; randomization between K and L approval helps the minimum rate between candidates h and i, j ; K' -Approval ($K' \neq K, K' \neq L$) produces a worse rate than either K or L approval; and this example is asymptotically design-invariant.

We prove each of these conditions in turn after specifying the example.

Recall that $t_{ij}^i(k)$ is the probability that i is approved but j is not, using k -Approval. Here, we will use:

$$t_{hi}^i(K), t_{hi}^i(L), t_{hi}^h(K), t_{hi}^h(L), t_{hj}^j(K), t_{hj}^j(L), t_{hj}^h(K), t_{hj}^h(L)$$

. The end row labeled “Total value” then sums up these values.

Example Specification Consider F such according to the following table, where the first column is the probabilities of the positions in the second set of columns. The third set of columns indicates whether those set of positions contribute to the given probabilities, for easy accounting.

Row	$Pr_F(\cdot)$	Positions of h, i, j			Contributes to? (Y = Yes)							
		$\sigma(h)$	$\sigma(i)$	$\sigma(j)$	$t_{hi}^h(K)$	$t_{hi}^i(K)$	$t_{hi}^h(L)$	$t_{hi}^i(L)$	$t_{hj}^h(K)$	$t_{hj}^i(K)$	$t_{hj}^h(L)$	$t_{hj}^i(L)$
1	a	K	$L + 1$	L	Y		Y		Y			
2	a	$K - 1$	K	L					Y			
3	$T_1 - a - \epsilon$	K	$L + 1$	$L + 2$	Y		Y		Y		Y	
4	$T_2 - 2a$	$L + 1$	$L + 2$	K						Y		Y
5	$T_2 - 2a$	$L + 1$	K	$L + 2$		Y		Y				
6	a	$L + 1$	K	L		Y		Y				Y
7	a	L	$K - 1$	K		Y				Y		
8	a	L	$L + 1$	$L + 2$			Y				Y	
9	a	$L + 1$	$L + 2$	L								Y
10	ϵ	See caption			Y		Y		Y		Y	
11	0	Otherwise										
Total value:					T_1	T_2	$T_1 + a$	$T_2 - a$	$T_1 + a$	$T_2 - a$	T_1	T_2

Table E.4: Where the constants such that $0 < \epsilon < a < \frac{T_2}{2} < T_2 < T_1 < T_1 + 2T_2 + a = 1$, i.e., the table describes a valid probability distribution. Row 10 is as follows: The first K candidates (the asymptotic winners) occupy the first K spots, in an order drawn uniformly at random. Similarly, The bottom $M - K$ candidates occupy the bottom $M - K$ spots, in an order drawn uniformly at random. This randomization ensures asymptotically design invariance.

The table does not specify the probabilities of other candidates appearing in any position, so it is possible that they dominate the learning rate (are hardest to learn). (In particular, if the same, asymptotically non-winning candidate q is always in position L in the case in row 3, then it may be hard to separate it from candidate h using L approval). However, we can specify the example further to ensure this does not happen.

Suppose candidates are indexed by their order in some strict ranking σ^* . Then, candidates $h = K, i = L = K + 1, j = K + 2$. Further suppose that candidates in $\{1 \dots K - 1\}$ always occupy, in order except in case of row 10, the best positions in a voter’s ranking that are not reserved for candidates h, i, j in the table above.

For candidates $q \in \{K + 3 \dots M\}$, we have to be more careful to avoid the case in parenthesis above. Suppose these $Q = M - K + 2$ candidates fill up the bottom spots in a voter’s ranking in a uniform at random order. In other words, they occupy spots $L + 3 \dots M$, and the worst spot among whichever of $K, L, L + 1, L + 2$ is missing in each row in the table above.

Rates between the h and i, j dominate the overall learning rate using K or L approval

We are now ready to show the first claim that learning between candidates h and i, j is hardest (when using either K or L approval).

By the specification above, candidates in $\{1 \dots K - 2\}$ are always approved, and so learning between those candidates and any non-winning candidate is faster than any large deviations rate. Similarly, candidate $K - 1$ always is ranked higher than candidates $q \in \{K + 3 \dots M\}$, and it is approved alone with high enough probability.

Then, the other candidates who may dominate the learning rate are candidate $K-1$ (in separation from i, j), or $q \in \{K+3 \dots M\}$ (in separation from h). From the above table:

$t_{hq}^h(K) = T_1 + a$	Rows 1,2,3,10
$t_{hq}^q(K) = \frac{2a}{Q}$	Rows 8,9
$t_{hq}^h(L) = \frac{Q-1}{Q} [T_1 - \epsilon] + 3a + \epsilon$	Rows 1,2,7,10; and 3,8 w.p. $\frac{Q-1}{Q}$
$t_{hq}^q(L) = \frac{2T_2 - 3a}{Q}$	Rows 4,5,9 w.p. $\frac{1}{Q}$
$t_{(K-1)i}^{K-1}(K) = T_1 + T_2$	Rows 1, 3, 4, 8, 9, 10
$t_{(K-1)i}^i(K) = 2a$	Rows 2, 7
$t_{(K-1)j}^{K-1}(K) = T_1 + T_2 + a$	Rows 1, 3, 5, 6, 8, 9, 10
$t_{(K-1)j}^j(K) = a$	Rows 7
$t_{(K-1)i}^{K-1}(L) = T_1 + T_2$	Rows 1, 3, 4, 8, 9, 10
$t_{(K-1)i}^i(L) = 2a$	Rows 2, 7
$t_{(K-1)j}^{K-1}(L) = T_1 + T_2 - 2a$	Rows 3, 5, 8, 10
$t_{(K-1)j}^j(L) = 2a$	Rows 2, 7

Now, suppose $3a > \frac{T_1}{Q}$ and $Q > 2$. (Both conditions occur for Q large enough). Then, applying Remark E.2.2 regarding learning rates being larger when the arguments are farther away from one another (holding one fixed), the resulting rates with these candidates are dominated by (larger than) the rates between candidates h and i, j , discussed next.

Randomizing improves the minimum rate between candidates h and i, j By Remark E.2.2,

$$r(T_1 + a, T_2 - a) > r(T_1, T_2)$$

Using K -Approval:

Rate between h, i :	$r_{hi}(K) = r(T_1, T_2)$
Rate between h, j :	$r_{hj}(K) = r(T_1 + a, T_2 - a)$
Overall rate:	$r(K) = \min(r_{hi}(K), r_{hj}(K)) = r(T_1, T_2)$

Using L -Approval:

$$\begin{aligned} \text{Rate between } h, i: & \quad r_{hi}(L) = r(T_1 + a, T_2 - a) \\ \text{Rate between } h, j: & \quad r_{hj}(L) = r(T_1, T_2) \\ \text{Overall rate:} & \quad r(K) = \min(r_{hi}(L), r_{hj}(L)) = r(T_1, T_2) \end{aligned}$$

Randomizing – For any $0 < p < 1$, eliciting K -Approval with probability p , and L -Approval otherwise:

$$\begin{aligned} \text{Rate between } h, i: & \quad r(T_1 + (1 - p)a, T_2 - (1 - p)a) \\ \text{Rate between } h, j: & \quad r(T_1 + pa, T_2 - pa) \\ \text{Overall rate:} & \quad r(K) = r(T_1 + \phi a, T_2 - \phi a) & \quad \phi = \min(p, 1 - p) \\ & > r(T_1, T_2) & \quad \text{Remark E.2.2} \end{aligned}$$

K' -Approval ($K' \neq K, K' \neq L$) produces a worse rate than either K or L approval For any $K' < K - 1$, h is approved with probability $\epsilon_2 < \epsilon$, and i, j are never approved. Then, the rate between h and i, j is $-\log(1 - \epsilon_2) \rightarrow 0$ as $\epsilon \rightarrow 0$. Identically, for $K' \geq L + 2 = K + 3$, both h and i, j are approved except with some probability $\epsilon_2 < \epsilon$.

For $K' = K - 1$, h is approved without i with probability $a + \epsilon_2$ (for some $\epsilon_2 < \epsilon$), and i is approved without h with probability a . Then, the rate between h and i is $-\log(2\sqrt{(a + \epsilon_2)a} + 1 - 2a - \epsilon_2) \rightarrow 0$ as $\epsilon \rightarrow 0$.

For $K' = K + 2 = L + 1$, h is approved without i with probability $T_2 - a + \epsilon_2$, and i is approved without h with probability 0. Then, the rate between them is $-\log(1 - T_2 + a - \epsilon_2)$. For T_2 small enough, this is a worse rate than using K or L approval.

The example described is asymptotically design-invariant From the above table, the probability that candidate $c \in \{1 \dots M\}$ is in position k or better, i.e., $\sigma(c) \leq k$ is:

Candidate	$k < K - 1$	$K - 1$	K	$L = K + 1$	$K + 2$	$K + 3$	$M > k > K + 3$
$w \in \{1 \dots K - 2\}$	> 0	$> 1 - \epsilon$	1	1	1	1	1
$K - 1$	> 0	$1 - 2a$	$1 - 2a$	$1 - 2a$	1	1	1
K	> 0	$> a$	$T_1 + a$	$T_1 + 3a$	1	1	1
$L = K + 1$	0	a	T_2	$< T_2 + \epsilon$	< 1	< 1	< 1
$K + 2$	0	0	$T_2 - a$	$< T_2 + 3a + \epsilon$	< 1	< 1	< 1
$q \in \{K + 3 \dots M\}$	0	0	$\frac{2a}{Q}$	$\frac{T_1 + 2T_2 - 3a - \epsilon}{Q}$	< 1	< 1	< 1

Conditions on constants in problem For the above claims to hold, we set conditions on the constants in the problem. They are

$$\begin{aligned}
0 < \epsilon < a < \frac{T_2}{2} < T_2 < T_1 < T_1 + 2T_2 + a = 1 \\
1 - \epsilon > 2\sqrt{T_1T_2} + 1 - T_1 - T_2 \\
\frac{T_1 + 2T_2 - 3a - \epsilon}{Q} < T_1 + 3a \\
3a > \frac{T_1}{Q} \\
Q > 2 \\
1 - T_2 + a - \epsilon > 2\sqrt{T_1T_2} + 1 - T_1 - T_2
\end{aligned}$$

This is a feasible set of constraints: Q can be set large enough to meet conditions 3,4,5 for any fixed T_1, a, T_2 that meet condition 1. Condition 2 is weaker than the last condition. That leaves the last condition along with the first one.

$$\begin{aligned}
1 - T_2 + a - \epsilon > 2\sqrt{T_1T_2} + 1 - T_1 - T_2 \\
\iff a - \epsilon > 2\sqrt{T_1T_2} - T_1 \\
\iff \frac{T_2}{2} - 2\epsilon > 2\sqrt{T_1T_2} - T_1 & \quad \text{set } \frac{T_2}{2} - \epsilon = a \\
\iff \frac{T_2}{2} + T_1 > 2\epsilon + 2\sqrt{T_1T_2}
\end{aligned}$$

which holds for T_1 large enough, and T_2, ϵ small enough.

□

Bibliography

- Atila Abdulkadiroğlu, Parag A Pathak, and Alvin E Roth. The New York City High School Match. *American Economic Review*, 95(2):364–367, 2005.
- Rediet Abebe, Solon Barocas, Jon Kleinberg, Karen Levy, Manish Raghavan, and David G Robinson. Roles for Computing in Social Change. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 252–260, 2020.
- Daron Acemoglu, Ali Makhdoui, Azarakhsh Malekian, and Asuman Ozdaglar. Fast and Slow Learning From Reviews. Working Paper 24046, National Bureau of Economic Research, November 2017. URL <http://www.nber.org/papers/w24046>.
- Philipp Afèche, Zhe Liu, and Costis Maglaras. Ride-Hailing Networks With Strategic Drivers: The Impact of Platform Control Capabilities on Performance. *SSRN Electronic Journal*, 2018. ISSN 1556-5068. doi: 10.2139/ssrn.3120544. URL <https://www.ssrn.com/abstract=3120544>.
- Stéphane Airiau and Ulle Endriss. Iterated Majority Voting. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory*, ADT '09, pages 38–49, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-04427-4. doi: 10.1007/978-3-642-04428-1_4. URL http://dx.doi.org/10.1007/978-3-642-04428-1_4.
- Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion Fraud Detection in Online Reviews by Network Effects. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.
- Carlos Alós-Ferrer and Dura-Georg Granić. Two Field Experiments on Approval Voting in Germany. *Social Choice and Welfare*, 39(1):171–205, 2012.
- Christina Aperjis and Ramesh Johari. Optimal Windows for Aggregating Ratings in Electronic Marketplaces. *Management Science*, 56(5):864–880, 2010.
- Charles Arthur. Marissa Mayer’s Appointment: What Does It Mean for Yahoo? *The Guardian*, July 2012. ISSN 0261-3077. URL <https://www.theguardian.com/technology/2012/jul/16/marissa-mayer-appointment-mean-yahoo>.

- Arash Asadpour, Daniel Freund, and Garrett J. van Ryzin. Escrow Payments: A Smoother Driver Pay Mechanism, October 2019. URL <https://www.abstractsonline.com/pp8/#!/6818/presentation/7365>.
- Itai Ashlagi, Maximilien Burq, Patrick Jaillet, and Amin Saberi. Maximizing Efficiency in Dynamic Matching Markets. *arXiv preprint arXiv:1803.01285*, 2018. URL <https://arxiv.org/pdf/1803.01285.pdf>.
- Baris Ata, Nasser Barjesteh, and Sunil Kumar. Spatial Pricing: An Empirical Analysis of Taxi Rides in New York City. *Working Paper*, 2019.
- Susan Athey and Michael Luca. Economists (and Economics) in Tech Companies. *Journal of Economic Perspectives*, 33(1):209–30, 2019.
- Shane Auerbach. Paying Rideshare Drivers for Pickups, October 2019. URL <https://simons.berkeley.edu/talks/tbd-78>.
- Haris Aziz, Serge Gaspers, Joachim Gudmundsson, Simon Mackenzie, Nicholas Mattei, and Toby Walsh. Computational Aspects of Multi-Winner Approval Voting. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 107–115. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
- Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. Justified Representation in Approval-Based Committee Voting. *Social Choice and Welfare*, 48(2):461–485, 2017.
- Jiaru Bai, Kut C. So, Christopher S. Tang, Xiquan (Michael) Chen, and Hai Wang. Coordinating Supply and Demand on an On-Demand Service Platform With Impatient Customers. *Manufacturing & Service Operations Management*, June 2018. doi: 10.1287/msom.2018.0707.
- Siddhartha Banerjee, Carlos Riquelme, and Ramesh Johari. Pricing in Ride-Share Platforms: A Queueing-Theoretic Approach. *SSRN Electronic Journal*, 2015. ISSN 1556-5068. doi: 10.2139/ssrn.2568258. URL <http://www.ssrn.com/abstract=2568258>.
- Siddhartha Banerjee, Daniel Freund, and Thodoris Lykouris. Pricing and Optimization in Shared Vehicle Systems: An Approximation Framework. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 517–517. ACM, 2017a.
- Siddhartha Banerjee, Sreenivas Gollapudi, Kostas Kollias, and Kamesh Munagala. Segmenting Two-Sided Markets. In *Proceedings of the 26th International Conference on World Wide Web*, pages 63–72, 2017b.

- Siddhartha Banerjee, Yash Kanoria, and Pengyu Qian. State Dependent Control of Closed Queueing Networks With Application to Ride-Hailing. March 2018. URL <http://arxiv.org/abs/1803.04959>.
- Gerdus Benade, Nevo Itzhak, Nisarg Shah, and Ariel D Procaccia. Efficiency and Usability of Participatory Budgeting Methods. 2018.
- Daniel Benjamin, Ori Heffetz, Miles Kimball, and Derek Lougee. The Relationship Between the Normalized Gradient Addition Mechanism and Quadratic Voting. *Public Choice*, 172(1):233–263, July 2017. ISSN 1573-7101. doi: 10.1007/s11127-017-0414-3. URL <https://doi.org/10.1007/s11127-017-0414-3>.
- Daniel J. Benjamin, Ori Heffetz, Miles S. Kimball, and Nichole Szembrot. Aggregating Local Preferences to Guide Marginal Policy Adjustments. *The American Economic Review*, 103(3):605–610, May 2013. ISSN 0002-8282. doi: 10.1257/aer.103.3.605. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3760035/>.
- Dimitris J. Bertsimas and Garrett van Ryzin. A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane. *Operations Research*, 39(4):601–615, August 1991. ISSN 0030-364X, 1526-5463. doi: 10.1287/opre.39.4.601.
- Dimitris J. Bertsimas and Garrett van Ryzin. Stochastic and Dynamic Vehicle Routing in the Euclidean Plane With Multiple Capacitated Vehicles. *Operations Research*, 41(1):60–76, February 1993. ISSN 0030-364X, 1526-5463. doi: 10.1287/opre.41.1.60.
- Omar Besbes and Marco Scarsini. On Information Distortions in Online Ratings. *Operations Research*, 66(3):597–610, June 2018. ISSN 0030-364X, 1526-5463. doi: 10.1287/opre.2017.1676.
- Omar Besbes, Francisco Castro, and Ilan Lobel. Spatial Capacity Planning. *SSRN Electronic Journal*, 2018a. ISSN 1556-5068. doi: 10.2139/ssrn.3292651. URL <https://www.ssrn.com/abstract=3292651>.
- Omar Besbes, Francisco Castro, and Ilan Lobel. Surge Pricing and Its Spatial Supply Response. *SSRN Electronic Journal*, 2018b. ISSN 1556-5068. doi: 10.2139/ssrn.3124571. URL <https://www.ssrn.com/abstract=3124571>.
- Sushil Bikhchandani. Intermediated Surge Pricing. *Journal of Economics & Management Strategy*, 29(1):31–50, 2020.
- Kostas Bimpikis, Ozan Candogan, and Daniela Saban. Spatial Pricing in Ride-Sharing Networks. (ID 2868080), November 2016. URL <https://papers.ssrn.com/abstract=2868080>.
- David Blum. Nine Potential Solutions to Abate Grade Inflation at Regionally Accredited Online US Universities: An Intrinsic Case Study. *The Qualitative Report*, 22(9):2288–2311, 2017.

- Gary Bolton, Ben Greiner, and Axel Ockenfels. Engineering Trust: Reciprocity in the Production of Reputation Information. *Management Science*, 59(2):265–285, 2013.
- George EP Box. Robustness in the Strategy of Scientific Model Building. In *Robustness in Statistics*, pages 201–236. Elsevier, 1979.
- Stephen Boyd and Almir Mutapcic. Subgradient Methods. *Lecture Notes of EE364b, Stanford University, Winter Quarter*, 2007.
- Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK ; New York, 2004. ISBN 978-0-521-83378-3.
- Nicholas Buchholz. Spatial Equilibrium, Search Frictions and Efficient Regulation in the Taxi Industry. 2017. URL https://scholar.princeton.edu/sites/default/files/nbuchholz/files/taxi_draft.pdf.
- Yves Cabannes. Participatory Budgeting: A Significant Contribution to Participatory Democracy. *Environment and Urbanization*, 16(1):27–46, April 2004. ISSN 0956-2478, 1746-0301. doi: 10.1177/095624780401600104. URL <http://eau.sagepub.com/content/16/1/27>.
- Luis Cabral and Ali Hortacsu. The Dynamics of Seller Reputation: Evidence From eBay. *The Journal of Industrial Economics*, 58(1):54–78, 2010.
- Luis Cabral and Lingfang Li. A Dollar for Your Thoughts: Feedback-Conditional Rebates on eBay. *Management Science*, 61(9):2052–2063, 2015.
- G rard P. Cachon, Kaitlin M. Daniels, and Ruben Lobel. The Role of Surge Pricing on a Service Platform With Self-Scheduling Capacity. *Manufacturing & Service Operations Management*, 19(3):368–384, June 2017. ISSN 1523-4614. doi: 10.1287/msom.2017.0618.
- Ioannis Caragiannis and Evi Micha. Learning a Ground Truth Ranking Using Noisy Approval Votes. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 149–155, Melbourne, Australia, August 2017. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-0-3. doi: 10.24963/ijcai.2017/22. URL <https://www.ijcai.org/proceedings/2017/22>.
- Ioannis Caragiannis and Ariel D. Procaccia. Voting Almost Maximizes Social Welfare Despite Limited Communication. *Artificial Intelligence*, 175(9):1655–1671, June 2011. ISSN 0004-3702. doi: 10.1016/j.artint.2011.03.005. URL <http://www.sciencedirect.com/science/article/pii/S0004370211000506>.
- Ioannis Caragiannis, Ariel D Procaccia, and Nisarg Shah. When Do Noisy Votes Reveal the Truth? In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, pages 143–160. ACM, 2013.

- Ioannis Caragiannis, Swaprava Nath, Ariel D Procaccia, and Nisarg Shah. Subset Selection via Implicit Utilitarian Voting. *Journal of Artificial Intelligence Research*, 58:123–152, 2017.
- Ioannis Caragiannis, Xenophon Chatzigeorgiou, George A Krimpas, and Alexandros A Voudouris. Optimizing Positional Scoring Rules for Rank Aggregation. *Artificial Intelligence*, 267:58–77, 2019.
- Juan Camilo Castillo, Dan Knoepfle, and Glen Weyl. Surge Pricing Solves the Wild Goose Chase. pages 241–242. ACM Press, 2017. ISBN 978-1-4503-4527-9. doi: 10.1145/3033274.3085098. URL <http://dl.acm.org/citation.cfm?doid=3033274.3085098>.
- Yeon-Koo Che and Johannes Horner. Optimal Design for Social Learning. 2015.
- Chuansheng Chen, Shin-ying Lee, and Harold W Stevenson. Response Style and Cross-Cultural Comparisons of Rating Scales Among East Asian and North American Students. *Psychological Science*, 6(3):170–175, 1995.
- M Keith Chen and Michael Sheldon. Dynamic Pricing in a Labor Market: Surge Pricing and Flexible Work on the Uber Platform. 2016. doi: 10.1145/2940716.2940798.
- Yiwei Chen and Ming Hu. Pricing and Matching With Forward-Looking Buyers and Sellers. SSRN Scholarly Paper ID 2859864, Social Science Research Network, Rochester, NY, July 2018. URL <https://papers.ssrn.com/abstract=2859864>.
- Yukun Cheng and Sanming Zhou. A Survey on Approximation Mechanism Design Without Money for Facility Games. In David Gao, Ning Ruan, and Wenxun Xing, editors, *Advances in Global Optimization*, number 95 in Springer Proceedings in Mathematics & Statistics, pages 117–128. Springer International Publishing, 2015. ISBN 978-3-319-08376-6 978-3-319-08377-3. URL http://link.springer.com/chapter/10.1007/978-3-319-08377-3_13. DOI: 10.1007/978-3-319-08377-3_13.
- Flavio Chierichetti and Jon Kleinberg. Voting With Limited Information and Many Alternatives. *SIAM Journal on Computing*, 43(5):1615–1653, 2014.
- Hun Chung and John Duggan. Directional Equilibria. *Journal of Theoretical Politics*, 30(3):272–305, July 2018. ISSN 0951-6298. doi: 10.1177/0951629818775515. URL <https://doi.org/10.1177/0951629818775515>.
- Cody Cook, Rebecca Diamond, Jonathan Hall, John List, and Paul Oyer. The Gender Earnings Gap in the Gig Economy: Evidence From Over a Million Rideshare Drivers. June 2018. doi: 10.3386/w24732. URL <https://www.nber.org/papers/w24732>.

- James Cook. Uber's Internal Charts Show How Its Driver-Rating System Actually Works, February 2015. URL <http://www.businessinsider.com/leaked-charts-show-how-ubers-driver-rating-system-works-2015-2>.
- Arthur H Copeland. A Reasonable Social Welfare Function. Technical report, mimeo, 1951. University of Michigan, 1951.
- Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, November 2012. ISBN 978-1-118-58577-1.
- Judd Cramer and Alan B. Krueger. Disruptive Change in the Taxi Business: The Case of Uber. *American Economic Review*, 106(5):177–182, May 2016. ISSN 0002-8282. doi: 10.1257/aer.p20161002.
- J. Pinto da Costa and L. Roque. Limit Distribution for the Weighted Rank Correlation Coefficient, Rw. *REVSTAT-Statistical Journal*, 4(3), 2006.
- Yuval Dagan, Yuval Filmus, Ariel Gabizon, and Shay Moran. Twenty (Simple) Questions. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 9–21, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4528-6. doi: 10.1145/3055399.3055422.
- Jean C de Borda. Mémoire Sur Les Élections Au Scrutin. 1781.
- Mathijs M de Weerd, Enrico H Gerding, and Sebastian Stein. Minimising the Rank Aggregation Error. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1375–1376. International Foundation for Autonomous Agents and Multiagent Systems, 2016.
- Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications*, volume 38 of *Stochastic Modelling and Applied Probability*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-03311-7. URL <http://link.springer.com/10.1007/978-3-642-03311-7>.
- Dorottya Demszky, Nikhil Garg, Rob Voigt, James Zou, Jesse Shapiro, Matthew Gentzkow, and Dan Jurafsky. Analyzing Polarization in Social Media: Method and Application to Tweets on 21 Mass Shootings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2970–3005, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1304. URL <https://www.aclweb.org/anthology/N19-1304>.
- Persi Diaconis. Group Representations in Probability and Statistics. *Lecture notes-monograph series*, 11:i–192, 1988.

- J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling. *IEEE Transactions on Automatic Control*, 57(3): 592–606, March 2012. ISSN 0018-9286. doi: 10.1109/TAC.2011.2161027.
- J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono. Optimal Rates for Zero-Order Convex Optimization: The Power of Two Function Evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, May 2015. ISSN 0018-9448. doi: 10.1109/TIT.2015.2409256.
- Educational Testing Service. TOEFL iBT Writing Sample Responses, 2005. URL http://toefl.uobabylon.edu.iq/papers/ibt_2014_12148630.pdf.
- Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. Properties of Multiwinner Voting Rules. *Social Choice and Welfare*, 48(3):599–632, 2017.
- Paul Embrechts, Filip Lindskog, and Alexander Mcneil. Chapter 8 - Modelling Dependence With Copulas and Applications to Risk Management. In Svetlozar T. Rachev, editor, *Handbook of Heavy Tailed Distributions in Finance*, volume 1 of *Handbooks in Finance*, pages 329–384. North-Holland, Amsterdam, 2003. URL <https://www.sciencedirect.com/science/article/pii/B9780444508966500108>. DOI: 10.1016/B978-044450896-6.50010-8.
- Piotr Faliszewski and Nimrod Talmon. A Framework for Approval-Based Budgeting Methods. *arXiv preprint arXiv:1809.04382*, 2018.
- Guiyun Feng, Guangwen Kong, and Zizhuo Wang. We Are on the Way: Analysis of On-Demand Ride-Hailing Systems. 2017. URL <https://dx.doi.org/10.2139/ssrn.2960991>.
- Apostolos Filippas, John J Horton, and Joseph M Golden. Reputation Inflation. Working Paper 25857, National Bureau of Economic Research, May 2019. URL <http://www.nber.org/papers/w25857>.
- Peter C Fishburn. Axioms for Approval Voting: Direct Proof. *Journal of Economic Theory*, 19(1): 180–185, 1978.
- Peter C. Fishburn and William V. Gehrlein. Borda’s Rule, Positional Voting, and Condorcet’s Simple Majority Principle. *Public Choice*, 28(1):79–88, December 1976. ISSN 0048-5829, 1573-7101. doi: 10.1007/BF01718459. URL <http://link.springer.com/10.1007/BF01718459>.
- James Fishkin, Nikhil Garg, Lodewijk Gelauff, Ashish Goel, Kamesh Munagala, Sukolsak Sakshuwong, Alice Siu, and Sravya Yandamuri. Deliberative Democracy with the Online Deliberation Platform. Technical report, October 2019.
- Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online Convex Optimization in the Bandit Setting: Gradient Descent Without a Gradient. In *Proceedings of the Sixteenth*

- Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 385–394, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics. ISBN 978-0-89871-585-9. URL <http://dl.acm.org/citation.cfm?id=1070432.1070486>.
- Andrey Fradkin, Elena Grewal, and David Holtz. The Determinants of Online Review Informativeness: Evidence From Field Experiments on AirBnb. 2018.
- Rupert Freeman, David M Pennock, Dominik Peters, and Jennifer Wortman Vaughan. Truthful Aggregation of Budget Proposals. *arXiv preprint arXiv:1905.00457*, 2019.
- Snehalkumar (Neil) S. Gaikwad, Mark Whiting, Karolina Ziulkoski, Alipta Ballav, et al. Boomerang: Rebounding the Consequences of Reputation Feedback on Crowdsourcing Platforms. pages 625–637. ACM Press, 2016.
- Nikhil Garg and Ramesh Johari. Designing Optimal Binary Rating Systems. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019a. URL <http://proceedings.mlr.press/v89/garg19a.html>.
- Nikhil Garg and Ramesh Johari. Designing Informative Rating Systems: Evidence From An Online Labor Market. 2019b. URL <https://arxiv.org/abs/1810.13028>.
- Nikhil Garg and Ramesh Johari. Designing Informative Rating Systems: Evidence From An Online Labor Market. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC '20)*, July 2020.
- Nikhil Garg and Hamid Nazerzadeh. Driver Surge Pricing. 2019. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3390346.
- Nikhil Garg and Hamid Nazerzadeh. Driver Surge Pricing. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC '20)*, July 2020.
- Nikhil Garg, Vijay Kamble, Ashish Goel, David Marn, and Kamesh Munagala. Collaborative Optimization for Collective Decision-making in Continuous Spaces. In *Proceedings of the 26th International Conference on World Wide Web*, pages 617–626, 2017.
- Nikhil Garg, Ashish Goel, and Benjamin Plaut. Markets for Public Decision-Making. 2018a.
- Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word Embeddings Quantify 100 Years of Gender and Ethnic Stereotypes. In *Proceedings of the National Academy of Sciences (PNAS)*, 2018b.
- Nikhil Garg, Lodewijk L Gelauff, Sukolsak Sakshuwong, and Ashish Goel. Who Is In Your Top Three? Optimizing Learning In Elections With Many Candidates. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 22–31, 2019a.

- Nikhil Garg, Vijay Kamble, Ashish Goel, David Marn, and Kamesh Munagala. Iterative Local Voting for Collective Decision-Making in Continuous Spaces. *Journal of Artificial Intelligence Research*, 64(1):315–355, January 2019b. ISSN 1076-9757. doi: 10.1613/jair.1.11358. URL <https://doi.org/10.1613/jair.1.11358>.
- Lodewijk Gelauff, Sukolsak Sakshuwong, Nikhil Garg, and Ashish Goel. Comparing Voting Methods for Budget Decisions on the ASSU Ballot. Technical report, 2018.
- Hollie Russon Gilman. Transformative Deliberations: Participatory Budgeting in the United States. *Journal of Public Deliberation*, 8(2), 2012. URL <http://search.proquest.com/openview/4c3b7fd3ae888f483cacad031a749131e/1?pq-origsite=gscholar>.
- Scott Gines. Tastes for True Talent: How Professional Baseball Scouts Define Talent and Decide Who Gets to Play. 2017.
- Amihai Glazer and Refael Hassin. The Economics of Cheating in the Taxi Market. *Transportation Research Part A*, 17(1):25–31, 1983.
- Peter Glynn and Sandeep Juneja. A Large Deviations Perspective on Ordinal Optimization. In *Simulation Conference, 2004. Proceedings of the 2004 Winter*, volume 1. IEEE, 2004.
- Ashish Goel, Anilesh K. Krishnaswamy, Sukolsak Sakshuwong, and Tanja Aitamurto. Knapsack Voting: Voting Mechanisms for Participatory Budgeting. 2016. URL http://web.stanford.edu/~anilesh/publications/knapsack_voting_full.pdf.
- Ashish Goel, Anilesh K. Krishnaswamy, and Kamesh Munagala. Metric Distortion of Social Choice Rules: Lower Bounds and Fairness Properties. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17*, pages 287–304, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4527-9. doi: 10.1145/3033274.3085138. URL <http://doi.acm.org/10.1145/3033274.3085138>.
- Harish Guda and Upender Subramanian. Your Uber Is Arriving: Managing On-Demand Workers Through Surge Pricing, Forecast Communication, and Worker Incentives. *Management Science*, 65(5):1995–2014, 2019. doi: 10.1287/mnsc.2018.3050. URL <https://doi.org/10.1287/mnsc.2018.3050>.
- John Guiver and Edward Snelson. Bayesian Inference for Plackett-Luce Ranking Models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 377–384. ACM, 2009.
- Jonathan V. Hall, Cory Kendrick, and Chris Nosko. The Effects of Uber’s Surge Pricing: A Case Study. 2015. URL <https://eng.uber.com/research/the-effects-of-ubers-surge-pricing-a-case-study/>.

- Jonathan V. Hall, John J. Horton, and Daniel T. Knoepfle. Labor Market Equilibration: Evidence From Uber. 2017. URL <https://eng.uber.com/research/labor-market-equilibration-evidence-from-uber/>.
- Takeshi Hamamura, Steven J. Heine, and Delroy L. Paulhus. Cultural Differences in Response Styles: The Role of Dialectical Thinking. *Personality and Individual Differences*, 44(4):932–942, March 2008. ISSN 01918869. doi: 10.1016/j.paid.2007.10.034. URL <http://linkinghub.elsevier.com/retrieve/pii/S0191886907003820>.
- Avinatan Hassidim, Assaf Romm, and Ran I Shorrer. “Strategic” Behavior in a Strategy-Proof Environment. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 763–764, 2016.
- Fred Hicks, Lee Valentine, John Morrow, and Ian McDonald. Choosing Natural Adjective Ladders, 2000. URL http://www.mcdonald.me.uk/storytelling/lichert_article.htm.
- Zoë Hitzig. Bridging the ‘Normative Gap’: Mechanism Design and Social Justice. *Available at SSRN 3242882*, 2018.
- Bryan Hooi, Neil Shah, Alex Beutel, Stephan Günnemann, Leman Akoglu, Mohit Kumar, Disha Makhija, and Christos Faloutsos. Birdnest: Bayesian Inference for Ratings-Fraud Detection. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 495–503. SIAM, 2016.
- Ming Hu and Yun Zhou. Dynamic Type Matching. *Rotman School of Management Working Paper*, (2592622), 2018.
- Nan Hu, Jie Zhang, and Paul A Pavlou. Overcoming the J-Shaped Distribution of Product Reviews. *Communications of the ACM*, 52(10):144–147, 2009.
- Nan Hu, Ling Liu, and Vallabh Sambamurthy. Fraud Detection in Online Consumer Reviews. *Decision Support Systems*, 50(3):614–626, 2011.
- Aanund Hylland and Richard Zeckhauser. *A Mechanism for Selecting Public Goods When Preferences Must Be Elicited*. 1980.
- Bar Ifrach, Costis Maglaras, Marco Scarsini, and Anna Zseleva. Bayesian Social Learning From Consumer Reviews. SSRN Scholarly Paper ID 2293158, Social Science Research Network, Rochester, NY, December 2017.
- Nicole Immorlica. The Design of Everyday Markets. In *The Future of Economic Design*, pages 517–522. Springer, 2019.

- Nicole Immorlica, Brendan Lucier, and Brian Rogers. Emergence of Cooperation in Anonymous Social Networks Through Social Capital. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, 2010.
- Kevin G. Jamieson, Robert Nowak, and Ben Recht. Query Complexity of Derivative-Free Optimization. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2012. URL <http://papers.nips.cc/paper/4509-query-complexity-of-derivative-free-optimization>.
- Libin Jiang and Jean Walrand. Scheduling and Congestion Control for Wireless and Processing Networks. *Synthesis Lectures on Communication Networks*, 3(1):1–156, January 2010. ISSN 1935-4185, 1935-4193. doi: 10.2200/S00270ED1V01Y201008CNT006. URL <http://www.morganclaypool.com/doi/abs/10.2200/S00270ED1V01Y201008CNT006>.
- Ramesh Johari, Vijay Kamble, and Yash Kanoria. Matching While Learning. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, EC '17, pages 119–119, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4527-9. doi: 10.1145/3033274.3084095. event-place: Cambridge, Massachusetts, USA.
- Valen E Johnson. *Grade Inflation: A Crisis in College Education*. Springer Science & Business Media, 2006.
- Vijay Kamble. Revenue Management on an On-Demand Service Platform. *Operations Research Letters*, 47(5):377–385, 2019.
- Yash Kanoria and Pengyu Qian. Near Optimal Control of a Ride-Hailing Platform via Mirror Backpressure. March 2019. URL <http://arxiv.org/abs/1903.02764>.
- Sumeet Katariya, Branislav Kveton, Csaba Szepesvári, and Zheng Wen. DCM Bandits: Learning to Rank with Multiple Clicks. *arXiv:1602.03146 [cs, stat]*, February 2016. URL <http://arxiv.org/abs/1602.03146>. arXiv: 1602.03146.
- John G Kemeny. Mathematics Without Numbers. *Daedalus*, 88(4):577–591, 1959.
- Alan J Klockars and Midori Yamagishi. The Influence of Labels and Positions in Rating Scales. *Journal of Educational Measurement*, 25(2):85–96, 1988.
- Noi Sian Koh, Nan Hu, and Eric K Clemons. Do Online Reviews Reflect a Product’s True Perceived Quality? An Investigation of Online Movie Reviews Across Cultures. *Electronic Commerce Research and Applications*, 9(5):374–385, 2010.
- Anurag Komanduri, Zeina Wafa, Kimon Prousaloglou, and Simon Jacobs. Assessing the Impact of App-Based Ride Share Systems in an Urban Context: Findings From Austin. *Transportation Research Record*, 2672(7):34–46, 2018.

- Nikita Korolko, Dawn Woodard, Chiwei Yan, and Helin Zhu. Dynamic Pricing and Matching in Ride-Hailing Platforms. *SSRN Electronic Journal*, page 40, 2018. ISSN 1556-5068. doi: 10.2139/ssrn.3258234. URL <https://www.ssrn.com/abstract=3258234>.
- Jon A Krosnick. Survey Research. *Annual Review of Psychology*, 50(1):537–567, 1999.
- Laura W Lackey and W JACK Lackey. Grade Inflation: Potential Causes and Solutions. *International Journal of Engineering Education*, 22(1):130, 2006.
- Martin Lackner and Piotr Skowron. Consistent Approval-Based Multi-Winner Rules. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, pages 47–48, New York, NY, USA, 2018a. ACM. ISBN 978-1-4503-5829-3. doi: 10.1145/3219166.3219170. URL <http://doi.acm.org/10.1145/3219166.3219170>.
- Martin Lackner and Piotr Skowron. A Quantitative Analysis of Multi-Winner Rules. *arXiv preprint arXiv:1801.01527*, 2018b.
- Steven P. Lalley and E. Glen Weyl. Quadratic Voting. SSRN Scholarly Paper ID 2003531, Social Science Research Network, Rochester, NY, December 2015. URL <http://papers.ssrn.com/abstract=2003531>.
- Richard N Landers and Tara S Behrend. An Inconvenient Truth: Arbitrary Distinctions Between Organizational, Mechanical Turk, and Other Convenience Samples. *Industrial and Organizational Psychology*, 8(2):142–164, 2015.
- David Timothy Lee, Ashish Goel, Tanja Aitamurto, and Helene Landemore. Crowdsourcing for Participatory Democracies: Efficient Elicitation of Social Choice Functions. In *Second AAAI Conference on Human Computation and Crowdsourcing*, September 2014. URL <https://www.aaai.org/ocs/index.php/HCOMP/HCOMP14/paper/view/8952>.
- Yanzhe Lei and Stefanus Jasin. Real-Time Dynamic Pricing for Revenue Management With Reusable Resources and Deterministic Service Time Requirements. 2016.
- Lingfang Li and Erte Xiao. Money Talks: Rebate Mechanisms in Reputation System Design. *Management Science*, 60(8):2054–2072, 2014.
- Alice Lu, Peter I. Frazier, and Oren Kislev. Surge Pricing Moves Uber’s Driver-Partners. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, pages 3–3, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5829-3. doi: 10.1145/3219166.3219192.
- Tyler Lu and Craig Boutilier. Learning Mallows Models With Pairwise Preferences. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 145–152, USA, 2011. Omnipress. ISBN 978-1-4503-0619-5. URL <http://dl.acm.org/citation.cfm?id=3104482.3104501>.

- Michael Luca and Max Bazerman. *The Power of Experiments: Decision Making in a Data-Driven World*. MIT Press, 2020.
- Michael Luca and Georgios Zervas. Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud. *Management Science*, 62(12):3412–3427, 2016.
- Hongyao Ma, Fei Fang, and David C. Parkes. Spatio-Temporal Pricing for Ridesharing Platforms. January 2018. URL <http://arxiv.org/abs/1801.04015>.
- Francis Maes, Louis Wehenkel, and Damien Ernst. Automatic Discovery of Ranking Formulas for Playing With Multi-Armed Bandits. In *Recent Advances in Reinforcement Learning*, Lecture Notes in Computer Science, pages 5–17. Springer, Berlin, Heidelberg, September 2011. ISBN 978-3-642-29945-2 978-3-642-29946-9. doi: 10.1007/978-3-642-29946-9_5. URL https://link.springer.com/chapter/10.1007/978-3-642-29946-9_5.
- Colin L Mallows. Non-Null Ranking Models. I. *Biometrika*, 44(1/2):114–130, 1957.
- Marie Jean Antoine marquis de Condorcet. *Essai Sur l'Application De l'Analyse a La Probabilite Des Decisions: Rendues a La Pluralite De Voix*. De l'Imprimerie royale, 1785.
- Nicholas Mattei and Toby Walsh. PrefLib: A Library of Preference Data <http://preflib.org>. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT 2013)*, Lecture Notes in Artificial Intelligence. Springer, 2013.
- Lucas Maystre and Matthias Grossglauser. Fast and Accurate Inference of Plackett–Luce Models. In *Advances in Neural Information Processing Systems*, pages 172–180. 2015.
- Nina Mazar, On Amir, and Dan Ariely. The Dishonesty of Honest People: A Theory of Self-Concept Maintenance. *Journal of Marketing Research*, 45(6):633–644, 2008.
- Patrice McDermott. Building Open Government. *Government Information Quarterly*, 27(4): 401–413, October 2010. ISSN 0740-624X. doi: 10.1016/j.giq.2010.07.002. URL <http://www.sciencedirect.com/science/article/pii/S0740624X10000663>.
- Reshef Meir, Maria Polukarov, Jeffrey S. Rosenschein, and Nicholas R. Jennings. Convergence to Equilibria in Plurality Voting. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, volume 10 of *AAAI'10*, page 823–828. AAAI Press, 2010.
- H. Moulin. On Strategy-Proofness and Single Peakedness. *Public Choice*, 35(4):437–455, 1980. ISSN 0048-5829. URL <http://www.jstor.org/stable/30023824>.
- Roger B. Nelsen. *An Introduction to Copulas*. Springer Science & Business Media, June 2007. ISBN 978-0-387-28678-5.

- Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust Stochastic Approximation Approach to Stochastic Programming. *SIAM Journal on Optimization*, 19(4): 1574–1609, 2009.
- Chris Nosko and Steven Tadelis. The Limits of Reputation in Platform Markets: An Empirical Analysis and Field Experiment. Working Paper, National Bureau of Economic Research, 2015.
- Jeffery O’Neill. Open STV, 2013. URL www.OpenSTV.org.
- Erhun Özkan. Joint Pricing and Matching in Ridesharing Systems. 2018.
- Erhun Özkan and Amy Ward. Dynamic Matching for Real-Time Ridesharing. *SSRN Electronic Journal*, 2016. ISSN 1556-5068. doi: 10.2139/ssrn.2844451. URL <http://www.ssrn.com/abstract=2844451>.
- Yiangos Papanastasiou, Kostas Bimpikis, and Nicos Savva. Crowdsourcing Exploration. *Management Science*, 64(4):1727–1746, April 2017. ISSN 0025-1909. doi: 10.1287/mnsc.2016.2697.
- A. Parasuraman, Dhruv Grewal, and R. Krishnan. *Marketing Research*. Cengage Learning, 2006. ISBN 978-0-618-66063-6.
- Sergey V Popov, Anna Popova, and Michel Regenwetter. Consensus in Organizations: Hunting for the Social Choice Conundrum in APA Elections. *Decision*, 1(2):123, 2014.
- Canice Prendergast. The Allocation of Food to Food Banks. *EAI Endorsed Trans. Serious Games*, 3(10):e4, 2016.
- Ariel D. Procaccia and Jeffrey S. Rosenschein. The Distortion of Cardinal Preferences in Voting. In *International Workshop on Cooperative Information Agents*, pages 317–331. Springer, 2006. URL http://link.springer.com/chapter/10.1007/11839354_23.
- Ariel D Procaccia and Nisarg Shah. Is Approval Voting Optimal Given Approval Votes? In *Advances in Neural Information Processing Systems 28*, pages 1801–1809, 2015. URL <http://papers.nips.cc/paper/5884-is-approval-voting-optimal-given-approval-votes.pdf>.
- Ariel D. Procaccia and Moshe Tennenholtz. Approximate Mechanism Design Without Money. In *Proceedings of the 10th ACM Conference on Electronic Commerce, EC ’09*, pages 177–186, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-458-4. doi: 10.1145/1566374.1566401. URL <http://doi.acm.org/10.1145/1566374.1566401>.
- Public Agenda. Public Spending By The People: Participatory Budgeting in the United States and Canada in 2014–15. Technical report, The Yankelovich Center for Public Judgment, May 2016. URL https://www.publicagenda.org/files/PublicSpendingByThePeople_PublicAgenda_2016.pdf.

- David Quarfoot, Douglas von Kohorn, Kevin Slavin, Rory Sutherland, David Goldstein, and Ellen Konar. Quadratic Voting in the Wild: Real People, Real Votes. *Public Choice*, 172(1):283–303, July 2017. ISSN 1573-7101. doi: 10.1007/s11127-017-0416-1. URL <https://doi.org/10.1007/s11127-017-0416-1>.
- Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning Diverse Rankings With Multi-Armed Bandits. In *Proceedings of the 25th International Conference on Machine Learning*, pages 784–791. ACM, 2008.
- Shiva Rajaraman. Five Stars Dominate Ratings, September 2009. URL <https://youtube.googleblog.com/2009/09/five-stars-dominate-ratings.html>.
- Thomas C. Ratliff. Some Startling Inconsistencies When Electing Committees. *Social Choice and Welfare*, 21(3):433–454, December 2003. ISSN 1432-217X. doi: 10.1007/s00355-003-0209-y. URL <https://doi.org/10.1007/s00355-003-0209-y>.
- Michel Regenwetter, Aeri Kim, Arthur Kantor, and Moon-Ho R Ho. The Unexpected Empirical Consensus Among Consensus Methods. *Psychological Science*, 18(7):629–635, 2007.
- Michel Regenwetter, Bernard Grofman, Anna Popova, William Messner, Clinton P Davis-Stober, and Daniel R Cavagnaro. Behavioural Social Choice: A Status Report. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1518):833–843, 2008.
- Ben Reiter. *Astrobball: The New Way to Win It All*. Crown Archetype, 2018.
- RideAustin. Driver Rates, 2019. URL <http://www.rideaustin.com/drivers/rates>.
- Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407, September 1951. ISSN 0003-4851, 2168-8990. doi: 10.1214/aoms/1177729586. URL <http://projecteuclid.org/euclid.aoms/1177729586>.
- Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. Kidney Exchange. *The Quarterly Journal of Economics*, 119(2):457–488, 05 2004. ISSN 0033-5533. doi: 10.1162/0033553041382157. URL <https://doi.org/10.1162/0033553041382157>.
- Matthew Salganik. *Bit by Bit: Social Research in the Digital Age*. Princeton University Press, 2019.
- Anwar Shah, editor. *Participatory Budgeting*. Public Sector Governance and Accountability Series. World Bank, Washington, D.C, 2007. ISBN 978-0-8213-6923-4 978-0-8213-6924-1. OCLC: ocm71947871.
- Adnan Shaout and Mohamed K Yousif. Performance Evaluation—Methods and Techniques Survey. *International Journal of Computer and Information Technology*, 3(5):966–979, 2014.

- Naum Z. Shor. *Nondifferentiable Optimization and Polynomial Problems*, volume 24 of *Nonconvex Optimization and Its Applications*. Springer US, Boston, MA, 1998. ISBN 978-1-4419-4792-5 978-1-4757-6015-6. doi: 10.1007/978-1-4757-6015-6. URL <http://link.springer.com/10.1007/978-1-4757-6015-6>.
- Yves Sintomer, Carsten Herzberg, and Anja Rocke. From Porto Alegre to Europe: Potentials and Limitations of Participatory Budgeting. *International Journal of Urban and Regional Research*, 32(1):164–178, 2008. URL http://www.cpa.zju.edu.cn/participatory_budgeting_conference/english_articles/paper2.pdf.
- Mike Staring. Two Paradoxes of Committee Elections. *Mathematics Magazine*, 59(3):158–159, 1986.
- Volker Strassen. The Existence of Probability Measures With Given Marginals. *The Annals of Mathematical Statistics*, 36(2):423–439, April 1965. ISSN 0003-4851, 2168-8990. doi: 10.1214/aoms/1177700153. URL <https://projecteuclid.org/euclid.aoms/1177700153>.
- Steven Tadelis. Reputation and Feedback Systems in Online Platform Markets. *Annual Review of Economics*, 8:321–340, 2016.
- Agostino Tarsitano. Comparing the Effectiveness of Rank Correlation Statistics. *Working Papers, Universita della Calabria, Dipartimento di Economia e Statistica*, pages 1–25, 2009.
- Maria Tataru and Vincent Merlin. On the Relationship of the Condorcet Winner and Positional Voting Rules. *Mathematical Social Sciences*, 34(1):81–90, August 1997. ISSN 0165-4896. doi: 10.1016/S0165-4896(97)00005-X. URL <http://www.sciencedirect.com/science/article/pii/S016548969700005X>.
- T. Nicolaus Tideman and Florenz Plassmann. Efficient Collective Decision-Making, Marginal Cost Pricing, And Quadratic Voting. SSRN Scholarly Paper ID 2836610, Social Science Research Network, Rochester, NY, August 2016. URL <https://papers.ssrn.com/abstract=2836610>.
- Uber. Community Guidelines, 2019a. URL <https://www.uber.com/legal/community-guidelines/us-en/>.
- Uber. Dependable Earnings, 2019b. URL <https://www.uber.com/drive/resources/dependable-earnings/>.
- Uber. New Driver Surge, 2019c. URL <https://www.uber.com/blog/your-questions-about-the-new-surge-answered/>.
- Uber. How Are Fares Calculated, 2019d. URL <https://help.uber.com/riders/article/how-are-fares-calculated?nodeId=d2d43bbc-f4bb-4882-b8bb-4bd8acf03a9d>.
- Uber. Service Fee, 2019e. URL <https://marketplace.uber.com/pricing/service-fee>.

- Abraham Wald. *Sequential Analysis*. Courier Corporation, 1973.
- Hao-Chuan Wang, Tau-Heng Yeo, Syavash Nobarany, and Gary Hsieh. Problem With Cross-Cultural Comparison of User-Generated Ratings on Mechanical Turk. In *Proceedings of the Third International Symposium of Chinese CHI*, pages 9–12. ACM, 2015.
- James Wiseman. Approval Voting in Subset Elections. *Economic Theory*, 15(2):477–483, 2000.
- Pu Yang, Krishnamurthy Iyer, and Peter Frazier. Mean Field Equilibria for Resource Competition in Spatial Settings. *Stochastic Systems*, 8(4):307–334, 2018.
- H Peyton Young. Social Choice Scoring Functions. *SIAM Journal on Applied Mathematics*, 28(4):824–838, 1975.
- H Peyton Young. Condorcet’s Theory of Voting. *American Political Science Review*, 82(4):1231–1244, 1988.
- Yisong Yue and Thorsten Joachims. Interactively Optimizing Information Retrieval Systems as a Dueling Bandits Problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1201–1208. ACM, 2009.
- Georgios Zervas, Davide Proserpio, and John Byers. A First Look at Online Reputation on Airbnb, Where Every Stay Is Above Average. Technical Report ID 2554500, Social Science Research Network, January 2015.
- Lingyu Zhang, Tao Hu, Yue Min, Guobin Wu, Junying Zhang, Pengcheng Feng, Pinghua Gong, and Jieping Ye. A Taxi Order Dispatch Model Based on Combinatorial Optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2151–2159. ACM, 2017.
- Yu Zhang, Jing Bian, and Weixiang Zhu. Trust Fraud: A Crucial Challenge for China’s E-Commerce Market. *Electronic Commerce Research and Applications*, 12(5):299–308, 2013.
- Zhibing Zhao, Peter Piech, and Lirong Xia. Learning Mixtures of Plackett-Luce Models. In *International Conference on Machine Learning*, pages 2906–2914, 2016. URL <http://arxiv.org/abs/1603.07323>.