Decision Making With Heterogeneous Agents: Elicitation, Aggregation, and Causal Effects

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters

<table>
<thead>
<tr>
<th>Citation</th>
<th>Mandal, Debmalya. 2019. Decision Making With Heterogeneous Agents: Elicitation, Aggregation, and Causal Effects. Doctoral dissertation, Harvard University, Graduate School of Arts &amp; Sciences.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citable link</td>
<td><a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:42029786">http://nrs.harvard.edu/urn-3:HUL.InstRepos:42029786</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>This article was downloaded from Harvard University’s DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA">http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA</a></td>
</tr>
</tbody>
</table>
Decision Making with Heterogeneous Agents: Elicitation, Aggregation, and Causal Effects

A dissertation presented
by
Debmalya Mandal
to
John A. Paulson School of Engineering and Applied Sciences
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Computer Science
Harvard University
Cambridge, Massachusetts
May 2019
The methods of artificial intelligence and statistical machine learning are finding tremendous success in various fields, with applications ranging from cancer screening to machine translation. However, continued improvement in predictive accuracy is not sufficient to guarantee that these systems can be used reliably across a variety of societal contexts. There are three main challenges in the development of robust AI systems – (a) **Elicitation** to obtain high-quality feedback on training instances, (b) **Aggregation** to understand and summarize the trade-offs arising from decisions across society, and (c) **Causal Inference** to estimate the impact of any system before deploying. My thesis makes progress on specific problems within each of these three challenges. A unifying theme is that of the need to be careful in handling the heterogeneity of agents, which is prevalent in many domains, be it the reviewers on a platform like Yelp or the consumers of a recommendation service like Amazon.

First, I propose new peer prediction mechanisms to elicit feedback on instances where the responses cannot be verified. I consider both the heterogeneous agents and tasks setting and show that the proposed mechanisms have better incentive guarantees both in theory and through empirical evaluation. In regard to aggregation, I consider theoretical aspects of voting rules, motivated by anticipated use of AI systems in the context of societal decision making. I provide a unified view of voting by considering elicitation and aggregation together, and provide a sharp characterization of the performance of such rules. Finally, I develop a tensor decomposition approach to estimating the impact of a policy that applies treatments over a sequence of rounds. I show that our estimator is consistent, propose an algorithm to efficiently solve the estimation problem, and through simulation show that it has better performance than
existing methods for causal inference under time-varying treatments.
## Contents

Abstract ................................................................. iii  
Acknowledgments .......................................................... xi  

**Introduction**  

1 Peer Prediction with Heterogeneous Users  
1.1 Introduction ......................................................... 7  
1.1.1 Related Work ..................................................... 11  
1.2 Model ................................................................. 13  
1.2.1 Multi-Task Peer Prediction ....................................... 15  
1.2.2 Task Assignments ............................................... 17  
1.2.3 Expected Payments .............................................. 18  
1.2.4 Informed Truthfulness .......................................... 18  
1.2.5 Learning and Agent Clustering ................................. 19  
1.3 Correlated Agreement for Heterogeneous Agents ............... 21  
1.3.1 Analysis of CAHU .............................................. 22  
1.4 Learning the Agent Signal Types .................................. 29  
1.4.1 Clustering ....................................................... 31  
1.4.2 Learning the Cluster Pairwise $\Delta$ Matrices .............. 35  
1.5 Clustering Experiments ............................................ 44  
1.6 Conclusion ......................................................... 47  

2 Peer Prediction for Heterogeneous Tasks  
2.1 Introduction ......................................................... 49  
2.1.1 Related Work ..................................................... 50  
2.2 Heterogeneous, Multi-Task Peer Prediction ..................... 51  
2.2.1 Delta Matrices ................................................... 53  
2.3 The Correlated-Agreement Heterogeneous (CAH) Mechanism .... 54  
2.3.1 Informed Truthfulness .......................................... 56  
2.3.2 Strong Truthfulness ............................................ 57  
2.3.3 Combining CAH with Estimation ............................... 58  
2.3.4 Cross Correlated Agreement .................................... 59
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.5 Asymmetric Strategy</td>
<td>60</td>
</tr>
<tr>
<td>2.4 Experimental Results</td>
<td>61</td>
</tr>
<tr>
<td>2.4.1 Unilateral Incentives for Truthful Reports</td>
<td>63</td>
</tr>
<tr>
<td>2.4.2 Benefit from Coordinated Misreports</td>
<td>64</td>
</tr>
<tr>
<td>2.4.3 Subjective vs Factual Tasks</td>
<td>66</td>
</tr>
<tr>
<td>2.5 Conclusion</td>
<td>67</td>
</tr>
<tr>
<td>3 Correlated Voting</td>
<td>68</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>68</td>
</tr>
<tr>
<td>3.2 Setup</td>
<td>70</td>
</tr>
<tr>
<td>3.2.1 Positively Correlated Preferences</td>
<td>71</td>
</tr>
<tr>
<td>3.3 Scoring Rules</td>
<td>72</td>
</tr>
<tr>
<td>3.3.1 EPIC Convergence for KT-correlation</td>
<td>73</td>
</tr>
<tr>
<td>3.3.2 EPIC Convergence for TS-correlation</td>
<td>76</td>
</tr>
<tr>
<td>3.4 Rank-Order Models</td>
<td>78</td>
</tr>
<tr>
<td>3.5 Conclusion</td>
<td>82</td>
</tr>
<tr>
<td>4 Limits and Possibilities of Implicit Utilitarian Voting</td>
<td>83</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>83</td>
</tr>
<tr>
<td>4.1.1 Our Results</td>
<td>85</td>
</tr>
<tr>
<td>4.1.2 Related Work</td>
<td>86</td>
</tr>
<tr>
<td>4.2 Model</td>
<td>88</td>
</tr>
<tr>
<td>4.3 Upper Bounds</td>
<td>91</td>
</tr>
<tr>
<td>4.3.1 Deterministic Elicitation, Deterministic Aggregation</td>
<td>91</td>
</tr>
<tr>
<td>4.3.2 Randomized Elicitation, Randomized Aggregation</td>
<td>96</td>
</tr>
<tr>
<td>4.4 Direct Lower Bounds For Deterministic Elicitation</td>
<td>99</td>
</tr>
<tr>
<td>4.5 Lower Bounds Through Multi-Party Communication Complexity</td>
<td>104</td>
</tr>
<tr>
<td>4.5.1 Setup</td>
<td>104</td>
</tr>
<tr>
<td>4.5.2 Multi-Party Fixed-Size Set-Disjointness</td>
<td>105</td>
</tr>
<tr>
<td>4.5.3 Lower Bounds on the Communication Complexity of Voting Rules</td>
<td>105</td>
</tr>
<tr>
<td>4.6 Discussion</td>
<td>109</td>
</tr>
<tr>
<td>5 Causal Inference with Time-Varying Treatments</td>
<td>112</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>112</td>
</tr>
<tr>
<td>5.1.1 Contributions</td>
<td>114</td>
</tr>
<tr>
<td>5.1.2 Related Work</td>
<td>114</td>
</tr>
<tr>
<td>5.2 Model</td>
<td>117</td>
</tr>
<tr>
<td>5.2.1 Outcome Model</td>
<td>118</td>
</tr>
<tr>
<td>5.2.2 Sequentially Randomized Experiment</td>
<td>119</td>
</tr>
<tr>
<td>5.2.3 Quantities to Estimate</td>
<td>120</td>
</tr>
</tbody>
</table>
List of Tables

1.1 Sample Complexity for the CAHU Mechanism .......................................................... 30
5.1 Marginal Structural Model vs Weighted Tensor Completion ........................................ 133
List of Figures

1.1 Fixed Task Assignment .................................................. 30
1.2 Uniform Task Assignment .............................................. 30
1.3 To Cluster or Not to Cluster (Algorithm 1.2) ....................... 32
1.4 The Incentive Error as a Fraction of the Maximum Payoff of an Agent 48
1.5 The Incentive Error of CAHU as a Fraction of the Expected Payoff of an Agent 48

2.1 Expected Benefit from Truthful Behavior vs. Some Other Strategy .... 64
2.2 Expected Score for Following Four Types of Strategies When Only a Fraction of the Population is Truthful ............... 65
2.3 Cumulative Distribution on Expected Payments at Truthful Reporting .... 66

3.1 The Probability of Manipulation Under Different Belief Models .......... 76

4.1 Existing results (in blue) and our results (in red) on the communication-distortion tradeoff. ........................................... 86

5.1 A directed acyclic graph describing the model for agent i ............... 117
5.2 Average Absolute Error in Estimating ATET Through Weighted Tensor Approximation (Rank) ........................................... 131
5.3 Average Absolute Error in Estimating ATET Through Weighted Tensor Approximation (History) ......................................... 132

B.1 More Histograms to Show the Expected Benefit from Truthful Behavior vs. Some Other Strategy ........................................... 169
List of Algorithms

1.1 Mechanism CAHU ................................................................. 23
1.2 Clustering ................................................................. 32
1.3 Learning-Δ-No-Assumption ............................................. 35
1.4 Estimating Aggregate Confusion Matrix ................................ 40
2.1 CAH mechanism ............................................................ 55
2.2 CAHR mechanism ............................................................ 59
4.1 PrefThreshold_{t,ℓ}, where ℓ ∈ [m] and ℓ ∈ N. ...................... 92
4.2 RandSubset(f,s), where f is a voting rule and s ∈ [m] .......... 97
5.1 Weighted Low Rank Tensor Approximation ....................... 128
5.2 Weighted Low Rank Tensor Approximation ....................... 130
5.3 Weighted Low Rank Tensor Approximation with Covariates ... 137
Acknowledgments

I would not have been able to write this thesis without the constant support and guidance of my amazing advisor, David C. Parkes. It is quite extraordinary how much David has influence my research interests while still allowing me to explore new research areas. You are an amazing person and I will be forever grateful for your support.

I am indebted to Nisarg Shah, who has been an amazing friend and collaborator. I hope to continue our collaborations in future. I would also like to thank Jacob LaRiviere for hosting me for a summer at Microsoft Research, Redmond. My exploration into causal inference began there while working on a project with Jacob. I would like to thank José Zubizarreta for many insightful discussions about causal inference and providing many valuable comments on a draft of this thesis. I must thank Yiling Chen for many encouraging discussions, and waking up at 5:30 AM to attend my defense! Finally, I would like to thank the other members of my thesis committee, Prof. Francesca Dominici, Prof. Kosuke Imai, and Prof. Madhu Sudan for many interesting discussions and valuable comments.

I had the opportunity to work with some amazing postdoctoral fellows at Harvard, particularly, Nisarg Shah, Goran Radanovic, Berk Ustun, and Yang Liu. I extremely enjoyed our discussions and hope to collaborate more in future. I am grateful for having a vibrant community of graduate students at EconCS including Hongyao, Eric, Thibaut, Paul, Chara, Juntao, Lily, Zhe, Ani, and Sai. I cannot imagine working in Maxwell-Dworkin for five years without you guys. I must also thank all the members of DENS groups including Ambarish, Anna, Ian, Yige, and Noémie.

I would also like to thank Prof. Yadati Narahari form Indian Institute of Science, who taught me Game Theory and encouraged me to pursue a research career. I am grateful to my friends back in India including Sambaran, Satyanath, Rohith, Swapnil, and Jay who have been encouraging me throughout my journey at Harvard. I am immensely thankful to my friends in the greater Boston area including Rajat, Dibayan, Puneet, Ashish, Shweta, Hari, Debjyoti, Ambarish, Tarangini, Ananth, Shubhayu, Manoj, and Sherin. They have made my stay at Boston really wonderful. I am also grateful to my friends in other parts of the United
States, including Deep, Ratul, Sourav, Arupratan, Reetuparna. They have been too gracious to host me whenever I wanted a respite from graduate school. I am indebted to my family, particularly my cousins Shukla, Shampa, Avijit, who made sure to check on me regularly and were always there to talk to. Finally, I would like to thank my parents, brother, and sister-in-law for believing me that I could finish a PhD and supporting me all the way.
Citations to Previously Published Work

1. A significant portion of the work presented in Chapter 1 is based on the following publication:


2. The work presented in Chapter 2 was done with our collaborator in Google Local Guides, who provided the dataset for analysis. This work is based on the following paper:


3. The work presented in Chapter 3 is based on the following publication:


4. The work presented in Chapter 4 is based on the following paper, which is presently under submission:


5. A preliminary version of the work presented in Chapter 5 appeared as:

Introduction

The methods of artificial intelligence and statistical machine learning are finding tremendous success in various fields, with applications ranging from cancer screening [Wu+19] to machine translation [Wu+16]. The success of such AI systems can be attributed to the rapid increase in predictive performance of neural networks because of various architectural innovations [He+16; Hua+17] and the availability of large datasets [Den+09]. The success of AI in tasks like image recognition and speech translation has naturally also led to its adoption in various, high-stakes societal problems [Hor17]. Examples include predictive policing [Lap18], criminal justice [Kle+17], and healthcare [Top19].

However, continued improvement in predictive accuracy is not sufficient to guarantee that these systems can be used reliably across a variety of societal contexts. Rather, there remain three major challenges in the development of robust AI systems:

1. Modern deep learning systems are data hungry [Mar18] and need millions of accurately labelled training instances. For example, AlexNet [KSH12], one of the most influential deep-net architectures from computer vision, required almost a million training examples from the Imagenet dataset. Acquiring such datasets is often costly and often overlooked in the design of AI systems. However, as such systems are adopted across a wide range of applications, we will need to consider data collection as an integral part of design and construct mechanisms to obtain high quality feedback from human labellers or annotators.

2. As AI systems are increasingly being used for making high-stakes decisions in society, we need to ensure that such systems reflect appropriate values and ethics. We cannot
simply use past data to train such systems as this will reflect and amplify past biases [CBN17]. Moreover, in many situations, it is not a priori clear what are the right trade-offs [Awa+18]. Therefore, it seems one fruitful direction will be to design rules that aggregate people’s opinions about the various trade-offs AI systems face. Social choice theory, beginning with the work of Arrow [Arr12], has studied the aggregation of preferences over alternatives. However, in the context of using AI for high-stake decision making, the trade-offs are often complex and hard to specify. Therefore, we need to redesign classical aggregation rules to not only aggregate opinions across a large number of users, but also to lessen the cognitive burden of the users.

3. There is a need to explicitly model environments in terms of causal models, particularly when we want to generalize results to unseen domains [Wel19]. As an example, consider the design of self-driving cars. It is quite expensive to get training examples [Mad17] and many rare examples like tricky intersections, complicated traffic rules and accidents are rarely going to come up in the training instances. Recently, Judea Pearl [PM18] argued that in order to build human-level intelligent machines, AI systems must build a causal model of the environment. This principle also applies if we want to generalize the training of self-driving cars to unseen domains. A causal model of the world would help the self-driving car to plan in an unseen world and make the design more robust to changes in domain. As these systems are adopted in practice, the implications of not having a model is even more severe. For example, once the self-driving cars are deployed, the system must be able to predict the responses of passengers and other people on the road in response to various actions. Since the number of such interactions is limitless, the designers need to encode such a model explicitly during the design.

My thesis makes progress on specific problems within each of these three domains. The main theme underlying my research is to be careful in handling the heterogeneity of agents, which is important because AI systems often need to consider a diverse group of users. For example, consider an online platform like Yelp, which elicits reviews from users about various places and restaurants. The users visiting such a broad platform vary in taste, location etc and
the platform must consider such heterogeneity when designing an incentive scheme for truthful reviews. Second, consider the problem of aggregating preferences on an issue like regulation of autonomous vehicles. (13) observed that there are significant variations in preferences along age, education, gender, income, and political and religious views. Finally, suppose we want to improve an existing personalized recommendation service on a platform like Amazon. With access to a very large amount of data, we can do individual treatment effect estimation and try to give better recommendations to each user, rather than simply estimating the average impact of the policy change. Such an analysis, in turn, improves the overall performance of the whole system.

The rest of the thesis can be organized into three groups – (a) Elicitation (chapters 1 and 2), (b) Aggregation (chapters 3 and 4), and (c) Causal Inference (chapter 5).

(a) **Elicitation**: Crowdsourcing has been the main source of obtaining annotated datasets, whether we want to detect whether an image contains a supernova or not on Galaxy Zoo or translate a text from English to Urdu on MTurk. The main challenge in crowdsourcing is to incentivize humans to provide accurate information in the absence of ground truth. Peer prediction studies the elicitation of information that cannot be verified, and offers a solution to this challenge. However, current peer prediction mechanisms ignore the fact that users who provide feedback may be quite different in the way they think about the world. For example, users reviewing a particular business may belong to different communities and have significantly different opinions and tastes. In Chapter 1, I present the first peer prediction mechanism for heterogeneous users. The proposed mechanism is informed truthful, meaning it provides strict incentives for being truthful, versus any other strategy, including not investing effort or adopting a random strategy. The mechanism works by first identifying the cluster of a user, and then learning the pairwise correlations between the cluster and other clusters. Moreover, if each question is associated with a true underlying answer, then the mechanism reduces to learning a cluster-specific confusion matrix, and we leverage techniques from latent variable modeling to learn the confusion matrices with a small number of samples per user. This work also provides the first connection between the literature on peer prediction and the
literature on latent variable models for label aggregation.

Now, consider an online platform such as Google Maps, which needs to elicit feedback and ratings from users about various places and restaurants. It is crucial that this information be accurate, so that the new visitors are well informed. But providing inputs can be costly, and Google Maps initially tried to incentivize users by providing free Google drive depending on the number of contributions. In turn, this created a problem, in promoting “spam” feedback from users. In response to this challenge, chapter 2 presents a new peer prediction algorithm to incentivize users to invest effort and report truthfully.

Standard peer prediction methods such as output agreement compare responses from two users and award them if they agree. The reward schemes are built using the probabilities of various responses. However, these probabilities are a priori unknown, and to estimate them, the current approaches assume that a user answers several questions from the same distribution. There are two problems with these kinds of mechanisms – (1) the tasks are not necessarily homogeneous, and (2) the users can coordinate and misreport, i.e., report similar, but uninformed answers. In chapter 2, I propose an informed truthful peer prediction mechanism to handle heterogeneous tasks. Experiments on data provided by Google Maps suggest that the mechanism provides more resistance to coordinated misreports than existing peer prediction mechanisms.

(b) **Aggregation:** The field of social choice studies the aggregation of individual preferences towards a collective decision through the design of voting rules that are used to select a winning candidate. Traditionally, social choice theorists study settings where individuals have independent, subjective preferences, and in this setting there are strong impossibility results. However, there is an opportunity to rethink the traditional setup for modern domains such as crowdsourcing, social networks, and AI systems. Here the preferences are not necessarily independent, e.g., a user’s preference on a social network may be influenced by the preferences of the user’s neighbors. In chapter 3, I prove that a large class of voting rules become non-manipulable exponentially quickly in the number of voters when beliefs are positively-correlated, i.e., after a voter observes her own preference
then she believes that the preferences of the other voters are close to her own preference. Moreover, the result also holds when the voters’ beliefs are generated from standard rank-order models from statistical machine learning. This work characterizes, for the first time, the performance of standard voting rules when voters’ preferences are dependent. One drawback of standard methods for preference elicitation is that they only ask users to provide rank feedback over alternatives. We could obtain a large improvement in social welfare if users could also reveal cardinal information, e.g., whether the user’s utility for some alternative is above some threshold or not. In Chapter 4, I propose a unified view of voting by expanding the design space to include both the elicitation rule, whereby voters map their (cardinal) preferences to votes, and the aggregation rule, which transforms the reported votes into collective decisions. Intuitively, there is a tradeoff between the communication requirements of the elicitation rule (i.e., the number of bits of information that voters need to provide about their preferences) and the efficiency of the outcome of the aggregation rule, which we measure through distortion (i.e., how well the utilitarian social welfare of the outcome approximates the maximum social welfare in the worst case). The results of this study chart the Pareto frontier of the communication-distortion tradeoff. Our main result shows that any voting rule with $m$ alternatives and distortion $d$ must have communication complexity of at least $\Omega(m/d^2)$ with deterministic elicitation and $\Omega(m/d^3)$ with randomized elicitation. Furthermore, we provide a new voting rule that almost matches the lower bounds.

(c) Causal Inference: Consider the problem of determining the impact of a policy, e.g., what is the effect of a marketing campaign on a user’s purchasing behavior, or how do discounts affect the number of trips taken by a user on a ride-sharing platform? Traditionally, such questions are answered with the help of A/B tests, where a group of users is placed into control and another group is placed into treatment, and we compare the outcomes on the two groups. However, there are two main challenges when dealing with a large online system – (1) the pool of users visiting a platform may be inherently heterogeneous, and (2) treatments may be applied sequentially over a large number of
rounds. Historically, most datasets have been too small to uncover the heterogeneity in treatment effects. However, with the increase in data availability, and improvement in computational and statistical methods, we have started to be able to uncover heterogeneity in treatment effects. And yet, these attempts only consider cross-sectional data, where the treatment is applied for a single time period. In chapter 5, I present an improved version of the Marginal Structural Models (MSM) [RHB00], the most widely used method for performing causal inference on time-series cross-sectional data, to handle both the heterogeneity of users and treatment across a large number of time intervals. I propose a new form of MSM, which models the potential outcomes using a three-dimensional tensor, where the dimensions correspond to the agents, the time intervals, and the set of possible histories. Unlike traditional MSM, the new model allows the dimensions to increase with the number of agents and time intervals, allowing for heterogeneity of users and effects across time. I show how to efficiently estimate the model, and show on a simulated dataset that the estimator performs better than existing approaches to handling time-varying treatments.
Chapter 1

Peer Prediction with Heterogeneous Users

1.1 Introduction

Peer prediction is the technique of eliciting truthful information in the absence of verification by comparing an agent’s response with those of their peers. Peer prediction mechanisms incentivize users to provide honest reports when the reports cannot be verified, either because there is no objective ground truth or because it is costly to acquire the ground truth. Peer prediction mechanisms leverage correlation in the reports of peers in order to score contributions. The main challenge of peer prediction is to incentivize agents to put effort to obtain a signal or form an opinion and then honestly report to the system. In recent years, peer prediction has been studied in several domains, including peer assessment in massively open online courses (MOOCs) [SP16; GWLB16], for feedback on local places in a city [Man+16], and in the context of collaborative sensing platforms [RF15a].

The simplest peer prediction mechanism is output agreement, which pairs up two users and rewards them in the event that their reports agree (the ESP game [AD04] can be interpreted this way). However, output agreement is not incentive aligned for reports of a priori unlikely signals. As a result, there has been a lot of attention in recent years on finding methods that work more generally and provide robustness to coordinated misreports.
All existing, general methods are essentially restricted to settings with homogeneous participants, whose signal distributions are identical. This is a poor fit with many suggested applications of peer prediction. Consider for example, the problem of peer assessment in MOOCs. [DeB+13] and [WDR14] observe that students differ based on their geographical locations, educational backgrounds, and level of commitment, and indeed the heterogeneity of assessment is clear from a study of Coursera data [Kul+15]. [Sim+13] observed that the users participating in a citizen science project can be categorized into five distinct groups based on their behavioral patterns in classifying an image as a Supernovae or not. A similar problem occurs in determining whether news headline is offensive or not. Depending on which social community a user belongs to, we should expect to get different opinions [ZGDNM16]. Moreover, [AG17] report that leading to the 2016 U.S. presidential election, people were more likely to believe the stories that favored their preferred candidate; [Fou+17] find that there is very low connectivity among Trump and Clinton supporters on social networks, which leads to confirmation bias among the two groups and clear heterogeneity about how they believe whether a piece of news is “fake” or not.

One obstacle to designing peer prediction mechanisms for heterogeneous agents is an impossibility result. No mechanism can provide strict incentives for truth-telling to a population of heterogeneous agents without knowledge of their signal distributions [RF15b]. This negative result holds for minimal mechanisms, which only elicit signals and not beliefs from agents. One way to alleviate this problem, without going to non-minimal mechanisms, is to use reports from the agents across multiple tasks to estimate their signal distributions. This is our goal: we want to design minimal peer prediction mechanisms for heterogeneous agents that use reports from the agents for both learning and scoring. We also want to provide robustness against coordinated misreports.

As a starting point, we consider the correlated agreement (CA) mechanism proposed by [Shn+16]. If the agents are homogeneous and the designer has knowledge of their joint signal distribution, the CA mechanism is informed truthful, i.e. no strategy profile, even if coordinated, can provide more expected payment than truth-telling, and the expected payment under an uninformed strategy (where an agent’s report is independent of her signal) is strictly less
than the expected payment under truth-telling. These two properties remove any incentive for coordinated deviations and strictly incentivize the agents to put effort in acquiring signals, respectively. In a detail-free variation, in which the designer learns the signal distribution from reports, approximate incentive alignment is provided (still maintaining the second property as a strict guarantee.) The detail-free CA mechanism can be extended to handle agent heterogeneity, but a naïve approach would require learning the joint signal distributions between every pair of agents, and the total number of reports that need to be collected would be prohibitive for many settings.

**Our contributions:** We design the first minimal and detail-free mechanism for peer prediction with heterogeneous agents, where the learning component has sample complexity that is only linear in the number of agents, while providing an incentive guarantee of approximate informed truthfulness. Like the CA mechanism, this is a multi-task mechanism in that each agent makes reports across multiple tasks. Our mechanism is robust to any coordination between agents as long as the task assignments are such that from an agent’s perspective every other agent is equally likely to be her peer. Hence, our mechanism is robust to any coordination between agents that happens prior to task assignment. Our mechanism will also be robust to coordinations after task assignments as long as the agents are not able to figure out which agents are more likely to be their peers based on the identity of the tasks they are assigned. For example, in the context of a MOOC, the organizer can anonymize the homeworks to be graded, and hence, it will require a lot of effort for students to figure out whose homeworks they are grading even after the homeworks have been assigned for grading. Since our mechanism has a learning component, the task assignments to agents should also be such that both the goals of incentive alignment and learning are simultaneously achieved. We consider two assignment schemes under which these goals can be achieved and analyze the sample complexity of our methods for these schemes.

The mechanism clusters the agents based on their reported behavior\(^1\) and learns the pairwise correlations between these clusters. The clustering introduces one component of the

\(^1\)One could also consider clustering the agents based on their observable covariates as long as agents with similar covariates have similar ‘signal type’. However, in the applications that we consider in this paper, for example MOOCs, such covariates may not be observable, and hence, we only rely on agent reports for clustering.
incentive approximation, and could be problematic in the absence of a good clustering such that agents within a cluster behave similarly. Using eight real-world datasets, which contain reports of users on crowdsourcing platforms for multiple labeling tasks, we show that the clustering error is small in practice even when using a relatively small number of clusters. The second component of the incentive approximation stems from the need to learn the pairwise correlations between clusters; this component can be made arbitrarily small using a sufficient number of signal reports.

Another contribution of this work is to connect, we believe for the first time, the peer prediction literature with the extensive and influential literature on latent, confusion matrix models of label aggregation [DS79]. The Dawid-Skene model assumes that signals are generated independently, conditional on a latent attribute of a task and according to an agent’s confusion matrix. We cluster the agents based on their confusion matrices and then estimate the average confusion matrices within clusters using recent developments in tensor decomposition algorithms [Ana+14; Zha+16]. These average confusion matrices are then used to learn the pairwise correlations between clusters and design reward schemes to achieve approximate informed truthfulness.

In effect, the mechanism learns how to map one agent’s signal reports onto the signal reports of the other agents. For example, consider the context of a MOOC, in which an agent in the “accurate” cluster accurately provides grades, an agent in the “extremal” cluster only uses grades ‘A’ and ‘E’, and an agent in the “contrarian” cluster flips good grades for bad grades and vice-versa. The mechanism might learn to positively score an ‘A’ report from an “extremal” agent matched with a ‘B’ report from an “accurate” agent, or matched with an ‘E’ report from a “contrarian” agent for the same essay. In practice, our mechanism will train on the data collected during a semester of peer assessment reports, and then cluster the students, estimate the pairwise signal distributions between clusters, and accordingly score the students (i.e., the scoring is done retroactively).
1.1.1 Related Work

We provide a brief review of the related work in peer prediction, and suggest [FR17] for a detailed discussion. We focus our discussion on related work about minimal mechanisms, but remark that we are not aware of any non-minimal mechanisms (following from the work of [Pre04]) that handle agent heterogeneity. [MRZ05] introduce the peer prediction problem, and proposed an incentive-aligned mechanism for the single-task setting. However, their mechanism requires knowledge of the joint signal distribution and is vulnerable to coordinated misreports. In regard to coordinated misreports, [JF+09] show how to eliminate uninformative, pure-strategy equilibria through a three-peer mechanism, and [KLS16] provide a method to design robust, single-task, binary signal mechanisms (but need knowledge of the joint signal distribution). [FW17] provide a characterization of minimal (single task) peer prediction mechanisms.

[WP13] introduce the combination of learning and peer prediction, coupling the estimation of the signal prior together with the shadowing mechanism. Some results make use of reports from a large population. [RF15a], for example, establish robust incentive properties in a large-market limit where both the number of tasks and the number of agents assigned to each task grow without bound. [RFJ16] provide complementary theoretical results, giving a mechanism in which truthfulness is the equilibrium with the highest payoff in the asymptote of a large population and with a structural property on the signal distribution.

[DG13] show that robustness to coordinated misreports can be achieved for binary signals in a small population by using a multi-task mechanism. The idea is to reward agents if they provide the same signal on the same task, but punish them if one agent’s report on one task is the same as another’s on a different task. The Correlated Agreement (CA) mechanism [Shn+16] generalizes this mechanism to handle multiple signals, and uses reports to estimate the correlation structure on pairs of signals without compromising incentives. In related work, [KS16] show that many peer prediction mechanisms can be derived within a single information-theoretic framework. Their results use different technical tools than those used by [Shn+16], and also include a different multi-signal generalization of the Dasgupta-Ghosh mechanism that provides robustness against coordinated misreports in the limit of a large number of tasks.
Shnayder, Frongillo, and Parkes [SFP16] adopt replicator dynamics as a model of population learning in peer prediction, and confirm that these multi-task mechanisms (including the mechanism by Kamble et al. [Kam+15]) are successful at avoiding uninformed equilibria.

There are very few results on handling agent heterogeneity in peer prediction. For binary signals, the method of [DG13] is likely to be an effective solution because their assumption on correlation structure will tend to hold for most reasonable models of heterogeneity. But it will break down for more than two signals, as explained by Shnayder et al. [Shn+16]. Moreover, although the CA mechanism can in principle be extended to handle heterogeneity, it is not clear how the required statistical information about joint signal distributions can be efficiently learned and coupled with an analysis of approximate incentives. For a setting with binary signals and where each task has one of a fixed number of latent types, Kamble et al. [Kam+15] design a mechanism that provides strict incentive compatibility for a suitably large number of heterogeneous agents, and when the number of tasks grows without bound (while allowing each agent to only provide reports on a bounded number of tasks). Their result is restricted to binary signals, and requires a strong regularity assumption on the generative model of signals. [KS16] design an information theoretic framework for peer prediction. Their mechanism pays each agent the mutual information between her report and her peer’s report. This mechanism can be extended to the heterogeneous agents setting as long as we can measure the mutual information between all pairs of agents. However, such a mechanism would require the agents to provide reports on a large number of tasks.

Finally, we consider only binary effort of a user, i.e. the agent either invests effort and receives an informed signal or does not invest effort and receives an uninformed signal. Shnayder et al. [Shn+16] work with the binary effort setting and provide strict incentive for being truthful. Therefore, as long as the mechanism designer is aware of the cost of investing effort, the payments can be scaled to cover the cost of investing effort. The importance of motivating effort in the context of peer prediction has also been considered by Liu and Chen [LC17b] and Witkowski et al. [Wit+13].² See Mandal et al. [Man+16] for a setting with

²Cai, Daskalakis, and Papadimitriou [CDP15] work in a different model, showing how to achieve optimal statistical estimation from data provided by rational agents. They only focus on the cost of effort. They do not consider possible misreports, and thus their mechanism is also vulnerable to coordinated misreports.
heterogeneous tasks but homogeneous agents. Liu and Chen [LC17a] also designed single-task peer prediction mechanism for the same setting but only when each task is associated with a latent ground truth.

1.2 Model

Let notation $[t]$ denote $\{1, \ldots, t\}$ for $t \in \mathbb{N}$. We consider a population of agents $P = [\ell]$, and use indices such as $p$ and $q$ to refer to agents from this population. There is a set of tasks $M = [m]$. For example, a task can be either grading an essay or answering a question in an online rating system. When an agent performs a task, she receives a signal from $N = [n]$. Such a signal usually indicates the quality of the task i.e. the number of points assigned to the essay or how good the food is at a restaurant. The agents need to put some effort to get an informative signal about the task. As mentioned before, we assume that the effort of an agent is binary i.e. either the agent puts full effort and receives an informative signal or the agent puts no effort and receives a signal drawn uniformly at random. We also assume that the tasks are ex ante identical, that is, the signals of an agent for different tasks are sampled i.i.d. For example, in the essay grading scenario, if the essays assigned to any student are drawn uniformly at random from a large population of essays, the student’s signal distribution for an assigned essay is ex ante almost identical to any other assigned essay.

Each agent is assigned a set of tasks and she decides, for each task, whether to put effort and receive an informative signal or put no effort and receive a random signal. This provides the agent with a set of signals, one for each task. Then the agent reports back to mechanism designer a set of signals, one for each assigned task. Before putting any effort to receive informative signals, the agents have no knowledge about the tasks apart from the fact they are ex-ante identical. Once the agents receive their signals, their reports are determined completely by these signals. In other words, the agents do not use any additional information to determine their reports. We will assume that, for each task, the message space and the signal space are the same. Since the payment made to the agents depend on their reported signals (messages), the reported signals can be very different than the observed signals. The goal of a peer prediction
mechanism is to ensure that the agents put effort in all the tasks and report their signals truthfully. For the MOOC setting, a student spends some amount of time to figure out the grade of each of her assigned essays. She might also decide to not look at an essay and report an arbitrary grade. The goal of our mechanism is to ensure that the students put some effort to determine the grades of the essays and report them truthfully back to the platform. We work in the setting where the agents are heterogeneous, i.e., the distribution of signals can be different for different agents. These differences are captured by the agents’ types and we say that the agents vary by signal type. In peer prediction, we compare the reports of an agent to the reports of their peers on the same tasks, and hence, we also need to talk about joint signal distribution of pairs of agents in addition to the signal distribution of an individual agent. In our setting, these joint signal distributions can be different for different pair of agents.

Let $S_p, S_q$ denote random variables for the signal observed by agents $p$ and $q$ on some task. Let $D_{p,q}(i,j)$ denote the joint probability that agent $p$ receives signal $i$ while agent $q$ receives signal $j$ on a task, i.e. $D_{p,q}(i,j) = \Pr(S_p = i, S_q = j)$. Let $D_p(i)$ and $D_q(j)$ denote the corresponding marginal probabilities, i.e. $D_p(i) = \Pr(S_p = i)$ and $D_q(j) = \Pr(S_q = j)$. An important part of our mechanisms are the delta matrices which are defined as follows. We define the Delta matrix $\Delta_{p,q}$ between agents $p$ and $q$ as

$$\Delta_{p,q}(i,j) = D_{p,q}(i,j) - D_p(i) \cdot D_q(j), \quad \forall i, j \in [n].$$

The delta matrices capture the correlation between pairs of realized signals. For example, if $\Delta_{p,q}(1,2) = D_{p,q}(1,2) - D_p(1)D_q(2) > 0$. This implies that $\Pr[S_p = 1|S_q = 2] > \Pr[S_p = 1]$. Therefore, the event of agent $p$ observing signal 1 is positively correlated with the event of agent $q$ observing signal 2. This would also mean that the event that agent $p$ receives signal 1 and agent $q$ receives signal 2 is more likely when these signals are for the same task, than when they are for different tasks. Our mechanism will use these correlations to decide the score for an agent given the reports of the agent and her peers. The correlated agreement (CA) mechanism [Shn+16] also uses these delta matrices to construct a scoring mechanism for agent reports, however, they work in a setting where agents are exchangeable, i.e. the delta matrix $\Delta_{p,q}$ is the same for all pairs $p, q$ of agents.
Example 1.1. For two agents $p$ and $q$, consider the following joint signal distribution $D_{p,q}$ is

$$
D_{p,q} = \begin{bmatrix}
0.2 & 0.3 \\
0.1 & 0.4 
\end{bmatrix}
$$

with marginal distributions $D_p = [0.5 \ 0.5]$ and $D_q = [0.3 \ 0.7]$, the Delta matrix $\Delta_{p,q}$ is

$$
\Delta_{p,q} = \begin{bmatrix}
0.2 & 0.3 \\
0.1 & 0.4 
\end{bmatrix} - \begin{bmatrix}
0.5 & 0.5 \\
0.3 & 0.7 
\end{bmatrix} = \begin{bmatrix}
0.05 & -0.05 \\
-0.05 & 0.05 
\end{bmatrix}.
$$

An agent’s strategy defines, for every signal it may receive and each task it is assigned, a probability distribution over signals it will report. Shnayder et al. [Shn+16] show that it is without loss of generality for the class of mechanisms we study in this paper to assume that an agent’s strategy is uniform across different tasks. Hence, we will make the assumption that an agent’s strategy is uniform across tasks. Formally, let $R_p$ denote the random variable for the report of agent $p$ for a given task. The strategy of agent $p$, denoted $F^p$, defines the distribution of reports for each possible signal $i$, with $F^p_{ir} = \Pr(R_p = r|S_p = i)$. Therefore if there are $n$ signals then the strategy $F^p : [n] \rightarrow \mathcal{P}_n$, where $\mathcal{P}_n$ is the set of all possible distributions with support in $[n]$. The collection of agent strategies, denoted $\{F^p\}_{p \in P}$, is the strategy profile. A strategy of agent $p$ is informed if there exist distinct $i, j \in [n]$ and $r \in [n]$ such that $F^p_{ir} \neq F^p_{jr}$, i.e., if not all rows of $F^p$ are identical. We say that a strategy is uninformed otherwise.

1.2.1 Multi-Task Peer Prediction

In this paper we consider multi-task peer prediction mechanisms defined in Shnayder et al. [Shn+16], and extend them to the setting of heterogeneous agents. In these mechanisms, each agent performs multiple-tasks and the score of an agent depends on its reports and the reports of its peers. For each agent, a random subset of her tasks is designated as bonus tasks, and its complement is designated as penalty tasks, without the knowledge of the agent. These mechanisms are characterized by scoring matrices for each pair of agents, which are used to score agents’ reports. In our mechanism, the scoring matrix $S_{p,q} : [n] \times [n] \rightarrow \{0,1\}$ for agent pair $p$ and $q$ will be such that $S_{p,q}(i,j) = 1$ when the event that agent $p$ receives signal $i$ is
positively correlated with the event that agent $q$ receives signal $j$ on the same task, otherwise $S_{p,q}(i,j) = 0$. We will thus use the delta matrices (which will be learnt from agent reports) to design these scoring matrices.

For signals $i$ and $j$, if $S_{p,q}(i,j) = 1$, then, for each bonus task of an agent $p$, we will add 1 to her score for reporting $i$ when the report of its peer agent $q$ on the same task is $j$, otherwise we will not add anything. Additionally, for each bonus task of agent $p$, we randomly select a penalty task and subtract some score her total score based on her report on the penalty task. For signals $i$ and $j$, if $S_{p,q}(i,j) = 1$, then we will subtract 1 from her score for reporting $i$ on the penalty task when the report of its peer agent $q$ on a different task is $j$, otherwise we will not subtract anything. The penalty is included in the score in order to avoid ‘uninformative equilibria’ where agents agree to report the same signal on every task without investing effort in gathering the signals. The total score of an agent will be sum of all the scores over all bonus tasks calculated this way.

In our mechanism the score of an agent on a bonus task will be ‘+1’ when its report is positively correlated with the report of its peer agent on the same task. The score of an agent on a penalty task will be ‘-1’ when its report is positively correlated with the report of its peer on a different task. The intuition behind our mechanism is that when signals $i$ and $j$ of agents $p$ and $q$ are correlated then it will be more likely that agents receive this pair of signals on tasks they share than on tasks they do not share. Hence, the overall score will be positive in expectation, when agents are truthful. Whenever the agents use any uninformed strategy then the event that ‘the report of agent $p$ is $i$ and the report of agent $q$ is $j’ is as likely to happen when they perform the same task as it is when they perform different tasks. Hence, the expected payment of any uninformed strategy will be zero. The correlated agreement (CA) mechanism [Shn+16] also uses a scoring matrix for scoring agent. However, in their homogeneous setting only one scoring matrix is required because the delta matrices are the same for each pair of agents. In our heterogeneous setting we have to use different scoring matrices for different pairs of agents.

Formally, for agent $p$, we denote the set of her bonus tasks by $M^p_1$ and the set of her penalty tasks by $M^p_2$. To calculate the payment to an agent $p$ for a bonus task $t \in M^p_1$, we do the
following:

1. Randomly select an agent \( q \in P \setminus \{ p \} \) such that \( t \in M^p_1 \), and the set \( M^p_2 \cup M^q_2 \) has at least 2 distinct tasks, and call \( q \) the peer of \( p \).

2. Pick tasks \( t' \in M^p_2 \) and \( t'' \in M^q_2 \) randomly such that \( t' \neq t'' \) (\( t' \) and \( t'' \) are the penalty tasks for agents \( p \) and \( q \) respectively)

3. Let the reports of agent \( p \) on tasks \( t \) and \( t' \) be \( r^t_p \) and \( r'^t_p \), respectively and the reports of agent \( q \) on tasks \( t \) and \( t'' \) be \( r^t_q \) and \( r''_q \) respectively.

4. The payment of agent \( p \) for task \( t \) is then \( S_{p,q}(r^t_p, r^t_q, r'^t_p, r''_q) \).

The total payment to an agent is the sum of payments for the agent’s bonus tasks.

1.2.2 Task Assignments

Since we work in the setting where agents perform multiple tasks, and hence, it is important to address how these tasks are assigned to agents. Our mechanism has two requirements from any task assignment–

1. From an agent’s perspective, every other agent is equally likely to be her peer. This requires agents not to know each other’s task assignments before deciding a strategy. For example, if agents of one ‘type’ are more likely to be peers with agents of another ‘type’ based on their task assignments, then they can coordinate amongst themselves to decide a more profitable strategy than truth-telling. Our mechanism will be robust to coordinations that happen before the task assignments. Our mechanism will also be robust to coordinations after task assignments as long as the agents are not able to figure out which agents are more likely to be their peers based on the identity of the tasks they are assigned.

2. We should always be able to find a peer agent \( q \) for any agent \( p \). Precisely, the tasks are assigned in a way that for every agent \( p \) we can find a peer agent \( q \) such that \( q \) has performed at least one bonus task that \( p \) has performed, and we have reports from \( p \) and \( q \) for two different tasks which are not the same as the bonus task.
In addition, our mechanism has a learning component, where we learn about the correlation between agents’ signals, and also cluster agents into groups. Hence, in order to learn these quantities, we need to collect sufficient reports from each agent. This imposes some other requirements for the task assignment. In Section 1.4 we propose two task assignment schemes that a principal can use that satisfy all these requirements.

1.2.3 Expected Payments

The expected payment to agent \(p\) under strategy profile \(\{F^q\}_{q \in P}\) for any bonus task performed by her, equal across all bonus tasks as the tasks are *ex ante* identical, is given as

\[
\begin{align*}
    u_p(F^p, \{F^q\}_{q \neq p}) &= \frac{1}{\ell} \sum_{q \neq p} \left\{ \sum_{i,j} D_{p,q}(i,j) \sum_{r_p,r_q} F^p_{i,p} F^q_{j,q} S_{p,q}(r_p, r_q) \right. \\
    & \quad - \sum_i D_p(i) \sum_{r_p} \sum_j D_q(j) \sum_{r_q} F^p_{i,p} F^q_{j,q} S_{p,q}(r_p, r_q) \left. \right\} \\
    &= \frac{1}{\ell} \sum_{q \neq p} \left\{ \sum_{i,j} (D_{p,q}(i,j) - D_p(i)D_q(j)) \sum_{r_p,r_q} F^p_{i,p} F^q_{j,q} S_{p,q}(r_p, r_q) \right\} \\
    &= \frac{1}{\ell} \sum_{q \neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{i,p} F^q_{j,q} S_{p,q}(r_p, r_q)
\end{align*}
\]

(1.2)

1.2.4 Informed Truthfulness

Following Shnayder et al. [Shn+16], we define the notion of approximate informed truthfulness for a multi-task peer prediction mechanism.

**Definition 1.1.** (**\(\epsilon\)-informed truthfulness**) We say that a multi-task peer prediction mechanism is \(\epsilon\)-informed truthful, for some \(\epsilon \geq 0\), if and only if for every strategy profile \(\{F^q\}_{q \in P}\) and every agent \(p \in P\), we have \(u_p(\Pi, \{\Pi\}_{q \neq p}) \geq u_p(F^p, \{F^q\}_{q \neq p}) - \epsilon\), where \(\Pi\) is the truthful strategy, and \(u_p(\Pi, \{\Pi\}_{q \neq p}) > u_p(F^p_0, \{F^q\}_{q \neq p})\) where \(F^p_0\) is an uninformed strategy.

An \(\epsilon\)-informed truthful mechanism ensures that every agent prefers (up to \(\epsilon\)) the truthful strategy profile over any other strategy profile, and strictly prefers the truthful strategy profile over any uninformed strategy. Moreover, no coordinated strategy profile provides more expected utility than the truthful strategy profile (upto \(\epsilon\)). For a small \(\epsilon\), this is responsive to
the main concerns about incentives in peer prediction: a minimal opportunity for coordinated
manipulations, and a strict incentive to invest effort in collecting and reporting an informative
signal.\(^3\)

### 1.2.5 Learning and Agent Clustering

Suppose that one knows \(\Delta_{p,q}\) for every pair of agents, then one can calculate the scoring
matrices \(S_{p,q}\) according to these delta matrices and use these scoring matrices to score the
agents. It is not hard to prove (see Lemma 1.3 for a proof) that such an extension of the CA
mechanism will be informed truthful. However, we seek to design a detail-free mechanism
where one does not have the knowledge of delta matrices, and one needs to learn them from
agent reports. However, it would require \(\Omega(\ell^2)\) samples to learn the delta matrices between
every pair of agents, which will often be impractical. Rather, the number of reports in a
practical mechanism should scale closer to linearly in the number of agents.

In response, we will assume that agents can be (approximately) clustered into a bounded
number \(K\) of agent signal types, such that agents of the same type have similar signal distribu-
tions. Hence, a cluster of agents will be treated as a meta-agent, and we will work with signal
distributions of these meta-agents. Formally, let \(G_1, \ldots, G_K\) denote a partitioning of agents into
\(K\) clusters. With a slight abuse of notation, we also use \(G(p)\) to denote the cluster to which
agent \(p\) belongs.

In order to reduce the sample complexity of our mechanism, we want that the clustering of
agents to be such that for each pair \(p,q\) of agents, the signals of meta-agents (clusters) \(G(p)\)
and \(G(q)\) are correlated in a similar manner as the signals of agents \(p\) and \(q\). With this in mind,
for \(s, t \in [K]\), let us define the cluster Delta matrix between clusters \(G_s\) and \(G_t\) to be the average
signal correlation taken over all pairs of agents \(p \in G_s\) and \(q \in G_t\), i.e.

\[
\Delta_{G_s, G_t} = \begin{cases} 
\frac{1}{|G_s| |G_t|} \sum_{p \in G_s, q \in G_t} \Delta_{p,q} & \text{if } s \neq t \\
\frac{1}{|G_s|^2 - |G_s|} \sum_{p, q \in G_s, p \neq q} \Delta_{p,q} & \text{if } s = t 
\end{cases}
\]

\(^3\)We do not model the cost of effort explicitly in this paper, but a binary cost model (effort \(\rightarrow\) signal, no-effort \(\rightarrow\) no signal) can be handled in a straightforward way. See Shnayder et al. [Shn+16].
Now, the clustering of agents should be such that for each pair of agents \( p, q \), we should be able to use \( \Delta_{G(p),G(q)} \) as a proxy for \( \Delta_{p,q} \). This will allow use learn Delta matrices for every cluster pair, instead of learning Delta matrices for every agent pair. This intuition results in the following definition of an \( \epsilon_1 \)-accurate clustering.

**Definition 1.2.** We say that clustering \( G_1, \ldots, G_K \) is \( \epsilon_1 \)-accurate, for some \( \epsilon_1 \geq 0 \), if for every pair of agents \( p, q \in P \),

\[
\| \Delta_{p,q} - \Delta_{G(p),G(q)} \|_1 \leq \epsilon_1,
\]

(1.3)

where \( \Delta_{G(p),G(q)} \) is the cluster Delta matrix between clusters \( G(p) \) and \( G(q) \).

**Example 1.2.** Let there be 4 agents \( p, q, r \), and \( s \). Let the pairwise Delta matrices be the following

\[
\Delta_{p,q} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}, \quad \Delta_{p,r} = \begin{bmatrix} -0.15 & 0.15 \\ 0.15 & -0.15 \end{bmatrix}, \quad \Delta_{p,s} = \begin{bmatrix} -0.05 & 0.05 \\ 0.05 & -0.05 \end{bmatrix},
\]

\[
\Delta_{q,r} = \begin{bmatrix} -0.05 & 0.05 \\ 0.05 & -0.05 \end{bmatrix}, \quad \Delta_{q,s} = \begin{bmatrix} -0.15 & 0.15 \\ 0.15 & -0.15 \end{bmatrix}, \quad \Delta_{r,s} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}.
\]

In this example, agents \( p \) and \( q \) tend to agree with each other, while agents \( r \) and \( s \) tend to agree with each other while disagreeing with \( p \) and \( q \). Let the clustering be \( G_1, G_2 \) where \( p,q \) belong to \( G_1 \) and \( r,s \) belong to \( G_2 \). Then the cluster Delta matrices are the following

\[
\Delta_{G_1,G_1} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}, \quad \Delta_{G_1,G_2} = \begin{bmatrix} -0.1 & 0.1 \\ 0.1 & -0.1 \end{bmatrix}, \quad \Delta_{G_2,G_2} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}.
\]

It is easy to observe that \( G_1, G_2 \) is a \( 0.2 \)-accurate clustering.

Our mechanism will use an estimate of \( \Delta_{G(p),G(q)} \) (instead of \( \Delta_{p,q} \)) to define the scoring matrix \( S_{p,q} \). Thus, the incentive approximation will directly depend on the accuracy of the clustering as well as how good the estimate of \( \Delta_{G(p),G(q)} \) is.

There is an inverse relationship between the number of clusters \( K \) and the clustering accuracy \( \epsilon_1 \): the higher the \( K \), the lower the \( \epsilon_1 \). In the extreme, we can let every agent be a separate cluster (\( K = \ell \)), which results in \( \epsilon_1 = 0 \). But a small number of clusters is essential for a reasonable sample complexity as we need to learn \( O(K^2) \) cluster Delta matrices. For instance,
in Example 1.2 we need to learn 3 Delta matrices with clustering, as opposed to 6 without clustering. In Section 1.4, we give a learning algorithm that can learn all the pairwise cluster Delta matrices with \( O(K) \) samples given a clustering of the agents. In Section 1.5, we show using real-world data that a reasonably small clustering error can be achieved with relatively few clusters.

### 1.3 Correlated Agreement for Heterogeneous Agents

In this section we define our Correlated Agreement for Heterogeneous Agents (CAHU) mechanism, presented as Algorithm 1.1. Our mechanism builds upon the multi-task Correlated Agreement (CA) mechanism of Shnayder et al. [Shn+16], which uses the correlation between signals of different agents to design a scoring matrix to score the agents. However, since we work in a heterogeneous setting we will need to design different scoring matrices for different pairs of agents, based on the different correlations between different pairs.

For intuition, consider the case when one has knowledge of the Delta matrices for all pairs of agents. In this case, in the multi-task peer prediction framework defined in Section 1.2.1, the scoring matrices \( S_{p,q} \) can be defined such that \( S_{p,q}(i,j) = 1 \) when \( \Delta_{p,q} > 0 \), and \( S_{p,q}(i,j) = 0 \) otherwise. Such a mechanism will be 0-informed truthful, as we prove in Lemma 1.3.

However, in order to design a detail-free mechanism with low sample complexity, we will assume that we have a clustering of agents such that the average cluster Delta matrices can be used as a proxy for agent Delta matrices. Hence, our mechanism works with a clustering of agents, and uses the cluster Delta matrices to design scoring matrices for pairs of agents. Here, we will describe our mechanism when a clustering as well as estimates of cluster Delta matrices are given as inputs to the mechanism. In Section 1.4, we will see how one can learn such a clustering and estimates of Delta matrices from agents reports.

Specifically, CAHU takes as input a clustering \( G_1, \ldots, G_K \) of agents. It also takes as input matrices \( \{\tilde{\Delta}_{G_p, G_q}\}_{s,t \in [K]} \) which are estimates of the cluster Delta matrices \( \{\Delta_{G_p, G_q}\}_{s,t \in [K]} \) defined in Section 1.2.5. The scoring matrix \( S_{p,q} \) for agent pair \( p \) and \( q \) is then defined such that \( S_{p,q}(i,j) = 1 \) when \( \Delta_{G(p), G(q)} > 0 \), and \( S_{p,q}(i,j) = 0 \) otherwise, where \( G(p) \) and \( G(q) \) denote the
clusters that \( p \) and \( q \) belong to, respectively. The CAHU mechanism then calculates the reward of an agent according to the framework of multi-task peer prediction discussed in Section 1.2.1. This would mean that an agent \( p \) gets a positive score whenever her report and her peer \( q \)'s report on a bonus task is such that there is positive correlation between the corresponding signals of clusters \( G(p) \) and \( G(q) \). However, we also include a penalty when this happens on different tasks. The idea is that if the clustering is \( \epsilon_1 \)-accurate and the estimates of cluster Delta matrices are accurate, then the mechanism should retain its truthfulness properties. With this in mind, we define an \((\epsilon_1, \epsilon_2)\)-accurate input to the algorithm as follows

**Definition 1.3.** We say that a clustering \( \{G_s\}_{s \in [K]} \) and the estimates \( \{\overline{G}_s, G_t\}_{s, t \in [K]} \) are \((\epsilon_1, \epsilon_2)\)-accurate if

\[
\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leq \epsilon_1 \quad \text{for all agents } p, q \in P, \text{ i.e., the clustering is } \epsilon_1 \text{-accurate, and}
\]

\[
\|\Delta_{G_s, G_t} - \overline{\Delta}_{G_s, G_t}\|_1 \leq \epsilon_2 \quad \text{for all clusters } s, t \in [K], \text{ i.e., the cluster Delta matrix estimates are } \epsilon_2 \text{-accurate.}
\]

An \( \epsilon_1 \) clustering intuitively means that if we pick one agent from cluster \( G_s \) and another agent from cluster \( G_t \) then their signal correlation is determined by the pair of clusters up to an error \( \epsilon_1 \) and is independent of the identities of the agents. On the other hand \( \epsilon_2 \)-accurate clustering simple means that we can estimate the cluster delta matrices up to an error \( \epsilon_2 \). When we have a clustering and estimates of the delta matrices which are \((\epsilon_1, \epsilon_2)\)-accurate, we prove that the CAHU mechanism is \((\epsilon_1 + \epsilon_2)\)-informed truthful. In Section 1.4, we present algorithms that can learn an \( \epsilon_1 \)-accurate clustering and \( \epsilon_2 \)-accurate estimates of cluster Delta matrices.

Throughout the rest of this section, we will use \( \epsilon_1 \) to denote the clustering error and \( \epsilon_2 \) to denote the learning error. We remark that the clustering error \( \epsilon_1 \) is determined by the level of similarity present in agent signal-report behavior, as well as the number of clusters \( K \) used, whereas the learning error \( \epsilon_2 \) depends on how many samples the learning algorithm sees.

### 1.3.1 Analysis of CAHU

In this section we will prove the incentive properties of the CAHU mechanism. We will first present an overview of the proof, before presenting it formally. Recall that the expected
payment of an agent in this setting is the following:

\[
u_p(F^p, \{F^q\}_{q \neq p}) = \frac{1}{\ell - 1} \sum_{q \neq p} \Delta_{p,q} \sum_{i,j} F^p_{iF^q_j} S_{p,q}(r_p, r_q).
\]

One can think of the expected payment to an agent \( p \) to be the average over all other agents \( q \), the expected payment when \( q \) is \( p \)'s peer agents. The expected payment when \( q \) is \( p \)'s peer agent is given by the quantity \( \sum_{i,j} \Delta_{p,q} \sum_{r_p, r_q} F^p_{iF^q_j} S_{p,q}(r_p, r_q) \).

For intuition, let us only consider deterministic strategies in this discussion. Our proof covers general randomized strategies in this discussion. For deterministic strategies we have that

\[
\sum_{r_p, r_q} F^p_{iF^q_j} S_{p,q}(r_p, r_q) = S_{p,q}(F^p_i, F^q_j),
\]

where \( F^p_i \) and \( F^q_j \) denote (deterministic) reports of agents \( p \) and \( q \) given signals \( i \) and \( j \), respectively. In this case the expected payment for \( p \) when \( q \) is her peer is \( \sum_{i,j} \Delta_{p,q} \sum_{r_p, r_q} F^p_{iF^q_j} S_{p,q}(r_p, r_q) \). Suppose that \( \Delta_{p,q} \) has positive diagonals, and negative non-diagonals, and the scoring matrix \( S_{p,q} \) is the identity matrix, then it is not hard to see that the maximum value
of \( \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(F_p^i, F_q^j) \) for any deterministic \( F_p \) and \( F_q \) is the trace of the matrix \( \Delta_{p,q} \). Moreover, this maximum is achieved when \( F_p \) and \( F_q \) are truthful. Also, suppose that agents \( p \) and \( q \) adopt an uniformed strategy, say reporting ‘1’ for every task, then the expected payment is \( \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(1,1) \) which is zero since the sum of the entries of the Delta matrices is always zero. For the general case, we will show that the maximum expected payment to \( p \) when agent \( q \) is her peer is given by \( \sum_{i,j} \Delta_{p,q}(i,j) \cdot \text{Sign}(\Delta_{p,q}(i,j)) \). Hence, when \( S_{p,q} = \text{Sign}(\Delta_{p,q}(i,j)) \), then this maximum is achieved when the agents are truthful. Also, the payment of any uninformed strategy is 0. Since, this holds for any peer agent \( q \), this would imply informed truthfulness of the mechanism where \( S_{p,q} = \text{Sign}(\Delta_{p,q}(i,j)) \). A similar argument also follow for any mixed strategies. A formal proof is presented in Lemma 1.3, and is very similar to the proof of informed truthfulness of the CA mechanism [Shn+16].

However, we use approximate cluster Delta matrices instead of agent Delta matrices, to design the scoring matrices. Hence, we need to additionally worry about the effect of approximations due to clustering and learning on the incentive properties of our mechanisms. We will show that even under these approximation a truthful strategy will attain an expected reward that is close to the maximum possible expected reward. Precisely, we will show that when the clustering is \( \epsilon_1 \)-accurate and the cluster Delta matrix estimates are \( \epsilon_2 \)-accurate then the expected reward of a truthful strategy is at most \( (\epsilon_1 + \epsilon_2) \) away from the maximum reward under any strategy and scoring matrices. Also, the expected reward of any uninformed strategy will always be zero. This will imply that CAHU is \( (\epsilon_1 + \epsilon_2) \)-informed truthful.

We will first need the following technical lemmas before proceeding to the main proof. Appendix A.1 contains their proofs.

**Lemma 1.1.** For any matrix \( \tilde{S} \in \{0,1\}^{n \times n} \), and any probability distributions \( \psi \in \mathcal{P}_n \) and \( \phi \in \mathcal{P}_n \), where \( \mathcal{P}_n \) is the set of all probability distributions over \([n] \), we have that

\[
0 \leq \sum_{r_1,r_2 \in [n]} \psi_{r_1} \tilde{S}(r_1,r_2) \phi_{r_2} \leq 1.
\]

We now prove another technical lemma which gives an upper bound on the maximum payoff to an agent \( p \) under any scoring matrix.
**Theorem 1.1.** Let \( \{\hat{S}_{p,q}\}_{p,q \in P} \) be an arbitrary set of scoring matrices where \( \hat{S}_{p,q} \in \{0,1\}^{n \times n} \) denotes the score matrix for agent \( p \) and agent \( q \). Then for every strategy profile \( \{F^q\}_{q \in P} \) we have that

\[
\sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{r_p} F^q_{r_q} \hat{S}_{p,q}(r_p,r_q) \leq \sum_{i,j: \Delta_{p,q}(i,j) > 0} \Delta_{p,q}(i,j).
\]

We will now analyze our mechanism formally using the above lemmas. The derivation of the following result closely follows a similar analysis due to Shnayder et al. [Shn+16]. We use \( u_p^*(\cdot) \) to denote the utility of agent \( p \) when the scoring matrices are \( \text{Sign}(\Delta_{p,q}(i,j)) \), for all pairs \( p,q \).

**Lemma 1.3.** For a strategy profile \( \{F^q\}_{q \in P} \) and an agent \( p \in P \), define

\[
u_p^*(F^p, \{F^q\}_{q \neq p}) = \frac{1}{\ell - 1} \sum_{q \neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{r_p} F^q_{r_q} S_{p,q}^*(r_p,r_q)
\]

where \( S_{p,q}^*(i,j) = \text{Sign}(\Delta_{p,q}(i,j)) \) for all \( i,j \in [n] \). Then, \( u_p^*(I, \{I\}_{q \neq p}) \geq u_p^*(F^p, \{F^q\}_{q \neq p}) \). Moreover, for any uninformed strategy \( F^p \), \( u_p^*(I, \{I\}_{q \neq p}) > u_p^*(F^p, \{F^q\}_{q \neq p}) \). This implies informed-truthfulness of the mechanism where \( S_{p,q}^* \) is used for scoring agents \( p \) and \( q \).

We now prove our main theorem that \((\varepsilon_1 + \varepsilon_2)\)-informed truthfulness holds when \((\varepsilon_1, \varepsilon_2)\)-accurate clustering and learning holds.

**Theorem 1.1.** With \((\varepsilon_1, \varepsilon_2)\)-accurate clustering and learning, mechanism CAHU is \((\varepsilon_1 + \varepsilon_2)\)-informed truthful if \( \min_p u_p^*(I, \{I\}_{q \neq p}) > \varepsilon_1 + \varepsilon_2 \). In particular,

1. For every profile \( \{F^q\}_{q \in P} \) and agent \( p \in P \), we have \( u_p(I, \{I\}_{q \neq p}) \geq u_p(F^p, \{F^q\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2 \).
2. For any uninformed strategy \( F^p_0 \), \( u_p(I, \{I\}_{q \neq p}) \geq u_p(F^p_0, \{F^q\}_{q \neq p}) \).

**Proof.** Fix a strategy profile \( \{F^q\}_{q \in P} \). We first show that \( u_p^*(I, \{I\}_{q \neq p}) \geq u_p(F^p, \{F^q\}_{q \neq p}) \), and then show that \( |u_p^*(I, \{I\}_{q \neq p}) - u_p(I, \{I\}_{q \neq p})| \leq \varepsilon_1 + \varepsilon_2 \). These together imply that \( u_p(I, \{I\}_{q \neq p}) \geq u_p(F^p, \{F^q\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2 \). For the former, we first observe (similarly, as in proof of Lemma 1.3) that the utility of truthful reporting when the scoring matrix \( S_{p,q}^*(i,j) = \text{Sign}(\Delta_{p,q}(i,j)) \), is given by

\[
u_p^*(I, \{I\}_{q \neq p}) = \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j: \Delta_{p,q}(i,j) > 0} \Delta_{p,q}(i,j)
\]
The utility \( u_p(F^p, \{ F^q \}_{q\in P\setminus \{p\}}) \) of an agent \( p \) for any strategy profile \( \{ F^p, \{ F^q \}_{q\in P\setminus \{p\}} \} \) under our mechanism, when the scoring matrix \( S_{p,q} = \Sign(\Delta_G(p),G(q)) \), is given by

\[
 u_p(F^p, \{ F^q \}_{q\in P\setminus \{p\}}) = \frac{1}{\ell - 1} \sum_{q \in P\setminus \{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{i,r_p} F^q_{j,r_q} S_{p,q}(r_p, r_q)
\]

Now, using Lemma 1.2 and the expressions for \( u^*_{p}(\mathbb{I}, \{ \mathbb{I} \}_{q\in P\setminus \{p\}}) \) and \( u_p(F^p, \{ F^q \}_{q\in P\setminus \{p\}}) \) we have that

\[
u^*_{p}(\mathbb{I}, \{ \mathbb{I} \}_{q\in P\setminus \{p\}}) \geq u_p(F^p, \{ F^q \}_{q\in P\setminus \{p\}}).
\]

For the latter, we have

\[
|u^*_{p}(\mathbb{I}, \{ \mathbb{I} \}_{q\neq p}) - u_p(\mathbb{I}, \{ \mathbb{I} \}_{q\neq p})| = \left| \frac{1}{\ell - 1} \sum_{q \in P\setminus \{p\}} \sum_{i,j} \Delta_{p,q}(i,j) (\Sign(\Delta_{p,q})_{i,j} - \Sign(\Delta_{G(p),G(q)})_{i,j}) \right|
\]

\[
\leq \frac{1}{\ell - 1} \sum_{q \in P\setminus \{p\}} \sum_{i,j} |\Delta_{p,q}(i,j) \Sign(\Delta_{p,q})_{i,j} - \Sign(\Delta_{G(p),G(q)})_{i,j}| \]

\[
\leq \frac{1}{\ell - 1} \sum_{q \in P\setminus \{p\}} \sum_{i,j} |\Delta_{p,q}(i,j) - \Delta_{G(p),G(q)}(i,j)| \]

\[
= \frac{1}{\ell - 1} \sum_{q \in P\setminus \{p\}} \|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \]

\[
\leq \frac{1}{\ell - 1} \sum_{q \in P\setminus \{p\}} \|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 + \|\Delta_{G(p),G(q)} - \Delta_{G(p),G(q)}\|_1 \]

\[
\leq \frac{1}{\ell - 1} \sum_{q \in P\setminus \{p\}} \varepsilon_1 + \varepsilon_2 = \varepsilon_1 + \varepsilon_2.
\]

To show that the third transition holds, we show that \( |a \cdot (\Sign(a) - \Sign(b))| \leq |a - b| \) for all real numbers \( a, b \in \mathbb{R} \). When \( \Sign(a) = \Sign(b) \), this holds trivially. When \( \Sign(a) \neq \Sign(b) \), note that the RHS becomes \( |a| + |b| \), which is an upper bound on the LHS, which becomes \( |a| \). The penultimate transition holds by \( \varepsilon_1 \)-accurate clustering and \( \varepsilon_2 \)-accurate estimates of cluster Delta matrices. This proves the first part of the theorem.

Now, we prove the second part of the theorem. For an uninformed strategy \( F^p \) such that all
the rows of $F^p$ are the same, i.e. $F^p_i = \psi$ for all $i$ where $\psi$ is a probability distribution, we have

\[
    u_p(F^p, \{F^q\}_{q \neq p}) = \frac{1}{\ell - 1} \sum_{q \neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q)
\]

\[
    = \frac{1}{\ell - 1} \sum_{q \neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p r_q} \psi_{r_p} F^q_{jr_q} S_{p,q}(r_p, r_q)
\]

\[
    = \frac{1}{\ell - 1} \sum_{q \neq p} \sum_{i,j} \psi_{r_p} F^q_{jr_q} S_{p,q}(r_p, r_q) \left( \sum_{i} \Delta_{p,q}(i,j) \right) = 0,
\]

where the last equality follows because the rows and columns of $\Delta_{p,q}$ sum to zero. Since $|u^*_p(\{I\}, \{I\}_{q \neq p}) - u_p(\{I\}, \{I\}_{q \neq p})| \leq \varepsilon_1 + \varepsilon_2$ we have

\[
u_p(\{I\}, \{I\}_{q \neq p}) \geq u^*_p(\{I\}, \{I\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2 > 0
\]
as $u^*_p(\{I\}, \{I\}_{q \neq p}) > \varepsilon_1 + \varepsilon_2$ for any $p$. \hfill \Box

The CAHU mechanism always ensures that there is no strategy profile which gives an expected utility more than $\varepsilon_1 + \varepsilon_2$ above truthful reporting. The condition $\min_p u^*_p(\{I\}, \{I\}_{q \neq p}) > \varepsilon_1 + \varepsilon_2$ is required to ensure that any uninformed strategy gives strictly less than the truth-telling equilibrium. This is important to promote effort in collecting and reporting an informative signal. Note that, the learning error $\varepsilon_2$ can be made if we have sufficient amount of data. Therefore, we need to guarantee that $\min_p u^*_p(\{I\}, \{I\}_{q \neq p}) > \varepsilon_1$ to ensure that any uninformed strategy gives strictly less than the truth-telling. Writing it out, this condition requires that for each agent $p$ the following holds:

\[
    \frac{1}{\ell - 1} \sum_{q \neq p, i,j; \Delta_{p,q}(i,j) > 0} \Delta_{p,q}(i,j) > \varepsilon_1. \tag{1.5}
\]

In particular, a sufficient condition for this property is that for every pair of agents the expected reward on a bonus task in the CA mechanism when making truthful reports is at least $\varepsilon_1$, i.e. for every pair of agents $p$ and $q$,

\[
    \sum_{i,j; \Delta_{p,q}(i,j) > 0} \Delta_{p,q}(i,j) > \varepsilon_1. \tag{1.6}
\]

In turn, as pointed out by Shnayder et al. [Shn+16], the LHS in (1.6) quantity can be interpreted as a measure of how much positive correlation there is in the joint distribution on
signals between a pair of agents. Note that it is not important that this is same-signal correlation.
For example, this quantity would be large between an accurate and an always-wrong agent in a binary-signal domain, since the positive correlation would be between one agent’s report and the flipped report from the other agent.

The incentive properties of the mechanism are retained when used together with learning the cluster structure and cluster Delta matrices. However, we do assume that the agents do not reveal their task assignments to each other. If the agents were aware of the identities of the tasks they are assigned, they could coordinate on the task identifiers to arrive at a profitable coordinated strategy. This is reasonable in practical settings as the number of tasks is often large. The next theorem shows that even if the agents could set the scoring matrices to be an arbitrary function \( \hat{S} \) through any possible deviating strategies, it is still beneficial to use the scoring matrices estimated from the truthful strategies. Let \( \hat{S} \) be an arbitrary scoring function i.e. \( \hat{S}_{p,q} \) specifies the score matrix for two agents from \( p \) and \( q \). We will write \( \hat{u}_p(F_p, \{F_q\}_{q \neq p}) \) to denote the expected utility of agent \( p \) under the CAHU mechanism with the reward function \( \hat{S} \) and strategy profile \( (F_p, \{F_q\}_{q \neq p}) \).

**Theorem 1.2.** Let \( \{\hat{S}_{p,q}\}_{p,q \in P} \) be an arbitrary set of scoring matrices where \( \hat{S}_{p,q} \in \{0,1\}^{n \times n} \) denotes the score matrix for agent \( p \) and agent \( q \). Then for every profile \( \{F_q\}_{q \in P} \) and agent \( p \in P \), we have

1. \( u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geq \hat{u}_p(F_p, \{F_q\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2 \).

2. If \( \min_{p} u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) > \varepsilon_1 \), then for any uninformed strategy \( F_0 \), \( u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) > \hat{u}_p(F_0, \{F_q\}_{q \neq p}) \).

**Proof.** Similar to the proof of Lemma 1.3, the utility of truthful reporting when the scoring matrix \( S_{p,q}^*(i,j) = \text{Sign}(\Delta_{p,q}(i,j)) \), is given by

\[
\hat{u}_p^*(\mathbb{I}, \{\mathbb{I}\}_{q \in P \setminus \{p\}}) = \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j; \Delta_{p,q}(i,j) > 0} \Delta_{p,q}(i,j)
\]

The utility \( \hat{u}_p(F_p, \{F_q\}_{q \in P \setminus \{p\}}) \) of an agent \( p \) for any strategy profile \( \{F_p, \{F_q\}_{q \in P \setminus \{p\}} \} \) when the scoring matrix is \( \hat{S}_{p,q} \), is given by

\[
\hat{u}_p(F_p, \{F_q\}_{q \in P \setminus \{p\}}) = \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p,r_q} F^p_{i r_p} F^q_{j r_q} \hat{S}_{p,q}(r_p, r_q)
\]
Now, using Lemma 1.2 and the expressions for $u_p^a(I, \{I\}_{q \neq p})$ and $\hat{u}_p(F^p, \{F^q\}_{q \neq p})$ we have that

$$u_p^a(I, \{I\}_{q \neq p}) \geq \hat{u}_p(F^p, \{F^q\}_{q \neq p}) \ .$$

Now the proof of Theorem 1.1 shows that $u_p(I, \{I\}_{q \neq p}) \geq u_p^a(I, \{I\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2$. Using the result above we get $u_p(I, \{I\}_{q \neq p}) \geq \hat{u}_p(I, \{I\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2$. Similar to the proof of Theorem 1.1 it can be shown that $\hat{u}_p(F^p_0, \{F^q\}_{q \neq p}) = 0$ for any uninformed strategy $F^p_0$. The proof of Theorem 1.1 also shows that $u_p(I, \{I\}_{q \neq p})$ can be made positive whenever $\min_p u_p^a(I, \{I\}_{q \neq p}) > \varepsilon_1$.

The above theorem implies that the incentive properties of our mechanism hold even when agents are allowed to coordinate their strategies and the mechanism is learned using reports from these coordinated strategies. To be precise, recall that $u_p(I, \{I\}_{q \neq p})$ is the expected payment to agent $p$ when the mechanism learns the true Delta matrix and the agent reports truthfully. This is no less than the expected payment minus $\varepsilon_1 + \varepsilon_2$ when the mechanism learns any other delta matrices and the agents misreport in any arbitrary way.

1.4 Learning the Agent Signal Types

In this section, we provide algorithms for learning a clustering of agent signal types from reports, and further, for learning the cluster pairwise $\Delta$ matrices. The estimates of the $\Delta$ matrices can then be used to give an approximate-informed truthful mechanism. Along the way, we couple our methods with the latent “confusion matrix” methods of Dawid and Skene [DS79].

Recall that $m$ is the total number of tasks about which reports are collected. Reports on $m_1$ of these tasks will also be used for clustering, and reports on a further $m_2$ of these tasks will be used for learning the cluster pairwise $\Delta$ matrices. We consider two different schemes for assigning agents to tasks for the purpose of clustering and learning (see Figures 1.1 and 1.2):

1. **Fixed Task Assignment**: Each agent is assigned to the same, random subset of tasks of size $m_1 + m_2$ of the given $m$ tasks.
2. **Uniform Task Assignment:** For clustering, we select two agents $r_1$ and $r_2$, uniformly at random, to be reference agents. These agents are assigned to a subset of tasks of size $m_1(< m)$. For all other agents, we then assign a required number of tasks, $s_1$, uniformly at random from the set of $m_1$ tasks. For learning the cluster pairwise $\Delta$-matrices, we also assign one agent from each cluster to some subset of tasks of size $s_2$, selected uniformly at random from a second set of $m_2(< m - m_1)$ tasks.

For each assignment scheme, the analysis establishes that there are enough agents who have done a sufficient number of joint tasks. Table 1.1 summarizes the sample complexity results, stating them under two different assumptions about the way in which signals are generated.

<table>
<thead>
<tr>
<th></th>
<th>No Assumption</th>
<th>Dawid-Skene</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Assignment</td>
<td>Clustering: $\tilde{O} \left( \frac{\ell n^2}{\gamma} \right)$</td>
<td>Clustering: $\tilde{O} \left( \frac{\ell n^2}{\gamma} \right)$</td>
</tr>
<tr>
<td></td>
<td>Learning: $\tilde{O} \left( \frac{Kn^2}{\gamma^2} \right)$</td>
<td>Learning: $\tilde{O} \left( \frac{Kn^2}{\gamma^2} \right)$</td>
</tr>
<tr>
<td>Uniform Assignment</td>
<td>Clustering: $\tilde{O} \left( \frac{\ell n^{2} + m_1}{\gamma} \right)$</td>
<td>Clustering: $\tilde{O} \left( \frac{\ell n^{2} + m_1}{\gamma} \right)$</td>
</tr>
<tr>
<td></td>
<td>Learning: $\tilde{O} \left( Km_2^{7/8} \sqrt{\frac{n^2}{(\varepsilon')^2}} \right)$</td>
<td>Learning: $\tilde{O} \left( Km_2^{7/8} \sqrt{\frac{n^2}{(\varepsilon')^2}} \right)$</td>
</tr>
</tbody>
</table>

*Table 1.1: Sample complexity for the CAHU mechanism. The rows indicate the assignment scheme and the columns indicate the modeling assumption. Here $\ell$ is the number of agents, $n$ is the number of signals, $\varepsilon'$ is a parameter that controls learning accuracy $^\dagger$, $\gamma$ is a clustering parameter, $K$ is the number of clusters, and $m_1$ (resp. $m_2$) is the size of the set of tasks from which the tasks used for clustering (resp. learning) are sampled.*

$^\dagger$For an arbitrary $m_2$, this bound is $Km_2$ as long as $m_2$ is $\Omega \left( n^7/(\varepsilon')^2 \right)$

$^\ddagger$In the no assumption approach (resp. Dawid-Skene Model), $\varepsilon'$ is the error in the estimation of the joint...
1.4.1 Clustering

We proceed by presenting and analyzing a simple clustering algorithm.

**Definition 1.4.** A clustering $G_1, \ldots, G_K$ is $\gamma$-good if for some $\gamma > 0$

\[
G(q) = G(r) \Rightarrow \| \Delta_{pq} - \Delta_{pr} \|_1 \leq \varepsilon - 4\gamma \quad \forall p \in [\ell] \setminus \{q, r\} \tag{1.7}
\]

\[
G(q) \neq G(r) \Rightarrow \| \Delta_{pq} - \Delta_{pr} \|_1 > \varepsilon \quad \forall p \in [\ell] \setminus \{q, r\} \tag{1.8}
\]

We first show that an $\gamma$-good clustering, if exists, must be unique.

**Theorem 1.3.** Suppose there exist two clustering $\{G_j\}_{j \in [K]}$ and $\{T_i\}_{i \in [K']}$ that are $\varepsilon$-good. Then $K' = K$ and $G_j = T_{\pi(j)}$ for some permutation $\pi$ over $[K]$.

**Proof.** Suppose equations 1.7 and 1.8 hold with parameters $\gamma_1$ and $\gamma_2$ respectively for the clusterings $\{G_j\}_{j \in [K]}$ and $\{T_i\}_{i \in [K']}$. If possible, assume there exist $T_i$ and $G_j$ such that $T_i \cap G_j \neq \emptyset$, $G_j \setminus T_i \neq \emptyset$ and $T_i \cap G_j \neq \emptyset$. Pick $s \in T_i \cap G_j$ and $r \in G_j \setminus T_i$. Then we must have, for any $p \notin \{q, s, r\}$,

1. $\| \Delta_{pr} - \Delta_{ps} \|_1 > \varepsilon$ (inter-cluster distance in $\{T_i\}_{i \in [K']}$)

2. $\| \Delta_{pr} - \Delta_{ps} \|_1 \leq \varepsilon - 4\gamma_1$ (intra-cluster distance in $\{G_j\}_{j \in [K]}$)

This is a contradiction. Now suppose $K' > K$. Then there must exist $T_i$ and $T_k$ such that $T_i \cup T_k \subseteq G_j$ for some $j$. Pick $q \in T_i$ and $r \in T_k$. Then, for any $p \notin \{q, r\}$

1. $\| \Delta_{pq} - \Delta_{pr} \|_1 > \varepsilon$ (inter-cluster distance in $\{T_i\}_{i \in [K']}$)

2. $\| \Delta_{pq} - \Delta_{pr} \|_1 \leq \varepsilon - 4\gamma_1$ (intra-cluster distance in $\{G_j\}_{j \in [K]}$)

This leads to a contradiction and proves that $K' \leq K$. Similarly we can prove $K \leq K'$. Therefore, we have shown that for each each $G_j$ there exists $i$ such that $G_j = T_i$. \qed

Since there is a unique $\varepsilon$-good clustering (up to a permutation), we will refer to this clustering as the **correct clustering**. The assumption that there exists an $\varepsilon$-good clustering is stronger than Equation (1.3) introduced earlier. In particular, identifying the correct clustering probability distribution (resp. aggregate confusion matrix).
Algorithm 1.2: Clustering

Input: $\varepsilon, \gamma$ such that there exists an $\varepsilon$-good clustering with parameter $\gamma$

Output: A clustering $\{\hat{G}_t\}_{t=1}^K$

1. $\hat{G} \leftarrow \emptyset$, $\hat{K} \leftarrow 0$  // $\hat{G}$ is the list of clusters, $\hat{K} = |\hat{G}|$
2. Make a new cluster $\hat{G}_1$ and add agent 1 Add $\hat{G}_1$ to $\hat{G}$, $\hat{K} \leftarrow \hat{K} + 1$
3. for $t \in [\hat{K}]$ do
4.  Pick an arbitrary agent $q_t \in \hat{G}_t$
5.  Pick $p_t \in [\ell] \backslash \{i, q_t\}$ (Fixed) or $p_t \in \{r_1, r_2\} \backslash \{i, q_t\}$ (Uniform), such that $p_t$ has at least $\Omega\left(\frac{n^2 \log(\ell/\delta)}{\gamma^2}\right)$ tasks in common with both $q_t$ and $i$
6.  Let $\tilde{\Delta}_{p_t, q_t}$ be the empirical Delta matrix from reports of agents $p_t$ and $q_t$
7.  Let $\tilde{\Delta}_{p_t, i}$ be the empirical Delta matrix from reports of agents $p_t$ and $i$
8. if $\exists t \in [\hat{K}] : \|\tilde{\Delta}_{p_t, q_t} - \tilde{\Delta}_{p_t, i}\|_1 \leq \varepsilon - 2\gamma$ then
9.  Add $i$ to $\hat{G}_t$ (with ties broken arbitrarily for $t$)
10. else
11.  Make a new cluster $\hat{G}_{\hat{K}+1}$ and add agent $i$ to it
12.  Add $\hat{G}_{\hat{K}+1}$ to $\hat{G}$, $\hat{K} \leftarrow \hat{K} + 1$

Figure 1.3: Algorithm 1.2 checks whether $i$ and $q_t$ are in the same cluster by estimating $\Delta_{p_t, q_t}$ and $\Delta_{p_t, i}$.

needs to satisfy Equation (1.8), i.e. the $\Delta$-matrices of two agents belonging to two different clusters are different with respect to every other agent. So, we need low inter-cluster similarities in addition to high intra-cluster similarities. The pseudo-code for the clustering algorithm is presented in Algorithm 1.2. This algorithm iterates over the agents, and forms clusters in a greedy manner. First, we prove that as long as we can find an agent $p_t$ that has $\Omega\left(\frac{n^2 \log(\ell/\delta)}{\gamma^2}\right)$ tasks in common with both $q_t$ and $i$, then the clustering produced by Algorithm 1.2 is correct with probability at least $1 - \delta$.

**Theorem 1.4.** If for all $i \in P$ and $q_t \in G(i)$, there exists $p_t$ which has $\Omega\left(\frac{n^2 \log(\ell/\delta)}{\gamma^2}\right)$ tasks in common with both $q_t$ and $i$, then Algorithm 1.2 recovers the correct clustering i.e. $\hat{G}_t = G_t$ for $t = 1, \ldots, K$ with
probability at least $1 - \delta$.

We need two key technical lemmas to prove Theorem 1.4. The first lemma shows that in order to estimate $\Delta_{p,q}$ with an L1 distance of at most $\gamma$, it is sufficient to estimate the joint probability distribution $D_{p,q}$ with an L1 distance of at most $\gamma/3$. With this, we can estimate the delta matrices of agent pairs from the joint empirical distributions of their reports.

**Lemma 1.4.** For all $p, q \in P$, $\|\tilde{D}_{p,q} - D_{p,q}\|_1 \leq \gamma/3 \Rightarrow \|\tilde{\Delta}_{p,q} - \Delta_{p,q}\|_1 \leq \gamma$.

Proof.

\[
\|\tilde{\Delta}_{p,q} - \Delta_{p,q}\|_1 = \sum_{i,j} |\tilde{D}_{p,q}(i,j) - \tilde{D}_p(i)\tilde{D}_q(j) - (D_{p,q}(i,j) - D_p(i)D_q(j))| \\
= \sum_{i,j} |\tilde{D}_{p,q}(i,j) - D_{p,q}(i,j)| + \sum_{i,j} |\tilde{D}_p(i)\tilde{D}_q(j) - \tilde{D}_p(i)D_q(j) + \tilde{D}_p(i)D_q(j) - D_p(i)D_q(j)| \\
\leq \gamma/3 + \sum_{i,j} |\tilde{D}_p(i) - D_p(i)| + \sum_{i,j} |\tilde{D}_q(j) - D_q(j)| \\
\leq \gamma/3 + \sum_{i,j} |\tilde{D}_{p,q}(i,j) - D_{p,q}(i,j)| + \sum_{i,j} |\tilde{D}_{p,q}(i,j) - D_{p,q}(i,j)| \\
\leq \gamma,
\]

as required. \qed

The second lemma is about learning the empirical distributions of reports of pairs of agents. This can be proved using Theorems 3.1 and 2.2 from the work of Devroye and Lugosi [DL12].

**Lemma 1.5.** Any distribution over a finite domain $\Omega$ is learnable within a L1 distance of $d$ with probability at least $1 - \delta$, by observing $O\left(\frac{|\Omega|}{d^2} \log(1/\delta)\right)$ samples from the distribution.

We can use the above lemma to show that the joint distributions of reports of agents can be learned to within an L1 distance $\gamma$ with probability at least $1 - \delta/K\ell$, by observing $O\left(\frac{n^2}{\tau} \log(K\ell/\delta)\right)$ reports on joint tasks.

**Corollary 1.5.** For any agent pair $p, q \in P$, the joint distribution of their reports $D_{p,q}$ is learnable within a L1 distance of $\gamma$ using $O\left(\frac{n^2}{\tau} \log(K\ell/\delta)\right)$ reports on joint tasks with probability at least $1 - \delta/K\ell$. 

33
We are now ready to prove Theorem 1.4.

**Proof of Theorem 1.4.** The proof is by induction on the number of agents $\ell$. Suppose all the agents up to and including $i - 1$ have been clustered correctly. Consider the $i$-th agent and suppose it belongs to the cluster $G_t$. Suppose $\hat{G}_t \neq \emptyset$. Then using the triangle inequality we have

$$\|\bar{\Delta}_{p_{i,q_t}} - \bar{\Delta}_{p_{i,i}}\|_1 \leq \|\bar{\Delta}_{p_{i,q_t}} - \Delta_{p_{i,q_t}}\|_1 + \|\Delta_{p_{i,q_t}} - \Delta_{p_{i,i}}\|_1 + \|\bar{\Delta}_{p_{i,i}} - \Delta_{p_{i,i}}\|_1$$

Since $q_t \in G_t$, we have $\|\Delta_{p_{i,q_t}} - \Delta_{p_{i,i}}\|_1 \leq \varepsilon/2 - 4\gamma$. Moreover, using lemma 1.4 and corollary 1.5 we have that, with probability at least $1 - \delta/K\ell$, $\|\bar{\Delta}_{p_{i,q_t}} - \Delta_{p_{i,q_t}}\|_1 \leq \gamma$ and $\|\bar{\Delta}_{p_{i,i}} - \Delta_{p_{i,i}}\|_1 \leq \gamma$. This ensures that $\|\bar{\Delta}_{p_{i,q_t}} - \bar{\Delta}_{p_{i,i}}\|_1 \leq \varepsilon/2 - 2\gamma$. On the other hand pick any cluster $G_s$ such that $s \neq t$ and $\hat{G}_s \neq \emptyset$. Then

$$\|\bar{\Delta}_{p_{i,q_t}} - \bar{\Delta}_{p_{i,i}}\|_1 \geq \|\Delta_{p_{i,q_t}} - \Delta_{p_{i,i}}\|_1 - \|\bar{\Delta}_{p_{i,q_t}} - \Delta_{p_{i,q_t}}\|_1 - \|\bar{\Delta}_{p_{i,i}} - \Delta_{p_{i,i}}\|_1$$

Since $i \notin G_s$ we have $\|\Delta_{p_{i,q_t}} - \Delta_{p_{i,i}}\|_1 > \varepsilon/2$. Again, with probability at least $1 - \delta/K\ell$, we have $\|\bar{\Delta}_{p_{i,q_t}} - \Delta_{p_{i,q_t}}\|_1 \leq \gamma$ and $\|\bar{\Delta}_{p_{i,i}} - \Delta_{p_{i,i}}\|_1 \leq \gamma$. This ensures that $\|\bar{\Delta}_{p_{i,q_t}} - \bar{\Delta}_{p_{i,i}}\|_1 > \varepsilon/2 - 2\gamma$. This ensures that condition on line (8) is violated for all clusters $s \neq t$. If $\hat{G}_t \neq \emptyset$ this condition is satisfied and agent $i$ added to cluster $\hat{G}_t$, otherwise the algorithm makes a new cluster with agent $i$. Now note that the algorithm makes a new cluster only when it sees an agent belonging to a new cluster. This implies that $\hat{K} = K$. Taking a union bound over the $K$ choices of $q_t$ for the $K$ clusters, we see that agent $i$ is assigned to its correct cluster with probability at least $1 - \delta/\ell$. Finally, taking a union bound over all the $\ell$ agents we get the desired result.

Next we show how the assumption in regard to task overlap is satisfied under each assignment scheme, and characterize the sample complexity of learning the clusterings under each scheme. In the fixed assignment scheme, all the agents are assigned to the same set of $m_1 = \Omega\left(\frac{n^2}{\ell^2} \log(K\ell/\delta)\right)$ tasks. Thus, for each agent pair $q_t$ and $i$, any other agent in the population can act as $p_t$. The total number of tasks performed is $O\left(\frac{tn^2}{\ell^2} \log(K\ell/\delta)\right)$.

In the uniform assignment scheme, we select two agents $r_1$ and $r_2$ uniformly at random to be reference agents, and assign these agents to each of $m_1 = \Omega\left(\frac{n^2}{\ell^2} \log(K\ell/\delta)\right)$ tasks. For all
other agents we then assign \( s_1 = \Omega\left(\frac{m^2}{\gamma} \log(K\ell/\delta)\right) \) tasks uniformly at random from this set of \( m_1 \) tasks. If \( m_1 = s_1 \), then the uniform task assignment is the same as fixed task assignment. However, in applications (e.g., [KOS11]), where one wants the task assignments to be more uniform across tasks, it will make sense to use a larger value of \( m_1 \). The reference agent \( r_1 \) can act as \( p_t \) for all agent pairs \( q_t \) and \( i \) other than \( r_1 \). Similarly, reference \( r_2 \) can act as \( p_t \) for all agent pairs \( q_t \) and \( i \) other than \( r_2 \). If \( q_t = r_1 \) and \( i = r_2 \) or \( q_t = r_2 \) and \( i = r_1 \), then any other agent can act as \( p_t \). The total number of tasks performed is \( W_p n_2 \log p K \{ d \} \), which is sufficient for the high probability result.

### 1.4.2 Learning the Cluster Pairwise \( \Delta \) Matrices

We proceed now under the assumption that the agents are clustered into \( K \) groups, \( G_1, \ldots, G_K \). Our goal is to estimate the cluster-pairwise delta matrices \( \Delta_{G_s,G_t} \) as required by Algorithm 1.1. We estimate the \( \Delta_{G_s,G_t} \) under two different settings: when we have no model of the signal distribution, and in the Dawid-Skene latent attribute model.

<table>
<thead>
<tr>
<th>Algorithm 1.3: Learning-( \Delta )-No-Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong> for ( t = 1, \ldots, K ) do</td>
</tr>
<tr>
<td><strong>2</strong> Chose agent ( q_t \in G_t ) arbitrarily</td>
</tr>
<tr>
<td><strong>3</strong> for each pair of clusters ( G_s,G_t ) do</td>
</tr>
<tr>
<td><strong>4</strong> Let ( q_s ) and ( q_t ) be the chosen agents for ( G_s ) and ( G_t ), respectively.</td>
</tr>
<tr>
<td><strong>5</strong> Let ( \hat{D}<em>{q_s,q_t} ) be the empirical estimate of ( D</em>{q_s,q_t} ) such that ( | \hat{D}<em>{q_s,q_t} - D</em>{q_s,q_t} |_1 \leq \epsilon' ) with probability at least ( 1 - \delta/K^2 )</td>
</tr>
<tr>
<td><strong>6</strong> Let ( \tilde{\Delta}<em>{q_s,q_t} ) be the empirical Delta matrix computed using ( \hat{D}</em>{q_s,q_t} )</td>
</tr>
<tr>
<td><strong>7</strong> Set ( \Delta_{G_s,G_t} = \tilde{\Delta}_{q_s,q_t} )</td>
</tr>
</tbody>
</table>

### Learning the \( \Delta \)-Matrices with No Assumption

We first characterize the sample complexity of learning the \( \Delta \)-matrices in the absence of any modeling assumptions. In order to estimate \( \tilde{\Delta}_{G_s,G_t} \), Algorithm 1.3 first picks agent \( q_s \) from cluster \( G_s \), estimates \( \tilde{\Delta}_{q_s,q_t} \) and use this estimate in place of \( \tilde{\Delta}_{G_s,G_t} \). For the fixed assignment scheme, we assign the agents \( q_s \) to the same set of tasks of size \( O\left(\frac{n^2}{(\epsilon')^2} \log(K/\delta)\right) \). For the uniform assignment scheme, we assign the agents to subsets of tasks of an appropriate size.
among the pool of \(m_2\) tasks.

**Theorem 1.6.** Given an \(\varepsilon\)-good clustering \(\{G_s\}_{s=1}^K\), if the number of shared tasks between any pair of agents \(q_s, q_t\) is \(O\left(\frac{n^2}{m^2} \log(K/\delta)\right)\), then Algorithm 1.3 guarantees that for all \(s, t\), \(\|\bar{\Delta}_{G_s, G_t} - \Delta_{G_s, G_t}\|_1 \leq 3\varepsilon' + 2\varepsilon\) with probability at least \(1 - \delta\). The total number of samples collected by the algorithm is \(O\left(\frac{Km^2}{(c')^2} \log(K/\delta)\right)\) (resp. \(O\left(\frac{Km^{2}_{\ast}}{(c')^2} \log(K/\delta)\right)\) w.h.p.) under the fixed (resp. uniform) assignment scheme.

We first prove a sequence of lemmas that will be used to prove the result. Appendix A.1 contains their proofs.

**Lemma 1.6.** For every pair of agents \(p, q\), we have

\[
\|\Delta_{p, q} - \Delta_{G(p), G(q)}\|_1 \leq 2 \cdot \max_{a, b, c \in P: G(a) = G(b)} \|\Delta_{a, c} - \Delta_{b, c}\|_1.
\]

The next lemma characterizes the error made by Algorithm 1.3 in estimating the \(\Delta_{G_s, G_t}\) matrices.

**Lemma 1.7.** For any two agents \(p \in G_s\) and \(q \in G_t\), \(\|\bar{D}_{p, q} - D_{p, q}\|_1 \leq \varepsilon' \Rightarrow \|\bar{\Delta}_{p, q} - \Delta_{G_s, G_t}\|_1 \leq 3\varepsilon' + 2\varepsilon\).

**Proof.** (Theorem 1.6) By Lemma 1.5, to estimate \(D_{p, q}\) within a distance of \(\varepsilon'\) with probability at least \(1 - \delta/K^2\), we need \(O\left(\frac{n^2}{(c')^2} \log(K^2/\delta)\right)\). By a union bound over the \(K^2\) pairs of clusters we see that with probability at least \(1 - \delta\), we have \(\|\bar{D}_{q_s, q_t} - D_{q_s, q_t}\|_1 \leq \varepsilon'\). This proves the first part of the theorem. When the assignment scheme is fixed, we can assign all the same tasks to \(K\) agents \(\{q_t\}_{t=1}^K\), and hence the total number of samples is multiplied by \(K\).

On the other hand, under the uniform assignment scheme, suppose each agent \(\{q_t\}_{t=1}^K\) is assigned to a subset of \(s_2\) tasks selected uniformly at random from the pool of \(m_2\) tasks. Now consider any two agents \(q_s\) and \(q_t\). Let \(X_i\) be an indicator random variable which is 1 when \(i \in [m_2]\) is included in tasks of \(q_s\), and 0 otherwise. Also, let \(Y_i\) be a similar random variable for the tasks of \(q_t\). Let \(Z_i = X_i \times Y_i\). The probability that both agents are assigned to a particular task \(i\), \(\Pr(Z_i = 1) = (s_2/m_2)^2\). Therefore, the expected number of overlapping tasks among the two agents is \(m_2 \cdot \left(\frac{s_2}{m_2}\right)^2 = \frac{s_2^2}{m_2}\), i.e. \(E[\sum_i Z_i] = \frac{s_2^2}{m_2}\). Now, we want to bound the deviations
from this expectations. Let \( R_j = E \left[ \sum_{i=1}^{m_2} Z_i \bigg| X_1, \ldots, X_j, Y_1, \ldots, Y_j \right] \), then \( R_j \) is a Doob martingale sequence for \( \sum_{i=1}^{j} Z_i \). Also, it is easy to see that this martingale sequence is bounded by 1, i.e. \( |R_{j+1} - R_j| \leq 1 \). Therefore, we apply the Azuma-Hoeffding bound (Lemma 1.8) as

\[
\Pr \left[ \left| \sum_i Z_i \right| > \frac{s_2^2}{2m_2} \right] \leq 2 \exp \left\{ -\frac{s_2^4}{8m_2^2} \right\}.
\]

Now substituting \( s_2 = m_2^{7/8} \cdot L^{1/2} \) where \( L = O \left( \frac{n^2}{\log(K^2/\delta)} \right) \), we get

\[
\Pr \left[ \sum_i Z_i < m_2^{3/4} L/2 \right] \leq 2 \exp \left\{ -\sqrt{m_2} L^2 \right\}.
\]

Taking a union bound over \( K^2 \) pairs of agents, if each agent completes \( m_2^{7/8} \cdot L^{1/2} \) tasks selected uniformly at random from the pool of \( m_2 \) tasks, then the probability that any pair of agents has number of shared tasks \( L \) is at least \( 1 - K^2 \exp\{-\sqrt{m_2} L^2\} \), which is exponentially small in \( m_2 \).

\[\square\]

**Lemma 1.8.** Suppose \( X_n, n \geq 1 \) is a martingale such that \( X_0 = 0 \) and \( |X_i - X_{i-1}| \leq 1 \) for each \( 1 \leq i \leq n \). Then for every \( t > 0 \)

\[
\Pr \left[ |X_n| > t \right] \leq 2 \exp \left\{ -t^2/2n \right\}
\]

**Learning the \( \Delta \)-matrices Under the Dawid-Skene Model**

In this section, we assume that the agents receive signals according to the Dawid and Skene [DS79] model. Here, each task has a latent attribute and each agent has a confusion matrix to parameterize its signal distribution conditioned on this latent value. Recall two notations from the introduction: \( D_p(i) \) is the marginal probability of observing signal \( i \) for agent \( p \) and \( D_{p,q}(i,j) \) is the joint probability that the agents \( p \) and \( q \) observe signals \( i \) and \( j \) respectively. Then the Dawid-Skene Model is formally defined as:

- Let \( \{\pi_k\}_{k=1}^n \) denote the prior probability over \( n \) latent values.
- Agent \( p \) has confusion matrix \( C^p \in \mathbb{R}^{n \times n} \), such that \( C^p_{ij} = D_p(S_p = j | T = i) \) where \( T \) is the
latent value. Given this, the joint signal distribution for a pair of agents \( p \) and \( q \) is

\[
D_{p,q}(S_p = i, S_q = j) = \sum_{k=1}^{n} \pi_k C_{ki}^p C_{kj}^q,
\]

and the marginal signal distribution for agent \( p \) is

\[
D_p(S_p = i) = \sum_{k=1}^{n} \pi_k C_{ki}^p.
\]

For cluster \( G_t \), we write \( C_t = \frac{1}{|G_t|} \sum_{p \in G_t} C^p \) to denote the aggregate confusion matrix of \( G_t \). As before, we assume that we are given an \( \varepsilon \)-good clustering, \( G_1, \ldots, G_K \), of the agents. Our goal is to provide an estimate of the \( \Delta_{G_a,G_b} \)-matrices.

Lemma 1.9 proves that in order to estimate \( \Delta_{G_a,G_b} \) within an L1 distance of \( \varepsilon' \), it is enough to estimate the aggregate confusion matrices within an L1 distance of \( \frac{\varepsilon'}{4} \). So in order to learn the pairwise delta matrices between clusters, we first ensure that for each cluster \( G_t \), we have \( \| \tilde{C}^t - C^t \|_1 \leq \varepsilon'/4 \) with probability at least \( 1 - \delta/K \), and then use the following formula to compute the delta matrices:

\[
\Delta_{G_a,G_b}(i,j) = \sum_{k=1}^{n} \pi_k \tilde{C}_{ki}^t \tilde{C}_{kj}^t - \sum_{k=1}^{n} \pi_k C_{ki}^s \sum_{k=1}^{n} \pi_k C_{kj}^s
\]

Lemma 1.9. For all \( G_a, G_b \), \( \| \tilde{C}^a - C^a \|_1 \leq \varepsilon'/4 \) and \( \| \tilde{C}^b - C^b \|_1 \leq \varepsilon'/4 \) \( \Rightarrow \| \Delta_{G_a,G_b} \| \leq \varepsilon' \).

We now turn to the estimation of the aggregate confusion matrix of each cluster. Let us assume for now that the agents are assigned to the tasks according to the uniform assignment scheme, i.e. agent \( p \) belonging to cluster \( G_a \) is assigned to a subset of \( B_a \) tasks selected uniformly at random from a pool of \( m_2 \) tasks. For cluster \( G_a \), we choose \( B_a = \frac{m_2}{|G_a|} \ln \left( \frac{m_2K}{\beta} \right) \). This implies:

1. For each \( j \in [m_2] \), \( \Pr[ \text{agent } p \in G_a \text{ completes task } j ] = \frac{\log(m_2K/\beta)}{|G_a|} \), i.e. each agent \( p \) in \( G_a \) is equally likely to complete every task \( j \).

2. \( \Pr[ \text{task } j \text{ is unlabeled by } G_a ] = \left( 1 - \frac{\log(m_2K/\beta)}{|G_a|} \right)^{|G_a|} \leq \frac{\beta}{m_2K} \). Taking a union bound over the \( m_2 \) tasks and \( K \) clusters, we get the probability that any task is unlabeled is at most \( \beta \). Now if we choose \( \beta = 1/\text{poly}(m_2) \), we observe that with probability at least \( 1 - 1/\text{poly}(m_2) \), each task \( j \) is labeled by some agent in each cluster when \( B_a = \tilde{O} \left( \frac{m_2}{|G_a|} \right) \).
All that is left to do is to provide an algorithm and sample complexity for learning the aggregate confusion matrices. For this, we will use \( n \) dimensional unit vectors to denote the reports of the agents (recall that there are \( n \) possible signals). In particular agent \( p \)'s report on task \( j \), \( r_{pj} \in \{0, 1\}^n \). If \( p \)'s report on task \( j \) is \( c \), then the \( c \)-th coordinate of \( r_{pj} \) is 1 and all the other coordinates are 0. The expected value of agent \( p \)'s report on the \( j \)th task is \( E[r_{pj}] = \sum_{k=1}^{n} \pi_k c^p_k \)

The aggregated report for a cluster \( G_t \) is given as \( R_{tj} = \frac{1}{|G_t|} \sum_{p \in G_t} r_{pj} \).

Suppose we want to estimate the aggregate confusion matrix \( C^1 \) of some cluster \( G_1 \). To do so, we first pick three clusters \( G_1, G_2 \) and \( G_3 \) and write down the corresponding cross moments.

Let \( (a, b, c) \) be a permutation of the set \( \{1, 2, 3\} \). We have:

\[
E[R_{aj}] = \sum_k \pi_k C^a_k
\]

\[
E[R_{aj} \otimes R_{bj}] = \sum_k \pi_k C^a_k \otimes C^b_k
\]

\[
E[R_{aj} \otimes R_{bj} \otimes R_{cj}] = \sum_k \pi_k C^a_k \otimes C^b_k \otimes C^c_k
\]

The cross moments are asymmetric, however using Theorem 3.6 in the work by Anandkumar et al. [Ana+14], we can write the cross-moments in a symmetric form.

**Lemma 1.10.** Assume that the vectors \( \{C^1_t, \ldots, C^n_t\} \) are linearly independent for each \( t \in \{1, 2, 3\} \). For any permutation \( (a, b, c) \) of the set \( \{1, 2, 3\} \) define

\[
R_{aj}' = E[R_{cj} \otimes R_{bj}] (E[R_{aj} \otimes R_{bj}])^{-1} R_{aj}
\]

\[
R_{bj}' = E[R_{cj} \otimes R_{aj}] (E[R_{bj} \otimes R_{aj}])^{-1} R_{bj}
\]

\[
M_2 = E[R_{aj}' \otimes R_{bj}'] \	ext{ and } M_3 = E[R_{aj}' \otimes R_{bj}' \otimes R_{cj}]
\]

Then \( M_2 = \sum_{k=1}^{n} \pi_k C^c_k \otimes C^c_k \) and \( M_3 = \sum_{k=1}^{n} \pi_k C^a_k \otimes C^b_k \otimes C^c_k \)

We cannot compute the moments exactly, but rather estimate the moments from samples observed from different tasks. Furthermore, for a given task \( j \), instead of exactly computing the aggregate label \( R_{gj} \), we select one agent \( p \) uniformly at random from \( G_g \) and use agent \( p \)'s
report on task \( j \) as a proxy for \( R_{gj} \). We will denote the corresponding report as \( \tilde{R}_{gj} \). The next lemma proves that the cross-moments of \( \{\tilde{R}_{gj}\}_{g=1}^{K} \) and \( \{R_{gj}\}_{g=1}^{K} \) are the same.

**Lemma 1.11.**

1. For any group \( G_a \), \( \mathbb{E} [\tilde{R}_{aj}] = \mathbb{E} [R_{aj}] \)

2. For any pair of groups \( G_a \) and \( G_b \), \( \mathbb{E} [\tilde{R}_{aj} \otimes \tilde{R}_{bj}] = \mathbb{E} [R_{aj} \otimes R_{bj}] \)

3. For any three groups \( G_a, G_b \) and \( G_c \), \( \mathbb{E} [\tilde{R}_{aj} \otimes \tilde{R}_{bj} \otimes \tilde{R}_{cj}] = \mathbb{E} [R_{aj} \otimes R_{bj} \otimes R_{cj}] \)

The next set of equations show how to approximate the moments \( M_2 \) and \( M_3 \):

\[
\hat{R}_{aj} = \left( \frac{1}{m_2} \sum_{j'=1}^{m_2} \tilde{R}_{aj'} \otimes \tilde{R}_{bj'} \right)^{\frac{1}{2}} \mathbb{E} [\tilde{R}_{aj}] 
\]

\[
\hat{R}_{bj} = \left( \frac{1}{m_2} \sum_{j'=1}^{m_2} \tilde{R}_{aj'} \otimes \tilde{R}_{bj'} \right)^{\frac{1}{2}} \mathbb{E} [\tilde{R}_{bj}] 
\]

\[
\hat{M}_2 = \frac{1}{m_2} \sum_{j'=1}^{m_2} \tilde{R}_{aj'} \otimes \tilde{R}_{bj'} \quad \text{and} \quad \hat{M}_3 = \frac{1}{m_2} \sum_{j'=1}^{m_2} \tilde{R}_{aj'} \otimes \tilde{R}_{bj'} \otimes \tilde{R}_{cj'}
\]

We use the tensor decomposition algorithm (1.4) on \( \hat{M}_2 \) and \( \hat{M}_3 \) to recover the aggregate confusion matrix \( \tilde{C}^c \) and \( \tilde{\Pi} \), where \( \tilde{\Pi} \) is a diagonal matrix whose \( k \)-th component is \( \tilde{\pi}_k \), an estimate of \( \pi_k \). In order to analyze the sample complexity of Algorithm 1.4, we need to make

**Algorithm 1.4: Estimating Aggregate Confusion Matrix**

- **Input:** \( K \) clusters of agents \( G_1, G_2, \ldots, G_K \) and the reports \( \tilde{R}_{gj} \in \{0, 1\}^n \) for \( j \in [m] \) and \( g \in [K] \)
- **Output:** Estimate of the aggregate confusion matrices \( \tilde{C}^c \) for all \( g \in [K] \)
- **1.** Partition the \( K \) clusters into groups of three
- **2.** For each group of three clusters \( \{g_a, g_b, g_c\} \) do
  - **3.** For \( (a, b, c) \in \{(g_b, g_c, g_a), (g_c, g_a, g_b), (g_a, g_b, g_c)\} \) do
    - **4.** Compute the second and the third order moments \( \hat{M}_2 \in \mathbb{R}^{n \times n} \), \( \hat{M}_3 \in \mathbb{R}^{n \times n \times n} \).
      - /* Compute \( \tilde{C}^g \) by tensor decomposition */
    - **5.** Compute whitening matrix \( \hat{Q} \in \mathbb{R}^{n \times n} \) such that \( \hat{Q}^T \hat{M}_2 \hat{Q} = I \)
    - **6.** Compute eigenvalue-eigenvector pairs \( (\hat{\lambda}_k, \hat{v}_k)_{k=1}^n \) of the whitened tensor \( \hat{M}_3(\hat{Q}, \hat{Q}, \hat{Q}) \)
    - **7.** by using the robust tensor power method
    - **8.** Compute \( \hat{\omega}_k = \hat{\lambda}_k^{-2} \) and \( \hat{\mu}_k = (\hat{Q}^T)^{-1} \hat{v}_k \)
    - **9.** For \( k = 1, \ldots, n \) set the \( k \)-th column of \( \tilde{C}^c \) by some \( \tilde{\mu}_k \) whose \( k \)-th coordinate has the greatest component, then set the \( k \)-th diagonal entry of \( \tilde{\Pi} \) by \( \tilde{\omega}_k \)

40
some mild assumptions about the problem instance. For any two clusters \( G_a \) and \( G_b \), define \( S_{ab} = \mathbb{E} \left[ R_{aj} \otimes R_{bj} \right] = \sum_{k=1}^{n} \pi_k C_k^a \otimes C_k^b \). We make the following assumptions:

1. There exists \( \sigma_L > 0 \) such that \( \sigma_n(S_{ab}) \geq \sigma_L \) for each pair of clusters \( a \) and \( b \), where \( \sigma_n(M) \) is the \( n \)-th smallest eigenvalue of \( M \).

2. \( \kappa = \min_{i \in [k]} \min_{s \in [a]} \min_{r \neq s} \{ C_{rr}^r - C_{rs}^r \} > 0 \)

The first assumption implies that the matrices \( S_{ab} \) are non-singular. The smallest eigenvalue of \( S_{ab} \) controls how many samples we need to approximate \( S_{ab} \) from its sample mean. The second assumption implies that within a group, the probability of assigning the correct label is always higher than the probability of assigning any incorrect label. Note that this assumption might be false for an individual confusion matrix. However, we are averaging over all the users within a cluster to get the cluster average confusion matrix and unless a large fraction of individuals within a cluster has the propensity to mislabel i.e. assign large probability on incorrect labels, this assumption is usually satisfied. The following theorem gives the number of tasks each agent needs to complete to get an \( \varepsilon' \)-estimate of the aggregate confusion matrices. We will use the following two lemmas due to Zhang et al. [Zha+16].

**Lemma 1.12.** For any \( \hat{\varepsilon} \leq \sigma_L/2 \), the second and the third empirical moments are bounded as

\[
\max \{ \| \hat{M}_2 - M_2 \|_{op}, \| \hat{M}_3 - M_3 \|_{op} \} \leq 31 \hat{\varepsilon}/\sigma_L^3
\]

with probability at least \( 1 - \delta \) where \( \delta = 6 \exp \left( -\left( \sqrt{\frac{\hat{\varepsilon}^2 + 1}{2}} - 1 \right)^2 \right) + n \exp \left( -\left( \sqrt{\frac{\hat{\varepsilon}^2}{n}} - 1 \right)^2 \right) \)

**Lemma 1.13.** For any \( \hat{\varepsilon} \leq \kappa/2 \), if the empirical moments satisfy

\[
\max \{ \| \hat{M}_2 - M_2 \|_{op}, \| \hat{M}_3 - M_3 \|_{op} \} \leq \hat{\varepsilon} H
\]

for \( H := \min \left\{ \frac{1}{2}, \frac{2\sigma_L^{3/2}}{15n(24\sigma_L^{-1} + 2\sqrt{2})}, \frac{\sigma_L^{3/2}}{4\sqrt{3/2}\sigma_L^{1/2} + 8n(24/\sigma_L + 2\sqrt{2})} \right\} \)

then \( \| \hat{C} - C \|_{op} \leq \sqrt{n} \hat{\varepsilon}, \| \hat{\Pi} - \Pi \|_{op} \leq \hat{\varepsilon} \) with probability at least \( 1 - \delta \) where \( \delta \) is defined in Lemma 1.12

Zhang et al. [Zha+16] prove Lemma 1.12 when \( \hat{M}_2 \) is defined using the aggregate labels \( R_{gj} \). However, this lemma holds even if one uses the labels \( \hat{R}_{gji} \). The proof is similar if one
uses Lemma 1.11. We now characterize the sample complexity of learning the aggregate confusion matrices.

**Theorem 1.7.** For any $\epsilon' \leq \min\left\{ \frac{31}{\sigma_L^2}, \frac{1}{2} \right\} n^2$ and $\delta > 0$, if the size of the universe of shared tasks $m_2$ is at least $O\left( \frac{n^7}{(\epsilon')^2\sigma_L} \log \left( \frac{nK}{\delta} \right) \right)$, then we have $\|\hat{C}^t - C^t\|_1 \leq \epsilon'$ for each cluster $G_t$. The total number of samples collected by Algorithm 1.4 is $\tilde{O}(Km_2)$ under the uniform assignment scheme.

**Proof.** Substituting $\hat{e} = \hat{e}_1 H \sigma_L^2/31$ in lemma 1.12 we get

$$\max\{\|\hat{M}_2 - M_2\|_{op}, \|\hat{M}_3 - M_3\|_{op}\} \leq \hat{e}_1 H$$

with probability at least $1 - (6 + n) \exp\left( - \left( \frac{m_2^{1/2} \hat{e}_1 H \sigma_L^2}{31n^{7/2}} - 1 \right)^2 \right)$. This substitution requires $\hat{e}_1 H \sigma_L^2/31 \leq \sigma_L/2$. Since $H \leq 1/2$, it is sufficient to have

$$\hat{e}_1 \leq 31/\sigma_L^2 \quad (1.18)$$

Now using Lemma 1.13 we see that $\|\hat{C}^t - C\|_{op} \leq \sqrt{n}\hat{e}_1$ and $\|\hat{I} - I\|_{op} \leq \hat{e}_1$ with the above probability. It can be checked that $H \geq \frac{\sigma_L^2}{2nH}$. This implies that the bounds hold with probability at least $1 - (6 + n) \exp\left( - \left( \frac{m_2^{1/2} \hat{e}_1 H \sigma_L^2}{7130n^{7/2}} - 1 \right)^2 \right)$. The second substitution requires

$$\hat{e}_1 \leq \kappa/2 \quad (1.19)$$

Therefore to achieve a probability of at least $1 - \delta$ we need

$$m_2 \geq \frac{7130^2 H^3}{\hat{e}_1^2 \sigma_L^{11}} \left( 1 + \sqrt{\log \left( \frac{6 + n}{\delta} \right)} \right)^2$$

It is sufficient that

$$m_2 \geq \Omega\left( \frac{n^3}{\hat{e}_1^2 \sigma_L^{11}} \max \left\{ \frac{n}{K} \log \left( \frac{n}{\delta} \right) \right\} \right)$$

to ensure $\|\hat{C}^t - C\|_{op} \leq \sqrt{n}\hat{e}_1$. For each $k$, $\|\hat{C}_k^t - C_k\|_1 \leq \sqrt{n}\|\hat{C}_k^t - C_k\|_2 \leq \sqrt{n}\|\hat{C}^t - C\|_{op} \leq n\hat{e}_1$. Substituting $\hat{e}_1 = \epsilon'/n^2$, we get $\|\hat{C}^t - C\|_1 = \sum_{k=1}^{n} \|\hat{C}_k^t - C_k\|_1 \leq n^2\hat{e}_1 = \epsilon'$ when $m_2 = \Omega\left( \frac{n^7}{(\epsilon')^2\sigma_L} \log \left( \frac{n}{\delta} \right) \right)$. By a union bound the result holds for all the clusters simultaneously with probability at least $1 - \delta K$. Substituting $\delta/K$ instead of $\delta$ gives the bound on the number of samples. Substituting $\epsilon' = \hat{e}_1/n^2$ in equations 1.18 and 1.19, we get the desired bound on $\epsilon'$.

Now to compute the total number of samples collected by the algorithm, note that each
agent in cluster $G_a$ provides $\frac{m_2}{|G_a|} \log \left( \frac{Km_2}{p} \right)$ samples. Therefore, total number of samples collected from cluster $G_a$ is $m_2 \log \left( \frac{Km_2}{p} \right)$ and the total number of samples collected over all the clusters is $Km_2 \log \left( \frac{Km_2}{p} \right)$.

**Discussion.** If the algorithm chooses $m_2 = \tilde{O} \left( \frac{n^2}{(\epsilon')^2 n_p} \right)$, then the total number of samples collected under the uniform assignment scheme is at most $\tilde{O} \left( \frac{n^2}{(\epsilon')^2 n_p} \right)$. So far we have analyzed the Dawid-Skene model under the uniform assignment scheme. When the assignment scheme is fixed, the moments of $R_{aj}$ and $\tilde{R}_{aj}$ need not be the same. In this case we will have to run Algorithm 1.4 with respect to the actual aggregate labels $\{R_{aj}\}_g^{K}$. This requires collecting samples from every member of a cluster, leading to a sample complexity of $O \left( \frac{n^7}{(\epsilon')^2 n_p} \log \left( \frac{K}{\delta} \right) \right)$.

In order to estimate the confusion matrices, Zhang et al. [Zha+16] require each agent to provide at least $O \left( n^5 \log((\ell + n)/\delta)/(\epsilon')^2 \right)$ samples. Our algorithm requires $O \left( n^7 \log(nK/\delta)/(\epsilon')^2 \right)$ samples from each cluster. The increase of $n^2$ in the sample complexity comes about because we are estimating the aggregate confusion matrices in L1 norm instead of the infinity norm. Moreover when the number of clusters is small ($K \ll \ell$), the number of samples required from each cluster does not grow with $\ell$. This improvement is due to the fact that, unlike Zhang et al. [Zha+16], we do not have to recover individual confusion matrices from the aggregate confusion matrices.

Note that the approach based on the work of Dawid and Skene [DS79], for the uniform assignment scheme, does not require all agents to provide reports on the same set of shared tasks. Rather, we need that for each group of three clusters (as partitioned by Algorithm 1.4 on line 1) and each task, there should exist one agent from those three clusters who completes the same task. In particular the reports for different tasks can be acquired from different agents within the same cluster. The assignment scheme makes sure that this property holds with high probability.

We now briefly compare the learning algorithms under the no-assumptions and model-based approach. When it is difficult to assign agents to the same tasks, and when the number of signals is small (which is often true in practice), the Dawid-Skene method has a strong advantage. Another advantage of the Dawid-Skene method is that the learning error $\epsilon'$ can be
made arbitrarily small since each aggregate confusion matrix can be learned with arbitrary accuracy, whereas the true learning error of the no-assumption approach is at least $2\varepsilon$ (see Theorem 1.6), and depends on the problem instance.

### 1.5 Clustering Experiments

Our goal in this section is to empirically evaluate the incentive that an agent has to use a non-truthful strategy under the CAHU mechanism in real-world scenarios. Recall that this incentive error comes from two sources:

- The clustering error. This represents how “clusterable” the agents are. From theory, we have the upper bound $\varepsilon_1 = \max_{p,q\in [k]} \|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1$.

- The learning error. This represents how accurate our estimates for the cluster Delta matrices are. From theory, we have the upper bound $\varepsilon_2 = \max_{i,j\in [K]} \|\Delta_{G_i,G_j} - \bar{\Delta}_{G_i,G_j}\|_1$.

Given this, the CAHU mechanism is $(\varepsilon_1 + \varepsilon_2)$-informed truthful (Theorem 1.1).

In our experiments, we focus solely on the clustering error due to two reasons. First, the available real-world datasets have little overlap between the tasks performed by different agents, making it harder for us to learn their true pairwise $\Delta$-matrices up to a reasonable accuracy and evaluate the error in our estimation. Note that the overlap is only needed to be able to evaluate the learning error of our approach; under the Dawid-Skene model, we do not require any overlap when using our approach in practice.

More importantly, the clustering error and the learning error differ in a key sense. Even with the best possible clustering, the clustering error $\varepsilon_1$ cannot be made arbitrarily small with a fixed number of clusters because it depends on how close the signal distributions of the agents really are. In contrast, the learning error $\varepsilon_2$ of the no-assumption approach is $3\varepsilon' + 2\varepsilon_1$, (Theorem 1.6) from which the part that does not depend on clustering ($\varepsilon'$) can be made arbitrarily small by simply acquiring a sufficient amount of data about agents’ behavior. Similarly, the learning error $\varepsilon_2$ in the Dawid-Skene approach — which we use in this experiment — can be made arbitrarily small too (Theorem 1.7). Hence, given a sufficient amount of data from the agents,
the total error would be dominated by the clustering error \( \varepsilon_1 \). In particular, we show that in practice even a relatively small number of clusters lead to a small clustering error.

We use eight real-world crowdsourcing datasets. Six of these datasets are from the SQUARE benchmark [SL13], selected to ensure a sufficient density of worker labels across different latent attributes as well as the availability of latent attributes for sufficiently many tasks. In addition, we also use the Stanford Dogs dataset [Kho+11] and the Expressions dataset [Moz+14; Moz+12]. Below, we briefly describe the format of tasks, the number of agents \( \ell \), and the number of signals \( n \) for each dataset.\(^4\)

- **Adult**: Rating websites for their appropriateness, \( \ell = 269, n = 4 \).
- **BM**: Sentiment analysis for tweets, \( \ell = 83, n = 2 \).
- **CI**: Assessing websites for copyright infringement, \( \ell = 10, n = 3 \).
- **Dogs**: Identifying species from images of dogs, \( \ell = 109, n = 4 \).
- **Expressions**: Classifying images of human faces by expression, \( \ell = 27, n = 4 \).
- **HCB**: Assessing relevance of web search results, \( \ell = 766, n = 4 \).
- **SpamCF**: Assessing whether response to a crowdsourcing task was spam, \( \ell = 150, n = 2 \).
- **WB**: Identifying whether the waterbird in the image is a duck, \( \ell = 53, n = 2 \).

Since all datasets specify the latent value of the tasks, we adopt the Dawid-Skene model and estimate the confusion matrices from the frequency with which each agent \( p \) reports each label \( j \) in the case of each latent attribute \( i \).

We first use a clustering algorithm to cluster the estimated confusion matrices. Typical clustering algorithms take a distance metric over the space of data points and attempt to minimize the maximum cluster diameter, which is the maximum distance between any two data points in a cluster. In contrast, our objective function (Equation (1.20)) is a complex function of the underlying confusion matrices. We therefore compare two approaches:

\(^4\)We filter each dataset to remove tasks for which the latent attribute is unknown, and remove workers who only perform such tasks. \( \ell \) is the number of agents that remain after filtering.
1) In this approach, we cluster the confusion matrices using the standard \( k \)-means++ algorithm with the \( L_2 \) norm distance (available in Matlab) and hope that resulting clustering leads to a small error.\(^5\)

2) In the following lemma (proof in Appendix A.1), we derive a distance metric over confusion matrices for which the maximum cluster diameter is provably an upper bound on the clustering error, and use \( k \)-means++ with this metric (implemented in Matlab).\(^6\) Note that computing this metric requires knowledge of the prior over the latent attribute (e.g., in the WB dataset, this would require knowing the probability that a random image of a waterbird is a duck), which can be estimated easily from a small amount of ground truth data.

**Lemma 1.14.** For all agents \( p, q, r \), we have 
\[
\| \Delta_{p,q} - \Delta_{p,r} \|_1 \leq 2 \cdot \sum_k \bar{\pi}_k \sum_j |C^q_{kj} - C^r_{kj}|.
\]

Note that \( \sum_k \bar{\pi}_k \sum_j |C^q_{kj} - C^r_{kj}| \leq \|C^q - C^r\|_1 \) because \( \sum_j |C^q_{ij} - C^r_{ij}| \leq \|C^q - C^r\|_1 \). Lemma 1.14, along with Lemma 1.6, shows that the incentive error due to clustering is upper bounded by four times the maximum cluster diameter under our metric, which defines the distance between \( C^q \) and \( C^r \) as \( \sum_k \bar{\pi}_k \sum_j |C^q_{kj} - C^r_{kj}| \).

For each dataset, we vary the number of clusters \( K \) from 5% to 15% of the number of agents in the dataset. Within the \( k \)-means++ algorithm, we use 20 random seeds and choose the best clustering produced.

Next, we compute the clustering error. Instead of using the weak bound \( \max_{p,q \in [T]} \| \Delta_{p,q} - \Delta_{G(p),G(q)} \|_1 \) on the clustering error (which is nevertheless helpful for our theoretical results),

---

\(^5\) We use \( L_2 \) norm rather than \( L_1 \) norm because the standard \( k \)-means++ implementation uses as the centroid of a cluster the confusion matrix that minimizes the sum of distances from the confusion matrices of the agents in the cluster. For \( L_2 \) norm, this amounts to averaging over the confusion matrices, which is precisely what we want. For \( L_1 \) norm, this amounts to taking a pointwise median, which does not even result in a valid confusion matrix. Perhaps for this reason, we observe that using the \( L_1 \) norm performs worse.

\(^6\) For computing the centroid of a cluster, we still average over the confusion matrices of the agents in the cluster. Also, since the algorithm is no longer guaranteed to converge (indeed, we observe cycles), we restart the algorithm when a cycle is detected, at most 10 times.
we use the following tighter bound from the proof of Theorem 1.1.

\[ |u_p^*(I, \{I\}_{q \neq p}) - u_p(I, \{I\}_{q \neq p})| = \left| \frac{1}{(L-1)} \sum_{q \in \mathcal{P}(\{p\})} \sum_{i,j} \Delta_{p,q}(i,j)(\text{Sign}(\Delta_{p,q})_{i,j} - \text{Sign}(\Delta_{C(p),C(q)})_{i,j}) \right| \]

(1.20)

Assuming no learning error, this would be an upper bound on the incentive that agent \( p \) has to use a non-truthful strategy under the CAHU mechanism. We compare this bound to both the maximum payoff that agent \( p \) can receive and the expected payoff that agent \( p \) would receive under our mechanism, and plot the result averaged over \( p \). Figures 1.4a and 1.4b similarly show the incentive of an average agent as a fraction of her maximum payoff with the standard \( L_2 \) metric and with our custom metric, respectively. Figures 1.5a and 1.5b show the incentive of an average agent as a fraction of her expected payoff with standard \( L_2 \) metric and with our custom metric, respectively. We note that the expected payoff is a stronger and more realistic benchmark than the maximum payoff.

In comparison to both the maximum and the expected payoffs, the incentive error is small — less than 20% of the expected payoff and less than 5% of the maximum payoff — even with the number of clusters \( K \) as small as 15% of the number of workers. The number of agents does not seem to significantly affect this bound as long as the number of clusters is a fixed percentage of the number of agents. We also note that using our custom metric leads to a somewhat smaller error than using the standard \( L_2 \) norm.

### 1.6 Conclusion

We have provided the first, general solution to the problem of peer prediction with heterogeneous agents. This is a compelling research direction, where new theory and algorithms can help to guide practice. In particular, heterogeneity is likely to be quite ubiquitous due to differences in taste, context, judgment, and reliability across users. Beyond testing these methods in a real-world application such as marketing surveys, there remain interesting directions for ongoing research. For example, is it possible to solve this problem with a similar sample complexity but without a clustering approach? Is it possible to couple methods of
Figure 1.4: The incentive error as a fraction of the maximum payoff of an agent, averaged over agents, on 8 different data sets when using k-means++ with the L^2 metric and with our custom metric.

(a) Standard L^2 metric
(b) Our custom metric

Figure 1.5: The incentive error as a fraction of the expected payoff of an agent, averaged over agents, on 8 different data sets when using k-means++ with the L^2 metric and with our custom metric.

(a) Standard L^2 metric
(b) Our custom metric

peer prediction with optimal methods for inference in crowdsourced classification [Ok+16], and with methods for task assignment in budgeted settings [KOS14]? This should include attention to adaptive assignment schemes [KO16] that leverage generalized Dawid-Skene models [Zho+15], and could connect to the recent progress on task heterogeneity within peer prediction [Man+16]. Finally, it is worth investigating if we can cluster the agents based on some observable characteristics like demographics, reputation scores etc and reduce the sample complexity of the original mechanism.
Chapter 2

Peer Prediction for Heterogeneous Tasks

2.1 Introduction

Peer prediction refers to the problem of scoring information reports in settings where the correctness of a report cannot be verified, either because there is no objectively correct answer or because this answer is too costly to acquire. This problem arises in diverse contexts; e.g., peer assessment of assignments in massive open online courses, and when collecting feedback about a new restaurant. Peer prediction algorithms use reports from multiple participants to score contributions.

Simple approaches compare the responses of two users and award them if they agree. But this does not promote truthful reporting when one user believes that it is unlikely that another user will have the same opinion. This problem can be alleviated by adjusting scores according to the frequency of reports [JF08; WP12; Kam+15].

A limitation of current approaches, however, is that tasks are assumed to be \textit{ex ante} identical, with each task associated with the same distribution on reports. But tasks on various maps platforms, which seek to elicit content from users about places in a city, are quite heterogeneous. On this kind of platform, a user is encouraged to answer several different types of questions (= tasks) related to the same place; e.g., “is the restaurant noisy?,” “is it accessible by wheelchair?,”
or “does it serve wine?” The questions are related to the same place, yet the prior beliefs about the distribution on reports for each type of question may be very different.

We design a new, multi-task peer prediction mechanism (the correlated agreement-heterogeneous (CAH) mechanism) that is responsive to this challenge. This new mechanism shares similar properties with the earlier correlated agreement (CA) mechanism [Shn+16]. In particular, it is informed truthful under weak conditions, meaning that it is strictly beneficial for a user to invest effort and acquire information, and that truthful reporting is the best strategy when investing effort, as well as an equilibrium. We evaluate CAH and its empirical version on distributions derived from user reports on a popular maps platform. 1 The results show that compared to existing peer prediction mechanisms, our mechanism provides better incentives against unilateral deviations from truthful strategies, and is more robust to collusion arising from coordinated misreports.

2.1.1 Related Work

Miller et al. 2005 introduced the peer prediction problem and proposed a minimal mechanism that has truthful reporting in an equilibrium, however the mechanism’s design requires knowledge of the joint signal distribution and is vulnerable to coordinated misreports. In response, Jurca and Faltings [JF+09] show how to eliminate uninformative, pure-strategy equilibria through a three-peer mechanism, and Kong et al. [KLS16] provide a method to design robust, single-task, binary signal mechanisms. There are also non-minimal mechanisms that elicit both a signal and a belief report [Pre04; WP12].

Witkowski and Parkes [WP12] first introduced the combination of learning and peer prediction, coupling the estimation of the signal prior together with the shadowing mechanism. There has also been work on making use of reports from a large population and coupling scoring with estimation. For a setting with latent ground truth model, Kamble et al. [Kam+15] provide mechanisms that guarantee strict incentive compatibility with a large number of agents. Radanovic et al. [RFJ16] provide a mechanism in which truthfulness is the highest-paying

1Name of platform removed to respect double-blind submission policy. Summary statistics, that define distributions on pairs of signal reports and are used for simulations, will be made available.
equilibrium in the asymptote of a large population and with a self-predicting condition that places a structure on the correlation structure.

Dasgupta and Ghosh [DG13] show that robustness to coordinated misreports can be achieved by using reports across multiple tasks along with access to partial information about the joint distribution. The main insight in the DG mechanism is to reward agents if they provide the same signal on the same task, but punish them if one agent’s report on one task is the same as another’s on another task. Shnayder et al. [Shn+16] generalize DG to handle multiple signals, and show how the required knowledge about the distribution (the correlation structure on pairs of signals) can be estimated from reports without compromising incentives. Their correlated agreement (CA) mechanism rewards pairs of reports on the same task (penalizes pairs of reports on different tasks) based on whether signals are positively or negatively correlated. On the other hand, [Aga+17] generalize the CA mechanism when users are heterogeneous and derive sample complexity bounds for learning the reward matrices. Shnayder et al. [SFP16] adopt replicator dynamics as a model of population learning in peer prediction, and confirm that these multi-task mechanisms (including Kamble et al. [Kam+15]) are successful at avoiding uninformed equilibria.

Liu and Chen [LC17b] designed single-task peer prediction mechanism for heterogeneous tasks only when each task is associated with a latent ground truth. Moreover, their mechanism is vulnerable to collusion by a constant fraction of the population. To the best of our knowledge, there is no prior work on extending the design of these multiple-task mechanisms to heterogeneous tasks, where pairs of reports may be on different types of tasks, with each task associated with a different signal distribution. Our work can be seen as a step forward to the design of robust, collusion-proof multi-task peer prediction mechanisms.

2.2 Heterogeneous, Multi-Task Peer Prediction

Consider two agents, 1 and 2, who are members of a large population. Each agent is assigned to a set of $M = \{1, 2, \ldots, m\}$ tasks. We adopt a binary effort model: if an agent invests effort he incurs a cost and obtains an informed signal, otherwise the agent receives no signal. There are
n signals. We do not assume that tasks are *ex ante* identical, however, we do assume that the signals for different tasks are drawn independently.

Let $S^1_k$ and $S^2_k$ respectively be the signals of agents 1 and 2 for task $k$ (if investing effort). Let $P_k(i,j) = \Pr[S^1_k = i, S^2_k = j]$ be the joint probability for a pair of signals $(i,j)$ on task $k$ and let $P_k(i)$ and $P_k(j)$ be the corresponding marginal probabilities. We assume that the agents are exchangeable in their roles in these distributions, with the same marginal distributions and joint distributions for any pair of agents.

An agent’s strategy maps every task and every received signal to a reported signal. Agents make reports without knowledge of each others’ reports. We assume that the type of task, and signal about a task (upon investing effort), is the only information available to an agent. We allow an agent’s strategy to be randomized i.e. a probability distribution over the set of possible signals.

Following earlier work [Shn+16], we assume that an agent will adopt the same strategy across all task types. While this is without loss of generality for multiple tasks with the same task type, an agent can sometimes do better in the present setting by adopting a different strategy for different types of tasks (see section 2.3.5 for an example). The new aspect to consider is the knowledge of the type of task itself, which allows for a new kind of coordination between participants. We justify our current assumption on grounds of behavioral simplicity, and leave further analysis to future work.

We will write $F$ and $G$ to denote the randomized strategies of agents 1 and 2 respectively. $F_{ij}$ will denote the probability of reporting signal $j$ when user 1 observes signal $i$. For a deterministic strategy $F$, we will just write $F_i$ to denote the reported signal when user 1 observes signal $i$. The notations are analogous for user 2. Let $I$ denote the truthful strategy i.e. $I_j = j$. We are interested in the following two incentive properties:

**Definition 2.1.** (Strong Truthful) A peer prediction mechanism is strong truthful iff for all strategies $F, G$ we have $E(I, I) \geq E(F, G)$, where equality may hold only when $F$ and $G$ are both the same permutation strategy (i.e. a bijection from received signals to reported signals.)

**Definition 2.2.** (Informed Truthful) A peer prediction mechanism is informed truthful iff for all strategies $F, G$ we have $E(I, I) \geq E(F, G)$, where equality may hold only when $F$ and $G$ are informed
strategies (i.e. reports depend on an agent’s signal).

2.2.1 Delta Matrices

Following Shnayder et al. [Shn+16] to multiple types of tasks, a first approach would be to define the following $n \times n$ matrix for task $k$:

$$\Delta_k(i, j) = P_{k}(i, j) - P_{k}(i)P_{k}(j).$$

Let $S_k$ be the sign matrix of $\Delta_k$ i.e. $S_k(i, j) = 1$ if $\Delta_k(i, j) > 0$ and $S_k(i, j) = 0$ otherwise.

In the original CA mechanism [Shn+16], each task is ex ante identical, and thus has the same delta matrix. Denote this matrix $\Delta$, with $S$ the corresponding sign matrix. The original CA mechanism works as follows:

1. Let $r_1^k$ ($r_2^k$) be the signal reported by agent 1 (2) on task $k$.

2. Pick a task $b$ uniformly at random as the bonus task, and pick penalty tasks $l'$ and $l''$ (with $l' \neq l''$) uniformly at random from the remaining tasks.

3. Pay each agent $S(r_1^b, r_2^b) - S(r_1^b, r_2^{l''})$.

A simple generalization is to pay $S_b(r_1^b, r_2^b) - S_b(r_1^b, r_2^{l''})$, where $S_b$ is the sign matrix corresponding to the bonus task. But this is not informed truthful for heterogeneous tasks. This is demonstrated in Example 2.1.

**Example 2.1** (CA is not informed truthful with heterogeneous tasks). Consider three tasks (1, 2 and 3) with the following joint probability distributions

<table>
<thead>
<tr>
<th></th>
<th>Y</th>
<th>N</th>
<th>Y</th>
<th>N</th>
<th>Y</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>0.4</td>
<td>0.22</td>
<td>0.7</td>
<td>0.14</td>
<td>0.4</td>
<td>0.22</td>
</tr>
<tr>
<td>N</td>
<td>0.22</td>
<td>0.16</td>
<td>0.14</td>
<td>0.02</td>
<td>0.22</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>$(P_1)$</td>
<td></td>
<td>$(P_2)$</td>
<td></td>
<td>$(P_3)$</td>
<td></td>
</tr>
</tbody>
</table>
and the following sign matrices:

\[
\begin{align*}
\text{sign}(\Delta_1) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & \text{sign}(\Delta_2) &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, & \text{sign}(\Delta_3) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.
\end{align*}
\]

Suppose each agent adopts the truthful strategy, and task 1 is the bonus task, and 2 and 3 are the penalty tasks for agents 1 and 2, respectively. Then the expected score is

\[
\sum_{i,j} p_{1}(i,j)s_{1}(i,j) - p_{2}(i)p_{3}(j)s_{1}(i,j),
\]

which evaluates to −0.0216. This is true irrespective of whether the penalty tasks for 1 and 2, respectively, are 2 and 3 or 3 and 2. Similarly, we can show that the expected scores are −0.1912 and −0.0216 when the bonus task is task 2 and 3, respectively.

Now consider the case when the first agent always reports N. Suppose task 1 is the bonus task and tasks 2 and 3 are the penalty tasks for 1 and 2, respectively. The expected score is

\[
\sum_{i,j} p_{1}(i,j)s_{1}(N,j) - p_{2}(i)p_{3}(j)s_{1}(i,j),
\]

which evaluates to 0. Similarly, for task 3 and 2 as the penalty for 1 and 2, respectively, the expected score is 0.22. So on average, the expected score for task 1 as bonus is 0.11. Similar calculations show expected scores of 0.22 and 0.11, for tasks 2 and 3 as bonus, respectively. Thus, the CA mechanism fails to be informed truthful for this example.

### 2.3 The Correlated-Agreement Heterogeneous (CAH) Mechanism

In this section, we extend the CA mechanism to handle heterogeneous tasks. The main idea is to modify the delta matrix for a bonus task to allow for the implied product distribution on signals on penalty tasks. Algorithm 2.1 describes the CAH mechanism.

In analyzing the properties of CAH, we note that it is sufficient to consider only deterministic strategies. The proof of this statement is analogous to Lemma 3.2 [Shn+16], and uses the fact that the maximization of a linear function over a convex region is extremal.

Given this, let \( F_i \) (\( G_j \)) denote the report of agents 1 (2) on signal \( i \) (\( j \)). The expected score for
Algorithm 2.1: CAH mechanism

**Input:** Joint probability distribution $P_b(\cdot, \cdot)$, marginal probability distributions $\{P_l(\cdot)\}_{l \neq b}$ and reports $\{r_k^I, r_k^P\}_{k=1}^m$.

1. $b \leftarrow \text{uniformly at random from } \{1, \ldots, m\}$ (bonus task)
2. $l' \leftarrow \text{uniformly at random from } \{1, \ldots, m\} \backslash \{b\}$ (penalty task assigned to agent 1)
3. $l'' \leftarrow \text{uniformly at random from } \{1, \ldots, m\} \backslash \{b, l'\}$ (penalty task assigned to agent 2)
4. Define $\Delta_b(i, j)$ as
   \[
   \Delta_b(i, j) = \frac{1}{(m-1)(m-2)} \sum_{i', j'' \in [m] \backslash \{b\} \& \ i' \neq j''} P_{l'}(i)P_{l''}(j) \tag{2.2}
   \]
5. Let $S_b(i, j)$ be the corresponding score matrix i.e.
   \[
   S_b(i, j) = 1 \text{ if } \Delta_b(i, j) > 0 \text{ and } S_b(i, j) = 0 \text{ otherwise}
   \]
6. Make payment $S_b(r_b^I, r_b^P) - S_b(r_{l'}^I, r_{l''}^P)$ to each agent

strategies $F$ and $G$, conditioned on some bonus task $b$, denoted as $E_b(F, G)$, is:

\[
E_{l', l''} \left[ \sum_{i, j} P_b(i, j)S_b(F_i, G_j) - \sum_{i, j} P_{l'}(i)P_{l''}(j) \right]
= \sum_{i, j} P_b(i, j)S_b(F_i, G_j) - \sum_{i, j} \frac{1}{(m-1)(m-2)} \sum_{i', j'' \in [m] \backslash \{b\} \& \ i' \neq j''} P_{l'}(i)P_{l''}(j)S_b(F_{i'}, G_{j'})
= \sum_{i, j} \Delta_b(i, j)S_b(F_i, G_j), \tag{2.3}
\]

where $l'$ and $l''$ denote agent 1 and agent 2’s penalty tasks, respectively. Thus, the expected score, averaged over the $m$ possible bonus tasks, is

\[
E(F, G) = \frac{1}{m} \sum_{b=1}^m E_b(F, G) = \frac{1}{m} \sum_{b=1}^m \sum_{i, j} \Delta_b(i, j)S_b(F_i, G_j) \tag{2.4}
\]

We now state a property about the delta matrices (2.2), the proof of which is in Appendix B.1.

**Lemma 2.1.** For each task $b$, we have $\sum_{i, j} \Delta_b(i, j) = 0$
2.3.1 Informed Truthfulness

The CAH mechanism is informed truthful under a weak condition on the signal distributions.

**Theorem 2.1.** If for each task \( b \), \( \Delta_b \) is symmetric and each entry of \( \Delta_b \) is non-zero, then the CAH mechanism is informed truthful.

**Proof.** For any bonus task \( b \), the truthful strategy \((I, I)\) has higher expected score than any other pair of strategies \( F, G \):

\[
E_b(I, I) = \sum_{i,j} \Delta_b(i, j) S_b(i, j) = \sum_i \sum_j \max(0, \Delta_b(i, j)) \geq \sum_{i,j} \Delta_b(i, j) S_b(F, G) = E(F, G).
\]

Consider an uninformed strategy \( F \), with \( F_i = r \) for all \( i \). Then for any \( G \), the expected score is

\[
\sum_{i=1}^n \sum_{j=1}^n \Delta_b(i, j) S_b(r, G_j) = \sum_{j=1}^n S_b(r, G_j) \sum_{i=1}^n \Delta_b(i, j) \leq \sum_{j=1}^n \max(0, \sum_{i=1}^n \Delta_b(i, j)).
\]

We need to show the following:

\[
\sum_{j=1}^n \max(0, \sum_{i=1}^n \Delta_b(i, j)) < \sum_{i=1}^n \sum_{j=1}^n \max(0, \Delta_b(i, j)).
\]

It is enough to show that for each \( b \), there exists a column \( j \) and two different rows \( i_1, i_2 \) such that \( \Delta_b(i_1, j) > 0 \) and \( \Delta_b(i_2, j) < 0 \). Suppose not. Then each column of \( \Delta_b \) has either all positive entries or all negative entries. Since each entry of \( \Delta_b \) is non-zero and Lemma 2.1 holds, there exist two columns \( j_1 \) and \( j_2 \) such that all entries of \( j_1 \) (\( j_2 \)) are positive (negative). This implies \( \Delta_b(j_2, j_1) > 0 \) and \( \Delta_b(j_1, j_2) < 0 \), which contradicts the fact that \( \Delta_b \) is symmetric. \( \square \)

Note that, we assume that the agents are exchangeable. This means that the joint probability distribution \( P_b \) is symmetric and so is \( \Delta_b \). Now, if \( \Delta_b(i, j) = 0 \) then the probability that users observe signals \((i, j)\) on task \( b \) is the same as they observe signals \((i, j)\) on two randomly selected different tasks. To understand why this is a very low probability event, consider a generative model of heterogeneous task types. For reasonable models of heterogeneity the probability of equality would be negligible. In fact, the condition can be further weakened. We only need it to hold for one task \( b \), and not for every \( b \). And for that task, we need that there exists signals
\(j, i_1, i_2\), with \(i_1 \neq i_2\), such that \(\Delta_b(i_1, j) > 0\) and \(\Delta_b(i_2, j) < 0\).

### 2.3.2 Strong Truthfulness

We state a sufficient condition for the CAH mechanism to satisfy the property of strong truthfulness.

**Condition 1:**

1. \(\Delta_b(i, i) > 0\), \(\forall b \forall i\).
2. \(\sum_{b=1}^{m} \Delta_b(i, j) < 0\), \(\forall i \neq j\).

**Theorem 2.2.** If \(\{\Delta_b\}_{b=1}^{m}\) satisfy Condition 1, then the CAH mechanism is strongly truthful.

**Proof.** Suppose both the agents adopt the truthful strategy, which corresponds to the identity matrix \(I\). Then the expected payment is given as

\[
E(I, I) = \sum_{b=1}^{m} \sum_{i, j : \Delta_b(i, j) > 0} \Delta_b(i, j) \tag{2.5}
\]

On the other hand for any two arbitrary deterministic strategies \(F\) and \(G\),

\[
E(F, G) = \sum_{b=1}^{m} \sum_{i, j} \Delta_b(i, j) S_b(F_i, G_j) \leq \sum_{b=1}^{m} \sum_{i, j : \Delta_b(i, j) > 0} \Delta_b(i, j) = E(I, I) \tag{2.6}
\]

To show strong truthfulness, consider a joint strategy \(F \neq G\). Then there exists \(i\) such that \(F_i \neq G_i\). This reduces the expected payment by at least

\[
\sum_{b=1}^{m} \Delta_b(i, i) S_b(F_i, G_i) \tag{2.7}
\]

Since \(F_i \neq G_i\), we have \(\sum_{b=1}^{m} \Delta_b(F_i, G_i) < 0\) and there exists \(l'\) such that \(\Delta_{b'}(F_{l'}, G_{l'}) < 0\) (or \(S_{b'}(F_{l'}, G_{l'}) = 0\)). Therefore, the expected payment reduces by at least \(\Delta_{b'}(i, i) > 0\).

Now consider identical, non-permutation strategy \(F = G\). Then there exist \(i \neq j\) such that \(F_i = G_j = k\) and the expected payment includes

\[
\sum_{b=1}^{m} \Delta_b(i, j) S_b(k, k) = \sum_{b=1}^{m} \Delta_b(i, j) < 0 \tag{2.8}
\]

The first equality uses the fact \(S_b(k, k) = 1\) since \(\Delta_b(k, k) > 0\) for each \(b\). \(\square\)
Condition 1 is slightly weaker than the categorical condition [Shn+16]. $\Delta_b$ is categorical if (1) $\Delta_b(i, i) > 0$ for all signals $i$, and (2) $\Delta_b(i, j) < 0$ whenever $i \neq j$; i.e., same-signal positive correlation and other-signal negative correlation. Condition 1 does not require every off-diagonal entry to be negative for all tasks $b$, but only that the average of the off-diagonal entries is negative. Categorical and Condition 1 are equivalent when there are only two signals.

### 2.3.3 Combining CAH with Estimation

As with the CA mechanism [Shn+16], the CAH mechanism remains (approximately) informed truthful even when the statistics used to determine scores are estimated from the reports of strategic agents. The reason is that the score matrix that corresponds to the correct statistics is the best possible score matrix for agents, and thus they cannot do better by cooperating in designing an alternate matrix.

(Algorithm 2.2) presents the detail-free version of CAH mechanism, which learns the delta matrices from the agents’ reports. We will refer to this implementation as CAHR (in short for CAH recomputed). The next theorem proves that CAHR is $(\varepsilon, \delta)$-informed truthful. Appendix B.1 includes a proof of this theorem.

**Theorem 2.3.** If there are at least $q = \Omega \left( \frac{n}{\varepsilon^2} \log \left( \frac{m}{\delta} \right) \right)$ agents reviewing each task, for $m$ tasks and $n$ possible signals, then with probability at least $1 - \delta$, then CAHR satisfies

$$E[\mathbb{I}, \mathbb{I}] \geq E[F, G] - \varepsilon \quad \forall F, G$$

Theorem 2.3 implies that truthful reporting is an approximate equilibrium for the detail-free CAH, and that (up to $\varepsilon$) there is no useful joint deviation. The proof follows from the fact that any joint distribution $P_b(\cdot, \cdot)$ (resp. marginal distribution $P_b(\cdot)$) can be learned with $\tilde{O}(n^2/\varepsilon^2)$ (resp. $\tilde{O}(n/\varepsilon^2)$) samples\(^2\) and observing that $q$ samples from a task gives us $q^2$ samples from the corresponding joint distribution. In addition, we can show a general version of Theorem 2.3. Suppose there are $t$ distinct types of tasks, and the number of tasks of type $k$ is $m_k$. Then it is sufficient to have $q = \tilde{\Omega} \left( \frac{1}{\sqrt{m_k \varepsilon^2}} \right)$ samples from each task of type $k$. This follows from the

\(^2\tilde{O}(\cdot)\) is $O(\cdot)$ without all the log factors
observation that if we have at least $q$ samples from each task of type $k$ then the total number of samples from the joint distribution $P_k(\cdot, \cdot)$ is at least \( m_k q^2 = \tilde{\Omega} \left( \frac{n^2}{\varepsilon^2} \right) \).

**Algorithm 2.2: CAHR mechanism**

**Input:** Agent $p$ of a population of $q$ agents provides reviews $(r^p_1, \ldots, r^p_m)$ on each of the $m$ tasks.

1. $T_k(i, j) \leftarrow$ observed freq of signal pair $i, j$ on task $k$;
2. Pair up the agents uniformly at random, and run CAH for each pair with the estimated distribution $\{T_k(\cdot, \cdot)\}_{k=1}^m$.

### 2.3.4 Cross Correlated Agreement

So far we have assumed that the probabilities of observing signals are independent across different tasks. However, two users' responses to two different tasks may be correlated (e.g. consider two questions – (1) does this restaurant serve alcohol and (2) does this restaurant serve wine?) We will write $P_{l', l''}(i, j)$ to denote the probability that a user sees signal $i$ on task $l'$ and another user observes signal $j$ on task $l''$. When there are no correlations among signals for different questions we have $P_{l', l''}(i, j) = P_{l'}(i)P_{l''}(j)$. The **Cross Correlated Agreement for Heterogeneous Tasks** (CCAH) mechanism generalizes CAH by using the probabilities $P_{l', l''}(\cdot, \cdot)$ for different pairs of tasks $(l', l'')$.

- CCAH is same as CAH except it defines $\Delta_b(i, j)$, the $(i, j)$-th entry of delta matrix for task $b$ as:

  $$P_b(i, j) - \frac{1}{(m - 1)(m - 2)} \sum_{l', l'' \in \{l_1, \ldots, l_m\} \setminus \{b\} \cap l' \neq l''} P_{l', l''}(i, j)$$

  CCAH is strong truthful and informed truthful under similar conditions stated in theorems 2.2 and 2.1 respectively. Moreover, a sample complexity result analogous to Theorem 2.3 holds for a detail-free implementation of CCAH. This is because if we have at least $q$ samples from both $l'$ and $l''$, then we have at least $q^2$ samples from the joint distribution $P_{l', l''}(\cdot, \cdot)$.
2.3.5 Asymmetric Strategy

In this section, we construct an example to show that if the agents can use the type of the task to adopt asymmetric strategy profiles, they can coordinate to obtain strictly better score than the truthful strategy profile. Consider the following example. There are \( m/2 \) tasks of type \( A \) and \( m/2 \) tasks of type \( B \) with the following joint probability matrices.

\[
\begin{bmatrix}
Y & N \\
Y & 0.4 & 0.1 \\
N & 0.1 & 0.4
\end{bmatrix}
\quad \begin{bmatrix}
Y & N \\
Y & 0.1 & 0.4 \\
N & 0.4 & 0.1
\end{bmatrix}
\]

\((A)\quad (B)\)

For large enough \( m \), the corresponding \( \Delta \) and sign matrices are given as:

\[
\begin{bmatrix}
Y & N \\
Y & 0.15 & -0.15 \\
N & -0.15 & 0.15
\end{bmatrix}
\quad \begin{bmatrix}
Y & N \\
Y & 0.15 & -0.15 \\
N & -0.15 & 0.15
\end{bmatrix}
\]

\((\Delta_1)\quad (\Delta_2)\)

\[
\begin{bmatrix}
Y & N \\
Y & 1 & 0 \\
N & 0 & 1
\end{bmatrix}
\quad \begin{bmatrix}
Y & N \\
Y & 0 & 1 \\
N & 1 & 0
\end{bmatrix}
\]

\(\text{(sign}(\Delta_1))\quad \text{(sign}(\Delta_2))\)

Under the truthful strategy profile, the expected score is the sum of the delta entries for which the sign entries are positive. Therefore, an agent get a score of \( 3/10 \) irrespective of the type of the task, yielding a total expected score of \( 3m/10 \). On the other hand, suppose the two agents adopt the following strategy: always report \( Y \) on tasks of type \( A \) and signal \( N \) on tasks of type \( B \). It is easy to show that this yields a payoff of \( 1/2 \) irrespective of the type of the task, yielding a total score of \( m/2 \) in expectation. Therefore, the agents can improve their expected utility by at least \( m/10 \) by an asymmetric strategy profile.
2.4 Experimental Results

Google Local Guides is a platform for collecting user generated content in regard to places on Google Maps. A user can provide information by answering ‘yes’, ‘no,’ or ‘not sure’ to a series of questions.\(^3\) A user is awarded one point for each contribution, where a contribution can be a review or a photograph or any update about the place, with a maximum of five points per place. Based on the number of points received a user is in one of five levels on the platform, with higher levels providing better benefits such as free Google Drive space, visibility on the Local Guides channel, and access to new products before they are generally released.

A type of task is specified by a triple of the form:

\[ \text{Region} \times \text{BusinessType} \times \text{Question} \]

A region is a US state, there are four business types such as “restaurant,” “bar,” “public location” or “cafe” (these are anonymized in our data), and there are 143 distinct questions in the data. The questions are also anonymized, but categorized by Google as “subjective” or “factual” (e.g., “is this restaurant noisy?” vs “does this cafe have free WiFi?”). Each task type has a corresponding pairwise signal distribution.

The data are counts of pairs of signal reports, broken down by (region, business type, question). The number of different questions (and thus types of tasks) per pair of region and business type varies from 75 to 135, with an average of 102. There are 51 regions and 4 business types per state. Thus, the total number of task types for which we have data is around 20,885.

For the purpose of our simulations we treat the distributions for these task types as describing the true signal distributions. The goal of the experiments is to compare, under this assumption, the robustness of the CAH mechanism with other mechanisms in the literature. For this, we consider the robust peer truth serum (RPTS) mechanism [RFJ16] (which sets a score of \( \alpha \left( \frac{1}{\hat{P}(i)} - 1 \right) \) for agreement on signal \( i \) and \(-\alpha\) otherwise) and the Kamble [Kam+15]

\(^3\)We ignore the ‘not sure’ response for a question because of unclear semantics: does it mean the user has missing information, or the question is not relevant to the location. Thus, \textit{a priori} it is unclear whether to expect correlation between different reports.
mechanism \(^4\) (which sets a score of \(1/\sqrt{\hat{P}(i,i)}\) for agreement on signal \(i\)). Here we will write \(P(\cdot, \cdot)\) (resp. \(P(\cdot)\)) to denote the true joint (resp. marginal) probabilities. And we write \(\hat{P}(\cdot, \cdot)\) (resp. \(\hat{P}(\cdot)\)) to denote the joint (resp. marginal) probabilities recomputed after some possible misreport.

Since CAH payments are always bounded between 0 and 1, we normalize the payments of RPTS and Kamble mechanisms so that their payments are always in \([0,1]\). Suppose the agents are reporting truthfully. Then RPTS pays \(\alpha (1/P(i) - 1)\) for agreement on signal \(i\) and \(-\alpha\) otherwise. So, we first add \(\alpha\) to the payment irrespective of the report (i.e. pay \(\alpha(1/P(i))\) for agreement on signal \(i\) and 0 otherwise). Now suppose \(P(0) > P(1)\) then RPTS pays more than 1 on agreement on signal 1. So we choose \(\alpha = \min(P(0), P(1))\) so that the payment always lies in \([0,1]\). For Kamble, we multiply the payment by a normalization factor \(\min(\sqrt{P(0,0)}, \sqrt{P(1,1)})\) such that the payments are again bounded between 0 and 1. Note that, our normalization scheme is static i.e. the normalization constants are not recomputed when the probabilities are estimated based on some possible misreports of the agents. When the agents misreport, we recompute the joint and marginal probabilities but use the fixed normalization constants mentioned above. This ensures that the implemented versions of the two mechanisms are just the scaled versions of RPTS and Kamble and have equivalent incentive properties in expectation.

In simulating CAH, we first compute the delta matrices for each task type using Equation (2.2). For this, we assume for a given (region, business type, question) that the penalty tasks are sampled from other questions associated with the same (region, business type). From these delta matrices, we then use Equation (2.3) to compute the expected score for each question, before averaging these scores over all questions associated with a (region, business type) pair. For the single task, RPTS and Kamble mechanisms, we compute the score for a (region, business type) by averaging the individual scores received on each question associated with the (region, business type) pair. Finally, since the payments of CAH are bounded between

\(^4\)Kamble et al. [Kam+15] also propose a mechanism for heterogeneous agents (Mechanism 2 in the paper). However, we don’t evaluate that mechanism here because (a) we are concerned with heterogeneity due to tasks and (b) On agreement on signal \(i\), mechanism 2 sets scores inversely proportional to the empirical frequency of signal \(i\). This is essentially a scaled version of the RPTS mechanism.
0 and 1, we normalize the payments of RPTS and Kamble to [0, 1]. Along with CAH, we also evaluate CAHR, the empirical version of the CAH mechanism. CAH has access to the true delta matrices, whereas, CAHR computes the delta matrices based on the reports of the agents and then uses these delta matrices to score reports.

### 2.4.1 Unilateral Incentives for Truthful Reports

We consider three kinds of strategic behaviors: constant-0 (report ‘yes’ all the time), constant-1 (report ‘no’ all the time) and random (report ‘yes’ w.p. 0.5).

We first consider unilateral incentives to make truthful reports, for various assumptions about how the behavior of the rest of the population. As an illustration, Figure 2.1 shows the expected benefit to being truthful vs following some other behavior, considering the average score for each (region, business type). We consider, in particular, the benefit to being truthful vs the alternate behavior when \( p = 0.8 \) of the population is truthful and the rest follow the same, alternate strategy. This models 20% of the agents being able to coordinate on a deviation from truthful play. \(^5\)

We observe that the support of the distribution for the CAH and CAHR mechanism is positive, and thus it retains an incentive for truthful behavior. We found this to be a common property for different values of \( p \), i.e. CAH and CAHR retains good unilateral incentives for all values of \( p \), even when all agents play the same way. By contrast, both the RPTS and Kamble fail under some strategy, i.e. there exists a strategy (random for Kamble and either random or constant-1 for RPTS) such that playing that strategy is more beneficial than playing truthful strategy when some fraction plays this alternate strategy. Although Figure 2.1 shows this for \( p = 0.8 \), we find this is representative of other values of \( p \) as well.

When the prior probability satisfies the self-predicting \(^6\) condition, the RPTS mechanism has truth-telling as a strict equilibrium and the truthful equilibrium provides at least as high payoff

---

\(^5\)For CAHR, we first recompute the joint probabilities when \( p \) fraction of the population is truthful and \( 1 - p \) fraction adopts some other strategy, and then compute the delta matrices with respect to the new joint probability distributions. On the other hand, CAH uses the delta matrices computed using the original joint probability distributions.

\(^6\)\( P(\cdot,\cdot) \) satisfies self-predicting if \( P(x|x) > P(x|y) \) for \( x \neq y \).
Figure 2.1: Histograms for the 204 (region, business type) pairs of expected benefit (averaged across questions) from truthful behavior vs. some other strategy, when fraction 0.8 is truthful and fraction 0.2 adopt the same, non-truthful strategy. Compared to RPTS and Kamble, CAH and CAHR always have positive support i.e. they always provide positive incentive to be truthful.

than any other coordinated equilibrium where all agents report the same. Since, incentive properties are not proven under RPTS except when the self-predicting condition is satisfied, we evaluated the RPTS mechanism by restricting only to questions that satisfy the self-predicting condition. However, the corresponding plot is similar to the plot shown in figure 2.1. To conclude, compared to single task mechanisms like RPTS and Kamble, CAH mechanisms provide good guarantees against unilateral deviation.

2.4.2 Benefit from Coordinated Misreports

Irrespective of whether or not a coordinated deviation is robust against agents choosing to make truthful reports instead, we also consider the expected payoff available to a group of
agents who manage to coordinate on some non-truthful play. Figure 2.2 plots the average and standard error for the expected payments associated with the 204 (region, business type) pairs. For each strategy and for a particular value of $p$, we plot the expected payment and the standard error across the 204 pairs, when $p$ fraction of population is truthful and the remaining $1 - p$ fraction of the population adopts the same strategy. The constant line shows the average expected payment across all the pairs when everyone is truthful.

**Figure 2.2:** Expected score for following each of four strategies, when $p$ fraction of the population is truthful and $1 - p$ fraction adopt the same strategy. The scores are averaged over questions associated with a typical (region, business type) pair. For RPTS and Kamble, we omit the plots for the expected score for const-0 and const-1 strategies as the scores under these strategies are significantly lower than the all truthful strategy. Both of them vulnerable to collusion by the random strategy for intermediate values of $p$.

Both CAH and and its recomputed version CAHR have the expected payments from all truthful strategy higher than the other three possible strategies (const-0, const-1 and random) for all possible values of $p$. This means that CAH and CAHR are robust against coordinated misreport by any fraction of the population. In fact, CAHR provides even stronger resistance
against such coordinated misreports. For RPTS and Kamble, we only plot the expected payments due to the all truthful strategy and the random strategy for various values for $p$. We omit the plots for the expected payments for const-0 and const-1 strategies since the payments under these strategies are significantly lower than the all truthful strategy under both RPTS and Kamble mechanism and do not provide profitable coordinated misreports. We now see that for intermediate values of $p$, the random strategy provide a profitable coordinated misreporting profile under both the RPTS and Kamble mechanism. Therefore, unlike CAH, single task mechanisms like RPTS, Kamble are not always robust to coordinated deviations.

### 2.4.3 Subjective vs Factual Tasks

Figure 2.3 shows the cumulative distribution on expected scores at truthful reporting in each mechanism, where each data point corresponds to a different (region, business type, question) triple. Two lines are shown for each mechanism: one corresponding to questions that are categorized as ‘factual’ and one corresponding to questions that are categorized as ‘subjective.’

![Figure 2.3: Cumulative distribution on expected payments at truthful reporting in each mechanism, with results separated into questions that are categorized as ‘factual’ and those that are categorized as ‘subjective.’](image)

The subjective questions tend to provide lower expected payment than the factual questions under the CAH mechanism. This is consistent with the intuition that people perceive subjective questions differently than factual questions. For the Kamble and RPTS mechanisms, the variability in expected payment is larger across factual questions than subjective questions, with the expected payment for subjective questions tending to fall in a narrow band.
2.5 Conclusion

We study the peer prediction problem when users complete heterogeneous tasks. We introduced the CAH mechanism, which is informed-truthful under mild conditions and can also be used together with estimating statistics from reports for the purpose of computing scores. The simulation results suggest that CAH provides better incentive for being truthful and is more resistant to coordinated misreports than the RPTS and Kamble mechanisms. We also noted that CAHR, the empirical version of CAH has similar incentive guarantees, in contrast to the empirical versions of the single-task peer prediction mechanisms. However, we would like to note that CAH remains interesting, particularly when a system has only a small amount of new reports but there is good information about the true world model. We believe that the theoretical guarantees of the multi-task mechanisms and their attractive incentive properties suggest that such mechanisms are ready to be applied and evaluated in practice, from peer grading to rating. The most important directions for future work are to design mechanisms that can handle agent heterogeneity (agents that vary by taste, judgment, noise, etc.) as well as task heterogeneity. We are also interested in developing specific versions of the CAH mechanism for particular models of heterogeneity, such as the generalized Dawid-Skene scheme [DS79; Zho+15], which models both task heterogeneity and agent heterogeneity in the presence of a latent ground truth.
Chapter 3

Correlated Voting

3.1 Introduction

Social choice theory studies how to aggregate preferences and select an outcome. A canonical problem is voting, where reports are preferences over a list of candidates and a voting rule selects the winning candidate.

The problem of voting is ubiquitous systems in which multiple agents interact with each other. Several tools and ideas from social choice theory have found applications in different areas of multi-agent systems including resource allocation [PPS15], rank aggregation [AT10], recommender systems [PHG00], choosing between multiple issues [LX09], and crowdsourcing [MPC13]. Voting also has strong connections to rank aggregation models in machine learning. These models view rank data as a noisy version of a true underlying rank, and try to recover the true rank.

It is typical to ignore the possibility that data is misreported in order to change the aggregate view. Consider, for example, the problem of a conference review system where a reviewer gives a ranking over a subset of submitted papers. The committee wants to aggregate truthful reports of opinion on submitted papers while different subcommunities may want to push decisions in one direction or another.

Indeed, the problem of incentive-aligned social choice is generally hopeless. Gibbard [Gib73] and Satterthwaite [Sat75] prove that when the number of alternatives is at least three, a voting
rule is unanimous and strategy-proof if and only if it is dictatorial. There are two standard ways of circumventing the Gibbard-Satterthwaite impossibility result. First, the preferences of voters may arise from a restricted preference class [Bar11]. Second, some have appealed to worst-case and average-case computational intractability of the problem of strategic manipulation, arguing that this provides robustness to strategic behavior ([FP10] and for a recent survey, see [CW16]).

However, these approaches do not consider a probabilistic model of social choice, with preferences sampled from a distribution. The relevant incentive question is to consider the situation that each voter may have information about how likely the preferences of other voters’ are. Our particular interest is in studying the problem of incentive-aligned preference aggregation when voters’ preferences are not necessarily independent. Whereas it is typical to assume that individual rank preferences of voters are independent, this is not always true. For example, in a conference review mechanism, once a reviewer realizes that paper A is significantly better than paper B, they may be more likely to place more probability on others thinking the same about the papers.

Majumdar and Sen [2004] initiated the study of *ordinally Bayesian incentive compatible* (OBIC) voting rules for the setting in which voters have incomplete information about each others’ preferences. OBIC is a weaker form of strategy-proofness, stating that truthful reporting is a Bayes-Nash equilibrium for any cardinal utility consistent with agent preferences. Later, Sen et al. [SBM15] prove that the Gibbard-Satterthwaite impossibility result breaks down if beliefs are positively correlated. These authors start with a social choice function $f$ satisfying certain properties and exhibit a class of positively correlated distributions for which $f$ is OBIC.

Instead of OBIC, we adopt the notion of *ex post incentive compatibility* (EPIC). EPIC is a weaker solution concept than strategy-proofness, and requires truth-telling to be a best response to every preference profile of others provided they are also truthful. Our goal is to understand the following question: given positively correlated beliefs that are neither uniform nor extremely correlated, how do common voting rules, and in particular scoring rules, perform with regard to EPIC?

We take an asymptotic view along the lines of Baharad and Neeman [BN02], who prove that when the preferences of agents are drawn uniformly and independently at random the probability that any scoring rule is not EPIC goes to zero at a rate proportional to $1/\sqrt{n}$.
for $n$ agents. Their result also holds for small, local correlations among preferences, but fails when the correlation is global or large enough. We first prove that positive correlation helps dramatically. Our main result is that the probability that any scoring rule is not EPIC goes to zero exponentially fast when an agent’s belief is that the preferences of others are positively correlated with his own preference order according to the Kendall-Tau distance. We also establish a general result for any conditionally independent and identical beliefs, showing convergence of scoring rules to EPIC at a rate proportional to $1/\sqrt{n}$.

Motivated by rank-order models from machine learning [Mar96], we introduce two examples of positively-correlated belief systems, namely Conditional Mallows and Conditional Plackett-Luce. These two families of belief systems span a wide range of positively correlated beliefs— from being arbitrarily close to uniform to being extremely correlated. Conditional Mallows is Kendall-Tau correlated and fits our main positive result. Conditional Plackett-Luce model is not, but we provide a different proof of exponential convergence to EPIC for this model.

### 3.2 Setup

The set of alternatives is $A = \{1, \ldots, m\}$ and the set of voters is $N = \{1, \ldots, n\}$. Let $\mathbb{P}$ be the set of all linear orders over $A$. Voter $i$ has a preference order $P_i \in \mathbb{P}$ over the $m$ alternatives. Any voting rule is represented by a social choice function $f : \mathbb{P}^n \rightarrow A$. We will write $P_{-i}$ to denote a preference profile of all the voters other than $i$. We will use $aP_ib$ to denote that alternative $a$ is preferred over alternative $b$ according to the preference order $P_i$.

Given preference $P_i$, let $\mu_i(P_{-i}|P_i)$ denote the probability that voter $i$ ascribes to voters other than $i$ having preference profile $P_{-i} \in \mathbb{P}^{n-1}$. Note that like Sen et al. [SBM15], we do not insist that the conditional probabilities $\mu_i(P_{-i}|P_i)$ should be generated from a given underlying common prior over the entire profile of $n$ voters. We will call a collection of conditional probability distributions $\mu_1, \ldots, \mu_n$ a belief system.

---

1 Although Baharad and Neeman [BN02] use the term “asymptotic strategyproofness”, they actually consider EPIC as the notion of equilibrium.
We adopt \textit{ex post} incentive compatibility (EPIC) as a solution concept. A voting rule is \textit{strategy-proof} if truthful reporting is a dominant strategy for each voter. EPIC provides truthful reporting as a best response to any preference profile of others, provided they are also truthful.

\textbf{Definition 3.1.} A social choice function \( f : \mathcal{P}^n \rightarrow A \) is EPIC with respect to the belief system \( \{ \mu_i \}_{i=1}^n \), if for each agent \( i \) and \( \forall P_i, P'_i \)

\[
f(P_i, P_{-i})P_i f(P'_i, P_{-i}) \forall P_{-i} \text{ in support of } \mu_i(\cdot|P_i)
\] (3.1)

For \( \mu_i(\cdot|P_i) \) with full support over \( \mathcal{P}^{n-1} \) for every \( P_i \), then \( f \) is EPIC iff it is also strategy-proof. However, EPIC and strategy-proof social choice functions become very different when we compute the probability that an agent can manipulate. Suppose agent \( i \) has preference \( P_i \) and \( L(P_i) = \{ P_{-i} \in \mathcal{P}^{n-1} : \exists P'_i f(P'_i, P_{-i})P_i f(P_i, P_{-i}) \} \), the set of all preference profiles of others such that agent \( i \) wants to deviate. Then the probability that \( i \) can manipulate in the sense of a dominant strategy is either 0 or 1 (it is 0 if \( L(P_i) = \emptyset \), and 1 otherwise). On the other hand, the probability that \( i \) can manipulate in the sense of an \textit{ex post} Nash equilibrium is \ \( \sum_{P_{-i} \in L(P_i)} \mu_i(P_{-i}) \). We are interested in showing that the last probability becomes exponentially small in the number of agents when the belief system is positively correlated.

\subsection{3.2.1 Positively Correlated Preferences}

Sen et al. [SBM15] introduce two types of correlation for a given belief system \( \{ \mu_i \}_{i=1}^n \): Top-Set correlation and Kendall-Tau correlation. Let \( B_k(P_i) \) denote the set of top \( k \) alternatives as ranked by \( P_i \). A belief system is Top-Set correlated (TS-correlated) if every voter \( i \) believes that the event where every other voter has the same top-\( k \) set of alternatives as \( i \) is strictly more likely than the event where every other voter has some other subset \( T \) as their top-\( k \) set.

\textbf{Definition 3.2 (Top-Set Correlated).} Belief system \( \{ \mu_i \}_{i=1}^n \) is TS-correlated if \( \forall k \in \{1, \ldots, m-1\}, \forall i, \forall P_i, \forall T \neq B_k(P_i) \) and \( |T| = k \):

\[
\sum_{\{P_{-i}: \forall j \neq i \in B_k(P_i) \}} \mu_i(P_{-i}|P_i) > \sum_{\{P_{-i}: \forall j \neq i \in B_k(P_i) = T \}} \mu_i(P_{-i}|P_i)
\] (3.2)
A belief system is *Kendall-Tau correlated* (KT-correlated) if every voter $i$ believes that the preference profiles of other voters are ordered in decreasing probability according to increasing sum Kendall-Tau distance between the preferences of others and voter $i$’s true preference $P_i$. Let $d(P_i, P_j)$ be the *Kendall-Tau distance* (i.e., the number of pairwise disagreements) between preferences $P_i$ and $P_j$.

**Definition 3.3 (Kendall-Tau Correlated).** Belief system $\{\mu_i\}_{i=1}^n$ is KT-correlated if for all $P_i$, $P'_{-i}$

$$\mu_i(P_{-i}|P_i) > \mu_i(P'_{-i}|P_i) \text{ if } \sum_{j \neq i} d(P_j, P_i) < \sum_{j \neq i} d(P'_j, P_i).$$

We use $D(P_{-i}|P_i)$ in place of $\sum_{j \neq i} d(P_j, P_i)$. It is easy to show that any KT-correlated belief system is also TS-correlated, but not vice-versa. [BMS11]

**Definition 3.4.** A belief system $\{\mu_i\}_{i=1}^n$ is *conditionally independent and identically distributed* (c.i.i.d.) if

$$\forall i, \forall P_i, \forall P_{-i} : \mu_i(P_{-i}|P_i) = \prod_{j \neq i} v(P_j|P_i)$$

(3.3)

where $v(\cdot|P_i)$ is a probability distribution over orders $\mathbb{P}$.

We work with c.i.i.d. belief systems. We will see examples of different families of positively correlated and c.i.i.d. belief systems in Section 3.4.

### 3.3 Scoring Rules

Scoring rules [You75] are defined in the following way. Fix a non-decreasing sequence of real numbers, $s_1 \geq s_2 \geq \ldots \geq s_m$, such that $s_1 > s_m$. If a voter ranks an alternative $x$ at position $j$ then $x$ gets a score of $s_j$. We will write $sc(j, P_i)$ to denote the score of an alternative $j$ according to the preference $P_i$. The score of an alternative is the sum of the scores received from all the voters. The alternative with the highest score is chosen as the outcome of the election. In case there is a tie, a winning alternative is selected according to some tie-breaking rule. We insist that $s_1 > s_m$, otherwise if $s_1 = s_m$, every alternative receives the same score and the resulting social choice function just depends on the tie-breaking rule. Some popular scoring rules are:
Plurality \((1, 0, 0, \ldots, 0)\), Borda \((m - 1, m - 2, \ldots, 1, 0)\) and Veto \((1, 1, 1, \ldots, 1, 0)\).

Baharad and Neeman [BN02] prove that the probability that any scoring rule is not EPIC goes to zero at rate of \(1/\sqrt{n}\) with the number of agents \(n\). They consider the following setting: (1) each voter is equally likely to have any preference order, i.e. the marginal distribution is uniform, and (2) the preferences of the voters are locally correlated, i.e. the voters can be numbered in a sequence such that the further they are apart, the more independent their preferences become. In an independent work, Slinko [Sli02] shows that the number of manipulable preference profiles (profiles where some agent can benefit by deviating unilaterally) goes to zero at a rate of \(1/\sqrt{n}\) for any scoring rule. This result also proves the same rate of convergence to EPIC for uniform i.i.d. preferences. We prove that when the preferences of the voters are KT-correlated and c.i.i.d., the probability that any given scoring rule is not EPIC goes to zero exponentially fast. Our result strengthens what is known about the asymptotic non-manipulability of scoring rules.

### 3.3.1 EPIC Convergence for KT-correlation

Suppose \(\{\mu_i\}_{i=1}^n\) is a c.i.i.d. belief system, that is, for all \(i\), we have \(\mu_i(P_\cdot|P_i) = \prod_{j \neq i} v(P_j|P_i)\). Consider any preference \(P_i \in \mathcal{P}\) and any two alternatives \(a\) and \(b\) such that \(aP_i b\). Let

\[
\mu_{a,b}(P_i) = \mathbb{E}_{P \sim v(\cdot|P_i)} [sc(a, P_i) - sc(b, P_i)]
\]  

(3.4)

denote the expected difference of scores between alternatives \(a\) and \(b\) for a random preference order of some other agent, this preference drawn according to the conditional distribution \(v(\cdot|P_i)\). Let \(\sigma^2_{a,b}(P_i)\) denote the variance of the difference in score between \(a\) and \(b\). We now state our main result.

**Theorem 3.1.** Suppose a belief system \(\{\mu_i\}_{i=1}^n\) is c.i.i.d. and KT-correlated. Then the probability that any given scoring rule is not EPIC w.r.t. \(\{\mu_i\}_{i=1}^n\) goes to zero at a rate proportional to \(O(e^{-cn})\), for some constant \(c > 0\).

A useful lemma establishes that \(\mu_{a,b}(P_i) > 0\), for any preference \(P_i\) and alternatives \(a, b\) such that \(aP_ib\).
Lemma 3.1. Suppose the belief system \( \{\mu_i\}_{i=1}^{n} \) is c.i.i.d. and KT-correlated. Consider a preference ordering \( P_i \), and alternatives \( a \) and \( b \) such that \( aP_i b \). Then \( \mu_{a,b}(P_i) > 0 \).

Proof. Select two preference orderings \( P_j \) and \( P_j' \) such that \( d(P_j, P_i) < d(P_j', P_i) \). Now consider the following two preference profiles for voters other than \( i \):

1. \( P_{-i}^1 = \langle P_i, \ldots, P_i, P_j, P_i, \ldots, P_i \rangle \)
2. \( P_{-i}^2 = \langle P_i, \ldots, P_i, P_j', P_i, \ldots, P_i \rangle \)

Then \( D(P_{-i}^1, P_i) = d(P_j, P_i) < d(P_j', P_i) = D(P_{-i}^2, P_i) \). Since \( \{\mu_i\}_{i=1}^{n} \) is KT-correlated, we have \( \mu_i(P_{-i}^1|P_i) > \mu_i(P_{-i}^2|P_i) \). Furthermore, \( \{\mu_i\}_{i=1}^{n} \) is c.i.i.d., therefore there exists a function \( \nu \) such that \( \mu_i(P_{-i}^1|P_i) = \prod_{j \neq i} \nu(P_j|P_i) \). \( \mu_i(P_{-i}^1|P_i) > \mu(P_{-i}^2|P_i) \) implies

\[
\nu(P_j|P_i)^{n-2} \nu(P_{j'}|P_i) > \nu(P_j|P_i)^{n-2} \nu(P_{j'}|P_i) \text{.}
\]

Therefore, we have \( d(P_j, P_i) < d(P_j', P_i) \) implies \( \nu(P_j|P_i) > \nu(P_{j'}|P_i) \). Now, partition the set of all preference orderings \( \mathcal{P} \) into \( \mathcal{P}_{a>b} \) and \( \mathcal{P}_{b>a} \). Every preference ordering \( P \) in \( \mathcal{P}_{a>b} \) ranks \( a \) above \( b \) and vice versa for \( \mathcal{P}_{b>a} \). Note that there exists a one-to-one mapping \( f : \mathcal{P}_{a>b} \rightarrow \mathcal{P}_{b>a} \), namely \( f(P) \) be the same as \( P \) except positions of alternatives \( a \) and \( b \) exchanged. Then

\[
\mu_{a,b}(P_i) = \sum_{P \in \mathcal{P}} (sc(a, P) - sc(b, P)) \nu(P|P_i) = \sum_{P \in \mathcal{P}_{a>b}} (sc(a, P) - sc(b, P))(\nu(P|P_i) - \nu(f(P)|P_i)) \text{.}
\]

Now for all \( P \in \mathcal{P}_{a>b} \), \( d(P, P_i) < d(f(P), P_i) \) and therefore \( \nu(P|P_i) > \nu(f(P)|P_i) \). Moreover there exists at least one \( P \) such that \( sc(a, P) > sc(b, P) \) (e.g. when \( P \) places \( a \) at top and \( b \) at bottom). This proves that \( \mu_{a,b}(P_i) > 0 \). \( \Box \)

We will also use the Chernoff-Hoeffding inequality for bounded random variables.

Lemma 3.2. (Theorem 2,[Hoe63]) Let \( X_1, X_2, \ldots, X_n \) be independent and for all \( i, a_i \leq X_i \leq b_i \). Let \( \bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i \) and \( \mu = E\left[\frac{1}{n} \sum_{i=1}^{n} X_i\right] \). Then

\[
\Pr\left[\bar{X}_n \leq \mu - t\right] \leq \exp\left\{-\frac{2t^2}{\sum_{i=1}^{n} (b_i - a_i)^2} \right\}
\]
Proof. (of Theorem 3.1.) Suppose voter 1 observes a preference order \( x_1 \). Let \( X_i \) be the random variable corresponding to the preference ordering seen by agent \( i \). Conditioned on the event \( X_1 = x_1 \), voter 1 believes that the random variables \( X_2, X_3, \ldots \) are i.i.d. with distribution \( \nu(\cdot|x_1) \).

Fix any two alternatives \( a \) and \( b \) such that \( a \prec x_1 \prec b \). We show that the probability that agent 1 can improve the ordering of \( a \) vs \( b \) in the social ranking, through making some misreport of his preference order, falls exponentially quickly in the number of agents \( n \). For \( i = 2, 3, \ldots, \) let \( Z_{a,b}^i = sc(a, X_i) - sc(b, X_i) \) be the difference of scores between alternatives \( a \) and \( b \) assigned by voter \( i \)'s true preference \( X_i \). Since \( X_2, X_3, \ldots \) are i.i.d., so are \( Z_{a,b}^2, Z_{a,b}^3, \ldots \). As introduced earlier, \( E[Z_{a,b}^i] = \mu_{a,b}(x_1) \) and \( \text{Var}(Z_{a,b}^i) = \sigma_{a,b}^2(x_1) \). Define \( Z_{a,b}^n = \sum_{i=2}^{n+1} Z_{a,b}^i \) (where voters 2 through \( n+1 \) each have c.i.i.d. preferences).

Now voter 1 wants to manipulate and report a preference ordering different from \( x_1 \) only if \( s_m - s_1 \leq Z_{a,b}^n \leq sc(b, x_1) - sc(a, x_1) \). To see this, first suppose \( Z_{a,b}^n < s_m - s_1 < 0 \). Then even by placing \( a \) at top and \( b \) at bottom, voter 1 can increase the difference in scores between \( a \) and \( b \) by at most \( s_1 - s_m \) and this still fails to cause \( a \) to rank higher than \( b \). So, in this case, voter 1 is happy to report \( x_1 \).

On the other hand, suppose \( Z_{a,b}^n > sc(b, x_1) - sc(a, x_1) \). If 1 reports truthfully then the net difference of scores between \( a \) and \( b \) is \( Z_{a,b}^n + sc(a, x_1) - sc(b, x_1) > 0 \) and \( a \) and \( b \) are already ordered according to 1's preferences.

Let \( \Delta_{a,b} = sc(b, x_1) - sc(a, x_1) \). Note that, \( \Delta_{a,b} \leq 0 \) since \( a \prec x_1 \prec b \). Then the probability of manipulation by voter 1 is bounded by \( \Pr[s_m - s_1 \leq Z_{a,b}^n \leq \Delta_{a,b}] \), which we bound using Chernoff-Hoeffding inequality (Lemma 3.2) as follows:

\[
\Pr[s_m - s_1 \leq Z_{a,b}^n \leq \Delta_{a,b}] \leq \Pr[Z_{a,b}^n \leq \Delta_{a,b}]
= \Pr\left[\frac{1}{n} \sum_{i=2}^{n+1} Z_{a,b}^i \leq \frac{1}{n} \Delta_{a,b}\right] = \Pr\left[\frac{1}{n} \sum_{i=2}^{n+1} Z_{a,b}^i \leq \mu_{a,b}(x_1) - t\right].
\]

Where \( t = \mu_{a,b}(x_1) - \frac{1}{n} \Delta_{a,b} \). Since \( Z_{a,b}^i \) is the difference of scores between alternatives \( a \) and \( b \) according to \( X_i \), we have \( Z_{a,b}^i \in [s_m - s_1, s_1 - s_m] \). Now applying Chernoff-Hoeffding inequality
Figure 3.1: The blue regions determine the probability of manipulation. For a KT-correlated belief (3.1a) we use the tail probability and for any general c.i.i.d. belief (3.1b) we use the area of the surrounding rectangle.

we get an upper bound of

$$\exp \left\{ -\frac{2n^2\gamma^2}{4n(s_1-s_m)^2} \right\} = \exp \left\{ -\frac{n}{2} \left( \frac{\mu_{a,b}(x_1)}{s_1-s_m} \right)^2 + \frac{\mu_{a,b}(x_1)\Delta_{a,b}}{(s_1-s_m)^2} - \frac{1}{2n} \left( \frac{\Delta_{a,b}}{s_1-s_m} \right)^2 \right\}. $$

As $n$ goes to infinity, $e^{-\gamma/n}$ goes to 1 for any constant $\gamma > 0$. Therefore, for large enough $n$, the probability of manipulation is $O(e^{-c_{a,b}(x_1)n})$ where $c_{a,b}(x_1) = \frac{1}{2} \left( \frac{\mu_{a,b}(x_1)}{s_1-s_m} \right)^2$. Since the preference ordering $x_1$ and the alternatives $a$ and $b$ can arbitrary, using a union bound we get the actual probability of manipulation is $e^{-cn}$ where the constant $c$ takes the worst-case, and

$$c = \min_{x \in \mathcal{P}} \min_{x_1 \neq a, b} \frac{1}{2} \left( \frac{\mu_{a,b}(x_1)}{s_1-s_m} \right)^2. $$

Lemma 3.1 proves that for any KT-correlated and c.i.i.d. belief system, $\mu_{a,b}(x_1) > 0$ for any $x_1$ and any $a$ and $b$ such that $a x_1 b$. This proves that $c > 0$, and finishes the proof.

3.3.2 EPIC Convergence for TS-correlation

Lemma 3.1 need not be true for a TS-correlated belief system. Here is a counter-example. Consider a situation with 2 voters, 3 alternatives, and Borda scoring rule $(2, 1, 0)$. Suppose voter 1’s preference ordering is $1 P_1 2 P_1 3$ (in short 123). We now construct a TS-correlated belief system for which $\mu_{1,2}(123) = E_{X_2 \sim v(123)} \left[ Z_{1,2}^2 \right] < 0$.

Any TS-correlated belief system needs to satisfy the following system of inequalities (we
drop the conditioning on preference ordering 123 for notational convenience):

\[
\begin{align*}
\mu_1(123) + \mu_1(132) & > \mu_1(213) + \mu_1(231) \\
\mu_1(123) + \mu_1(132) & > \mu_1(312) + \mu_1(321) \\
\mu_1(123) + \mu_1(213) & > \mu_1(132) + \mu_1(312) \\
\mu_1(123) + \mu_1(213) & > \mu_1(231) + \mu_1(321)
\end{align*}
\]

Let us choose the following distribution: \( \mu_1(123) = 1 - 6\varepsilon, \mu_1(132) = \mu_1(213) = \mu_1(312) = \mu(321) = \varepsilon, \mu_1(231) = 2\varepsilon \). It can be easily verified that \( \mu_1 \) satisfies the inequalities above as long as \( \varepsilon < 1/8 \). Now \( \mu_{1,2}(123) = 1 - 9\varepsilon < 0 \) if \( \varepsilon > 1/9 \). To get a counter-example, one can choose any \( \varepsilon \in (1/9, 1/8) \).

However, we now prove that the rate of convergence is not worse than \( O \left( 1/\sqrt{n} \right) \) for any c.i.i.d. belief system. Our proof uses the Berry-Esseen theorem which quantifies the rate of convergence of the central limit theorem.

**Lemma 3.3. (Berry-Esseen)** Let \( X_1, X_2, \ldots, X_n \) be i.i.d. with \( E[X_i] = 0, E[X_i^2] = \sigma^2, E[X_i^3] = \rho < \infty \). If \( F_n(x) \) is the distribution function of \( (X_1 + \ldots + X_n)/\sigma \sqrt{n} \) and \( \Phi(x) \) is the distribution function of standard normal random variable then, \( \forall x \in \mathbb{R} \),

\[
|F_n(x) - \Phi(x)| \leq \frac{3\rho}{\sigma^3 \sqrt{n}}.
\]

See [Dur10] (Theorem 3.4.9) for a proof.

**Theorem 3.2.** Suppose a belief system \( \{\mu_i\}_{i=1}^n \) is c.i.i.d. Then the probability that any given scoring rule is not EPIC w.r.t. \( \{\mu_i\}_{i=1}^n \) goes to zero at a rate proportional to \( O \left( 1/\sqrt{n} \right) \).

**Proof.** As we proved in Theorem 3.1, the probability of manipulation is bounded by

\[
\Pr \left[ s_m - s_1 \leq Z_{a,b}^n \leq \Delta_{a,b} \right].
\]

Now, we use the central limit theorem instead of Chernoff-Hoeffding inequality to bound the probability of manipulation. \( Z_{a,b}^2, Z_{a,b}^3, \ldots \) are iid with mean \( \mu_{a,b}(x_1) \) and variance \( \sigma_{a,b}^2(x_1) \). Therefore, by the central limit theorem

\[
\frac{Z_{a,b}^n - n\mu_{a,b}(x_1)}{\sigma_{a,b}(x_1)\sqrt{n}} \overset{d}{\to} \mathcal{N}(0,1).
\]

(3.6)
Let us write \( T_n = \frac{Z_{a,b}^n - n\mu_{a,b}(x_1)}{\sigma_{a,b}(x_1) \sqrt{n}} \). Then

\[
\Pr \left[ s_m - s_1 \leq Z_{a,b}^n \leq \Delta_{a,b} \right] = \Pr \left[ \frac{s_m - s_1 - n\mu_{a,b}(x_1)}{\sigma_{a,b}(x_1) \sqrt{n}} \leq T_n \leq \frac{\Delta_{a,b} - n\mu_{a,b}(x_1)}{\sigma_{a,b}(x_1) \sqrt{n}} \right] = F_{T_n} \left( \frac{\Delta_{a,b} - n\mu_{a,b}(x_1)}{\sigma_{a,b}(x_1) \sqrt{n}} \right) - F_{T_n} \left( \frac{s_m - s_1 - n\mu_{a,b}(x_1)}{\sigma_{a,b}(x_1) \sqrt{n}} \right)
\]

Here \( F_{T_n} \) is the distribution function of \( T_n \). Now we use the following relation:

\[
\forall l \leq u, \quad F_{T_n}(u) - F_{T_n}(l) \leq |F_{T_n}(u) - F_{T_n}(l)| \leq |F_{T_n}(u) - \Phi(u)| + |\Phi(u) - \Phi(l)| + |\Phi(l) - F_{T_n}(l)|.
\]

We can bound the first and the third term by the Berry-Esseen Theorem (Lemma 3.3) (since \( Z_{a,b}^i \) is a discrete distribution, its third absolute moment \( \rho_{a,b}(x_1) \) is finite) to get

\[
F_{T_n}(u) - F_{T_n}(l) \leq \frac{6\rho_{a,b}(x_1)}{\sigma_{a,b}(x_1)^2 \sqrt{n}} + \int_l^u \phi(t) dt.
\]  

(3.7)

Using the last relation, we can finally prove the following bound on probability of manipulation:

\[
\frac{6\rho_{a,b}(x_1)}{\sigma_{a,b}(x_1)^2 \sqrt{n}} + \int_l^u \phi(t) dt + 1 \geq e^{-t^2/2} dt.
\]

Since \( e^{-t^2/2} \leq 1 \), the last integral can be bounded above by \( \frac{1}{\sqrt{2\pi} \sigma_{a,b}(x_1) \sqrt{n}} \). This proves that the probability of manipulation is \( O\left(1/\sqrt{n}\right) \).

3.4 Rank-Order Models

Now we present some examples of belief systems that are c.i.i.d. and positively correlated. Our main source of examples is the rank-order models from machine learning.

Mallows Model

The Mallows [Mal57] model was originally defined with respect to a fixed (latent) preference ordering \( \sigma \) and a dispersion parameter \( r \in (0, 1] \). Let \( \pi \) be any preference ordering, then the
Mallows model specifies $Pr[\pi|\sigma] \propto e^{-d(\pi,\sigma)}$, i.e. the probability of observing the ordering $\pi$ decays exponentially with its Kendall-Tau distance from $\sigma$. We are not aware of any adaptation of the Mallows model for a conditional belief system. However any such adaptation should capture the idea that once a voter $i$ observes a preference ordering $P_i$, then she believes that the preference ordering of any other agent is a noisy version of $P_i$, with its probability decaying exponentially with the Kendall-Tau distance from $P_i$. This motivates the following belief system:

**Definition 3.5.** A belief system $\{\mu_i\}_{i=1}^n$ is a Conditional Mallows model if $\exists r \in (0,1]$ such that

$$\mu_i(P_{-i}|P_i) \propto r^{\sum_{j \neq i} d(P_j,P_i)}. \quad (3.8)$$

**Theorem 3.3.** Every Conditional Mallows model is c.i.i.d. and KT-correlated.

The proof is trivial since if $\{\mu_i\}_{i=1}^n$ is a conditional belief system then $v(P_j|P_i) \propto r^{d(P_j,P_i)}$. As a consequence of Theorem 3.3, if $\{\mu_i\}_{i=1}^n$ is a conditional Mallows model, then any scoring rule becomes EPIC at a rate proportional to $O(e^{-cn})$ where $c$ is as defined in Equation (3.5).

**Plackett-Luce Model**

The Plackett-Luce model [Pla75; Luc59] on a set of alternatives $\{1, \ldots, m\}$ has $m$ parameters: $\gamma_j > 0$ for each alternative $j$, such that $\sum_{j=1}^m \gamma_j = 1$. For a permutation $\pi$ of the $m$ alternatives, let $\pi[k]$ denote the alternative placed at position $k$. The probability of permutation $\pi$ is given as:

$$Pr[\pi|\{\gamma_j\}_{j=1}^m] = \frac{\gamma_{\pi[1]}}{\gamma_{\pi[1]} + \ldots + \gamma_{\pi[m]}} \cdot \frac{\gamma_{\pi[2]}}{\gamma_{\pi[2]} + \ldots + \gamma_{\pi[m]}} \cdots \frac{\gamma_{\pi[m-1]}}{\gamma_{\pi[m-1]} + \gamma_{\pi[m]}} \cdot \frac{\gamma_{\pi[m]}}{\gamma_{\pi[m]}}. \quad (3.9)$$

We propose the following belief system $\{\mu_i\}_{i=1}^n$ based on the Plackett-Luce Model.

**Definition 3.6.** A belief system $\{\mu_i\}_{i=1}^n$ is a Conditional Plackett-Luce model if there exists $m$ parameters $\gamma_1 > \gamma_2 > \ldots > \gamma_m$ and $\sum_{j=1}^m \gamma_j = 1$ such that $\mu_i(P_{-i}|P_i) = \prod_{j \neq i} v(P_j|P_i)$, where

$$v(P_j|P_i) = Pr[P_j|\{\gamma_{\pi[j]}\}_{i=1}^m]. \quad (3.10)$$
To compute \( v(P_i|P_i) \), we first permute the \( m \) parameters \( \{\gamma_i\}_{i=1}^m \) according to \( P_i \) so that 
\( \gamma_{P_i[1]} > \gamma_{P_i[2]} > \ldots > \gamma_{P_i[m']} \) and then use the standard Plackett-Luce model.

The Conditional Plackett-Luce model is not KT-correlated. Suppose there are two voters
\( (n = 2) \) and four alternatives \( (m = 4) \). We will write 1234 to denote the preference ordering
1 \( P_1 \) 2 \( P_1 \) 3 \( P_1 \) 4. We now construct a set of parameters \( \{\gamma_i\}_{i=1}^4 \) such that
\( \mu_1(1342 | 1234) > \mu_1(2134 | 1234) \). Since
\( d(1342, 1234) = 2 > 1 = d(2134, 1234) \), \( \mu_1(\cdot) \) is not KT-correlated. Let
\( \gamma_1 = 1 - 6\varepsilon, \gamma_2 = 3\varepsilon, \gamma_3 = 2\varepsilon, \gamma_4 = \varepsilon \). For, \( \gamma_1 > \gamma_2 \), we require
\( 1 - 6\varepsilon > 3\varepsilon \) or \( \varepsilon < 1/9 \). Now,

\[
\mu_1(1342 | 1234) > \mu_1(2134 | 1234) \\
\iff \frac{\gamma_1 \gamma_2 \gamma_3 \gamma_4}{\gamma_3 + \gamma_4 + \gamma_2} > \frac{\gamma_1 \gamma_2 \gamma_3 \gamma_4}{\gamma_1 + \gamma_3 + \gamma_4 + \gamma_2} \\
\iff 1/12 > 2\varepsilon/(1 - 3\varepsilon) \iff \varepsilon < 1/27.
\]

Therefore, as long as \( \varepsilon < 1/27 \), the belief system fails to be KT-correlated. However, we now
prove that the Conditional PL model is TS-correlated. We use the following lemma about the
likelihoods in the Conditional PL model.

**Lemma 3.4.** For all \( k, \) all \( 1 \leq k \leq m \), such that \( l_1, \ldots, l_k \) distinct, \( \sum_{\pi: \pi[1]=l_1, \ldots, \pi[k]=l_k} \operatorname{Pr}(\pi | \{\gamma_i\}_{i=1}^m) = \frac{\gamma_{l_1} \gamma_{l_2} \cdots \gamma_{l_k}}{1 - \gamma_{l_1} - \cdots - \gamma_{l_k}} \).


**Theorem 3.4.** Every Conditional PL Model is TS-correlated.

**Proof.** Let \( \{\mu_i\}_{i=1}^n \) be a Conditional PL belief system with parameters \( \{\gamma_i\}_{i=1}^m \) where \( \gamma_1 > \gamma_2 > \ldots > \gamma_m > 0 \) and \( \sum_{i=1}^m \gamma_i = 1 \). For any set \( T \) of size \( k \) \( (1 \leq k \leq m - 1) \),

\[
\sum_{\substack{(P_i, T) : B_k(P) = T \\forall j \neq i}} \mu_i(P_{-i} | P_i) = \left( \sum_{\{P : B_k(P) = T\}} v(P|P_i) \right)^{n-1}.
\]

This follows from two observations : (1) \( \{\mu_i\}_{i=1}^n \) is c.i.i.d., so \( \mu_i(P_{-i} | P_i) = \prod_{j \neq i} v(P_j | P_i) \), and (2)
Using the multinomial expansion, \( \left( \sum_{\{P : B_k(P) = T\}} v(P|P_i) \right)^{n-1} \) can be written as

\[
\sum_{\forall j \neq i P_j \in \{P : B_k(P) = T\}} \prod_{j \neq i} v(P_j | P_i) \quad (3.11)
\]

80
Now, from the definition of a TS-correlated belief system (3.2), it is enough to show that 
\[ \sum_{P : B_k(P) = B_k(P_1)} v(P | P_1) > \sum_{P : B_k(P) = T} v(P | P_1) \]. Without loss of generality, we can assume that \( P_i \) places \( i \)-th alternative at position \( i \). Let \( T = \{t_1, \ldots, t_k\} \) such that \( \gamma_{t_1} > \gamma_{t_2} > \ldots > \gamma_{t_k} \). We will write \( S([k]) \) to denote the set of all permutations over the set \( \{1, \ldots, k\} \). We have:

\[
\begin{aligned}
&\sum_{P : B_k(P) = B_k(P_1)} v(P | P_1) > \sum_{P : B_k(P) = T} v(P | P_1) \\
\Leftrightarrow & \sum_{\pi \in S([k])} \sum_{P : P[j] = \pi[j], 1 \leq j \leq k} \text{Pr}[P | \{\gamma_{t}\}_{t=1}^{n}] > \sum_{\pi \in S([k])} \sum_{P : P[j] = \{\pi[j]\}_{j=1}^{k}, 1 \leq j \leq k} \text{Pr}[P | \{\gamma_{t}\}_{t=1}^{n}]
\end{aligned}
\]

\{using Lemma 3.4\}

\[
\begin{aligned}
\Leftrightarrow & \sum_{\pi \in S([k])} \left\{ \frac{\gamma_{\pi[1]} \gamma_{\pi[2]} \cdots \gamma_{\pi[k]}}{1 - \gamma_{\pi[1]} - \cdots - \gamma_{\pi[k-1]}} - \gamma_{t_{\pi[1]}[1]} \cdots \gamma_{t_{\pi[k]}[k]} \right\} > 0
\end{aligned}
\]

Since \( T \) is different from \( B_k(P) \), the set of top-\( k \) alternatives in preference ordering \( P \), we have for any \( \pi \in S([k]) \), \( \gamma_{\pi[j]} \geq \gamma_{t_{\pi[j]}[j]} \) for \( j = 1, \ldots, k \) and \( \exists t \) \( \gamma_{\pi[t]} > \gamma_{t_{\pi[t]}[t]} \). This implies that for all \( l \geq t \)

\[
\frac{\gamma_{\pi[l]} \gamma_{\pi[l+1]} \cdots \gamma_{\pi[l+k-1]}}{1 - \gamma_{\pi[l]} - \cdots - \gamma_{\pi[l+k-1]}} > \frac{\gamma_{t_{\pi[l]}[l]} \cdots \gamma_{t_{\pi[l+k]}[l+k-1]}}{1 - \gamma_{t_{\pi[l]}[l]} - \cdots - \gamma_{t_{\pi[l+k]}[l+k-1]}}
\]

which is sufficient to guarantee (3.13).

Since the Conditional PL model is not KT-correlated, we cannot use Theorem 3.1 to claim exponential EPIC convergence of any scoring rule under this model. However, the next theorem shows that the convergence is indeed exponential, and that this is true for any scoring rule.

**Theorem 3.5.** The probability that any scoring rule is not EPIC w.r.t. a Conditional Plackett-Luce model goes to zero at a rate proportional to \( O(e^{-cn}) \), for some constant \( c > 0 \).

**Proof.** Let the belief system \( \{\mu_i\}_{i=1}^{m} \) be a Conditional Plackett-Luce model with parameters \( \{\gamma_i\}_{i=1}^{m} \) where \( \gamma_1 > \gamma_2 > \ldots > \gamma_m > 0 \) and \( \sum_{i=1}^{m} \gamma_i = 1 \).

Consider a voter \( i \) with preference ordering \( P_i \) and two alternatives \( a \) and \( b \) such that \( aP_i b \). As we have argued in Theorem 3.1, it is sufficient to show that \( \mu_{a,b}(P_i) > 0 \). Without loss of
We have provided a positive result for incentive-aligned social choice in the presence of positions question of closing the gap between exponential convergence and 1 question for future work is whether exponential, EPIC convergence holds for correlated quickly in the number of voters. We have instantiated the general framework to conditional correlated beliefs. In terms of Kendall-Tau distance, we show that all scoring rules become EPIC exponentially correlation between voter preferences. When the beliefs of agents are positively correlated and $P_i$ places the \( i \)-th alternative at position $i$. We have:

\[
\mu_{a,b}(P_i) = \mathbb{E}_{X_j \sim v_i(P_i)} \left[ sc(a, X_j) - sc(b, X_j) \right]
\]

\[
= \sum_{i_1 < i_2} (s_{i_1} - s_{i_2}) \left\{ \Pr \left[ X_j[i_1] = a, X_j[i_2] = b \mid \{\gamma_l\}_{l=1}^m \right] - \Pr \left[ X_j[i_1] = b, X_j[i_2] = a \mid \{\gamma_l\}_{l=1}^m \right] \right\}
\]

Therefore, it is enough to show that for any $l_2 > l_1$, $\Pr \left[ X_j[i_1] = a, X_j[i_2] = b \mid \{\gamma_l\}_{l=1}^m \right] > \Pr \left[ X_j[i_1] = b, X_j[i_2] = a \mid \{\gamma_l\}_{l=1}^m \right]$. Now, suppose alternatives $a$ and $b$ are placed respectively at positions $r$ and $s$ of $P_i$. Therefore, $\gamma_r > \gamma_s$. Then by Lemma 3.4 we have,

\[
\Pr \left[ X_j[i_1] = a, X_j[i_2] = b \mid \{\gamma_l\}_{l=1}^m \right] = \sum_{i_1, \ldots, i_{l_2-2} \in [m]} \sum_{(a,b) \in S(\{i_1, \ldots, i_{l_2-2}\})} f(\gamma_{r|\gamma_{i_1}}, \ldots, \gamma_{r|\gamma_{i_{l_1-1}}}, \gamma_r, \gamma_{r|\gamma_{i_{l_1}}}, \ldots, \gamma_{r|\gamma_{i_{l_2-2}}}, \gamma_s) \text{ where}
\]

\[
f(\lambda_1, \ldots, \lambda_k) = \frac{\lambda_2}{1 - \lambda_1} \ldots \frac{\lambda_k}{1 - \lambda_1 - \ldots - \lambda_{k-1}}.
\]

It is easy to see that $\gamma_r > \gamma_s$ implies $f(\gamma_{r|\gamma_{i_1}}, \ldots, \gamma_{r|\gamma_{i_{l_1-1}}}, \gamma_r, \gamma_{r|\gamma_{i_{l_1}}}, \ldots, \gamma_{r|\gamma_{i_{l_2-2}}}, \gamma_s) > f(\gamma_{r|\gamma_{i_1}}, \ldots, \gamma_{r|\gamma_{i_{l_1-1}}}, \gamma_s, \gamma_{r|\gamma_{i_{l_1}}}, \ldots, \gamma_{r|\gamma_{i_{l_2-2}}}, \gamma_r)$, as $\gamma_r, \gamma_s$ cancels out from both sides, and for all $j$ such that $l_1 \leq j \leq l_2 - 2$, we have $(1 - \gamma_{r|\gamma_{i_1}} - \ldots - \gamma_{r|\gamma_{i_{l_1}}}, \gamma_r)^{-1} > (1 - \gamma_{r|\gamma_{i_1}} - \ldots - \gamma_{r|\gamma_{i_{l_1}}}, \gamma_s)^{-1}$.

\[\square\]

### 3.5 Conclusion

We have provided a positive result for incentive-aligned social choice in the presence of correlation between voter preferences. When the beliefs of agents are positively correlated in terms of Kendall-Tau distance, we show that all scoring rules become EPIC exponentially quickly in the number of voters. We have instantiated the general framework to conditional variations on the popular Mallows and Plackett-Luce models for rank aggregation. One question for future work is whether exponential, EPIC convergence holds for correlated distributions that are not conditionally independent and identical. We also leave open the question of closing the gap between exponential convergence and $1/\sqrt{n}$ for the class of TS-correlated beliefs.
Chapter 4

Limits and Possibilities of Implicit Utilitarian Voting

4.1 Introduction

Social choice theory studies the aggregation of individual preferences into collective decisions. While its origins can be traced back to the contributions of Condorcet [Con85] and others in the 18th Century, the field was founded in its modern form in the 20th Century. With his famous impossibility result, Arrow [Arr51] pioneered the axiomatic approach to voting, in which voting rules that aggregate ranked preferences of individuals are compared qualitatively based on the axiomatic desiderata they satisfy or violate. This approach underlies most of the work on voting in social choice theory [see, e.g., ASS10; Sen86].

By contrast, research in computational social choice [Bra+16] has put more emphasis on quantitative evaluation of voting rules. In particular, Procaccia and Rosenschein [PR06] introduced the implicit utilitarian voting framework, in which it is assumed that individuals (a.k.a. voters) have underlying cardinal utilities for the different alternatives, and express ranked preferences that are consistent with their utilities. The goal is to choose an alternative that maximizes (utilitarian) social welfare — the sum of utilities — by relying on the reported rankings as a proxy for the latent utilities. Specifically, voting rules are compared by their distortion, which is the worst-case ratio of the maximum social welfare to the social welfare of
the alternative they choose. The implicit utilitarian voting approach has received significant attention in the past decade [CP11; Bou+15; Ben+17; Ans+18; AP17; GAX17; GKM17; CDK18; BPQ19; Car+17; Bor+19; GLS19; FFG16; BG18a; BDG18], and voting rules based on it have been deployed on the online voting platform robovote.org.

Benadè et al. [Ben+17] observe that implicit utilitarian voting has another advantage: it allows comparing not only voting rules that aggregate ranked preferences, but also voting rules that aggregate other types of ballots, which they refer to as *input formats*. They further argue that we can associate each input format with the best rule for aggregating votes in that format, and ultimately compare the input formats themselves based on the lowest distortion they make possible. They also introduce a new input format, *threshold approval*, whereby each voter is asked to report whether her utility for each alternative is above or below a given threshold; this input format allows achieving logarithmic distortion.

The results of Benadè et al. [Ben+17] beg the question: why should we set only a single threshold? What if we set two thresholds and ask each voter to report whether her utility for each alternative is below the lower threshold, between the two thresholds, or above the higher threshold? What if we set five thresholds? Or a million for that matter? Intuitively there is a tradeoff between the number of thresholds and the distortion that can be achieved. However, perhaps adding thresholds is not the most efficient way to drive down distortion; there may be other input formats that encapsulate more useful information. (Spoiler alert: this is indeed the case.)

Our goal in this paper is to characterize the *optimal* tradeoff between elicitation and distortion: as we elicit more information from voters about their utilities, we should be able to achieve lower distortion. But exactly how low? To answer this question, we need a precise way to reason about the complexity of vote elicitation. We use the nomenclature of communication complexity [KN96], and, in particular, examine the number of bits needed to report a vote. Note that this is simply the logarithm of the number of possible votes that a voter can provide in a given input format. Hence, plurality votes that ask a voter to report which of the $m$ alternatives is her top choice contain $\log m$ bits of information, while ranked preferences that
ask a voter to rank all \( m \) alternatives contain \( \log m! = \Theta(m \log m) \) bits of information.\(^1\) Our main research question is this:

**For any \( k \), given a budget of at most \( k \) bits per vote, what is the minimum distortion any voting rule can achieve?**

### 4.1.1 Our Results

Before outlining our results, we describe our framework in a bit more detail. A voting rule \( f \) is composed of two parts. Its *elicitation rule* \( \Pi_f \) elicits information from voters about their utilities. Essentially, it chooses a (possibly randomized) mapping from utility functions to finitely many (say \( k \)) possible responses, and each voter uses this mapping to cast her vote. The communication complexity of \( f \), denoted \( C(f) \), is then \( E[\log k] \), where the expectation is due to random choices made by \( \Pi_f \). The *aggregation rule* \( \Gamma_f \) aggregates the votes cast by voters to choose a single alternative (possibly in randomized way). The distortion of \( f \), denoted \( \text{dist}(f) \), is the worst-case ratio of the maximum social welfare to the (expected) social welfare of this chosen alternative. The distortion is typically a function of the number of alternatives \( m \). Our goal is to study the tradeoff between \( C(f) \) and \( \text{dist}(f) \).

Figure 4.1 shows our results and positions them in the context of previous results. We note that any upper bound with deterministic elicitation (resp. aggregation) also serves as an upper bound with randomized elicitation (resp. aggregation), and the converse holds for lower bounds.

For deterministic elicitation, it is known that plurality achieves \( \Theta(m^2) \) distortion with deterministic aggregation and \( \log m \) communication complexity, and that it is trivial to achieve \( \Theta(m) \) distortion with randomized aggregation and zero communication complexity [Car+17]. Our lower bounds from Section 4.4 establish that these are the best possible asymptotic bounds with communication complexity at most \( \log m \). We show that these bounds do not hold for randomized elicitation by constructing a new voting rule in Section 4.3, RandSubset, which

---

\(^1\)Our use of the number of bits of information can be seen as a conceptual measure of cognitive burden. In many applications of voting, voters do not really communicate their votes electronically in bits. Hence, in our work, unlike in much of the work on communication complexity, the number of bits may not be an integer (however, \( 2 \) raised to the number of bits is always an integer). This distinction is crucial for some of our lower bounds.
uses randomized elicitation and achieves $o(m)$ distortion with communication complexity at most $\log m$.

We also propose a family of voting rules, PREFTHRESHOLD, which use deterministic elicitation and aggregation, and can achieve $d$ distortion with $O(m \log (d \log m)/d)$ communication complexity. In Section 4.5, we leverage tools from multi-party communication complexity to show that this result is nearly optimal: any voting rule with $d$ distortion must have $\Omega(m/d^2)$ communication complexity with deterministic elicitation and $\Omega(m/d^3)$ communication complexity with randomized elicitation. Note that our upper and lower bounds differ by a factor that is almost linear or almost quadratic in $d$, and sublogarithmic in $m$. This implies a surprising fact: when our goal is to achieve near-constant distortion, randomization cannot significantly help.

### 4.1.2 Related Work

There are two threads of research on implicit utilitarian voting. The first thread does not make any assumptions on utilities, other than that they are normalized [PR06; CP11; Bou+15; Ben+17; BPQ19; Car+17; BG18a; BDG18]. The second thread assumes that utilities are induced
by a metric [Ans+18; AP17; GAX17; GKM17; CDK18; GLS19; FFG16; Bor+19]; this structure generally enables lower distortion. Our approach is consistent with the former thread.

In addition to the work of Benadè et al. [Ben+17], discussed above, an especially relevant paper is that of Caragiannis and Procaccia [CP11]. Their goal is also to achieve low distortion while keeping the communication requirements low. To that end, they employ specific voting techniques such as approving a single alternative (like plurality) or approving a subset of alternatives (like approval voting) — these require $\log m$ and $m$ bits per voter, respectively — but use what they call an embedding to describe how voters translate their cardinal preferences into votes. For example, in the case of approving a single alternative, a natural randomized embedding is to ask each voter to approve each alternative with probability proportional to its utility; roughly speaking, this achieves distortion that goes to 1 as the number of voters goes to infinity. The key difference between the work of Caragiannis and Procaccia [CP11] and our work is that our design space is much larger: we simultaneously optimize both the embedding and the voting technique (together, these form our elicitation rule), as well as the aggregation rule.\(^2\) It is interesting to note that Caragiannis and Procaccia [CP11] draw their motivation from settings where the voters are software agents, which can actually compute their utilities for alternatives, and are endowed with algorithms that map utilities to votes (so the algorithms themselves can conceivably be quite intricate). Although we are more interested in settings where voters are people, it is worth keeping the software-agents-as-voters setting in mind too because our model and results are equally relevant to it.

Further afield, Conitzer and Sandholm [CS05] study the communication complexity of voting rules, in a fundamentally different sense from ours. They are interested in studying how much information about the voters’ ranked preferences has to be elicited in order to compute the outcome under a given voting rule. By contrast, we are interested in designing the voting rule, and the very way in which preferences are represented, in order to minimize distortion.

In addition, the voting rules we design, which lead to the best known communication-distortion tradeoffs, ask voters to report their approximate utility either for their top few

\(^2\)That said, in this work we focus only on deterministic embeddings. That is, we study elicitation rules in which voters deterministically translate their cardinal preferences into votes, and show that the foregoing result is impossible to achieve in this case. We discuss implications of randomized embeddings in Section 4.6.
choices or for a randomly chosen subset of alternatives. Related ideas have been explored previously [FO14] or in parallel [BS19] in the computational social choice literature, albeit in fundamentally different models.

Another loosely related line of work was initiated by Balcan and Harvey [BH18] and Badanidiyuru et al. [Bad+12]. Their goal is to sketch combinatorial valuation functions, that is, to encode such functions using a polynomial number of bits in a way that the value of each subset can be recovered approximately. We deal with much simpler valuation functions, but, on the other hand, are looking to achieve much lower communication complexity. We are also interested in how multiple such ‘sketches’ can be aggregated to achieve a socially desirable outcome. More generally, most of the work on sketching combinatorial valuation functions [BH18; Bad+12], optimizing combinatorial functions using access to a value oracle [BRS17; FMV11], or maximizing welfare in combinatorial auctions [MSV08] assumes input which consists of real numbers; their focus is on using only polynomially many (instead of exponentially many) real numbers. By contrast, in our framework, asking for even a single real number leads to infinite communication complexity.

4.2 Model

For \( k \in \mathbb{N} \), define \( [k] = \{1, \ldots, k\} \). Let \( x \sim D \) denote that random variable \( x \) has distribution \( D \). Let \( \log \) denote the logarithm to base 2, and \( \ln \) denote the logarithm to base \( e \).

There is a set of alternatives \( A \) with \( |A| = m \), and a set of voters \( N = [n] \). Each voter \( i \in N \) is endowed with a valuation \( v_i : A \to \mathbb{R}_+ \), where \( v_i(a) \geq 0 \) represents the value of voter \( i \) for alternative \( a \). Equivalently, we view \( v_i \in \mathbb{R}_+^m \) as a vector which contains the voter’s value for each alternative. We slightly abuse notation and let \( v_i(S) = \sum_{a \in S} v_i(a) \) for \( S \subseteq A \). Collectively, voter valuations are denoted \( \vec{v} = (v_1, \ldots, v_n) \). Given \( \vec{v} \), the (utilitarian) social welfare of an alternative \( a \) is \( sw(a, \vec{v}) = \sum_{i \in N} v_i(a) \). Our goal is to elicit information about voter valuations and use it to find an alternative with as high social welfare as possible.

Valuations: We adopt the standard normalization assumption that \( \sum_{a \in A} v_i(a) = 1 \) for each \( i \in N \). This can be thought of as a “one voter, one vote” principle for cardinal valuations.
as it prevents voters from overshadowing other voters by expressing very high values.\textsuperscript{3} An equivalent interpretation is that we allow voter valuations that are not normalized but aim to maximize normalized (utilitarian) social welfare defined as
\[ \text{nsw}(a) = \sum_{i \in N} \left[ \frac{v_i(a)}{\sum_{b \in A} v_i(b)} \right]. \]
We stick to the former interpretation for the sake of simplicity. Let \( \Delta^m \) denote the \( m \)-simplex, i.e., the set of all vectors in \( \mathbb{R}_+^m \) whose coordinates sum to 1. Hence, we have that \( v_i \in \Delta^m \) for each \( i \in N \). Given such a vector \( v_i \in \Delta^m \), let \( \text{supp}(v_i) \subseteq A \) denote the support of \( v_i \), i.e., the set of alternatives \( a \) for which \( v_i(a) > 0 \).

**Query space:** If we ask voters to report their value for every alternative, we can easily find an alternative maximizing social welfare. However, reporting real-valued utilities requires infinitely many bits of communication. Our goal is to maximize social welfare subject to a finite bound on the number of bits of information that can be elicited from each voter.

Consider any interaction with voter \( i \) which elicits finitely many bits of information and in which the voter responds deterministically. In this interaction, the voter must provide one of finitely many (say \( k \)) possible responses. We say that this interaction elicits \( \log k \) bits of information.\textsuperscript{4} It effectively partitions \( \Delta^m \) into \( k \) compartments, where the compartment corresponding to each response is the set of all valuations which would result in the voter choosing that response. In other words, any interaction which elicits \( \log k \) bits of information is equivalent to a query which partitions \( \Delta^m \) into \( k \) compartments and asks the voter to pick the compartment in which her valuation belongs.

Let \( Q \) denote the set of all queries which partition \( \Delta^m \) into finitely many compartments. For a query \( q \in Q \), let \( k(q) \) denote the number of compartments created by \( q \); the number of bits elicited is \( \log k(q) \).\textsuperscript{5} This query space incorporates traditional elicitation methods studied in the social choice literature. For instance, plurality votes (which ask voters to report their favorite

\textsuperscript{3}Effectively, voters are only allowed to report the intensity of their relative preference for one alternative over another.

\textsuperscript{4}In case of a multi-round interaction, we can consider the string obtained by concatenating the voter’s responses in different rounds. This is equivalent to a single-round interaction in which the voter is asked to provide the entire string upfront, and the number of bits elicited is logarithm of the number of possible strings.

\textsuperscript{5}Note that the number of bits elicited may not be an integer, but \( 2 \) raised to the power of the number of bits must an integer. We could take the ceiling to enforce an integral number of bits, and this would only minimally increase elicitation, but some of our lower bounds are sensitive to this non-integral formulation.
alternative) use $m$ compartments, $k$-approval votes (which ask voters to report the set of their $k$ favorite alternatives) use $\binom{m}{k}$ compartments, threshold approval votes (which ask voters to approve alternatives for which their value is at least a given threshold) use $2^m$ compartments, and ranked votes (which ask voters to rank all alternatives) use $m!$ compartments.

**Voting Rule:** A voting rule (or simply, a rule) $f$ consists of two parts: an elicitation rule $\Pi_f$ and an aggregation rule $\Gamma_f$. The (randomized) elicitation rule $\Pi_f$ is a distribution over $Q$, according to which a query $q$ is sampled. Each voter $i$ provides a response $\rho_i$ to this query, depending on her valuation $v_i$. We say that the elicitation rule is deterministic if it has singleton support (i.e. it chooses a query deterministically). The (randomized) aggregation rule $\Gamma_f$ takes voter responses $\vec{\rho} = (\rho_1, \ldots, \rho_n)$ as input, and returns a distribution over alternatives. We say that the aggregation rule is deterministic if it always returns a distribution with singleton support. Slightly abusing notation, we denote by $f(\vec{v})$ the (randomized) alternative returned by $f$ when voter valuations are $\vec{v} = (v_1, \ldots, v_n)$. We measure the performance of $f$ via two metrics.

1. The communication complexity of $f$ for $m$ alternatives, denoted $C_m(f) = \mathbb{E}_{q \sim \Pi_f} [\log k(q)]$, is the expected number of bits of information elicited by $f$ from each voter. We drop $m$ from the superscript when its value is clear from the context.

2. The distortion of $f$ for $m$ alternatives, denoted $\text{dist}_m(f)$, is the worst-case ratio of the optimal social welfare to the expected social welfare achieved by $f$. Again, we drop $m$ from the superscript when its value is clear from the context. Formally,

$$\text{dist}(f) = \sup_{\vec{v} \in (\Delta^m)^n} \frac{\max_{a \in A} \text{sw}(a, \vec{v})}{\mathbb{E}_{\vec{\rho} \sim f(\vec{v})} [\text{sw}(\hat{a}, \vec{v})]}.$$ 

While it is desirable for a voting rule to have low communication complexity and low distortion, typically eliciting more information from voters enables achieving low distortion. Our goal is to understand the Pareto frontier of the tradeoff between communication complexity and distortion.
4.3 Upper Bounds

In this section, we derive upper bounds on the best distortion a voting rule can achieve given an upper bound on its communication complexity (equivalently, this gives an upper bound on the communication complexity required to achieve a given level of distortion). We study both deterministic and randomized elicitation, and our results are constructive.

Our main contributions in this section are two families of voting rules: $\text{PrefThreshold}$, which use deterministic elicitation and aggregation, and $\text{RandSubset}$, which convert a given voting rule into one which uses randomized elicitation.

4.3.1 Deterministic Elicitation, Deterministic Aggregation

We begin by designing voting rules which use deterministic elicitation and deterministic aggregation — the most practical combination. Caragiannis et al. [Car+17] show that plurality achieves $O(\sqrt{m})$ distortion with $\log m$ communication complexity, and even voting rules that elicit ranked preferences, and thus have $O(m \log m)$ communication complexity, cannot achieve asymptotically better distortion.

We propose a novel voting rule $\text{PrefThreshold}_{t,\ell}$, parametrized by $t \in [m]$ and $\ell \in \mathbb{N}$. It is presented as Algorithm 4.1. Its elicitation rule asks each voter to report the set of her $t$ most preferred alternatives, and for each alternative in this set, report her approximate value for it by picking one of $\ell + 1$ subintervals of $[0, 1]$. Note that for $t = 1$, we use $\ell$ subintervals of $[1/m, 1]$; this is valid because a voter’s value for her most favorite alternative must be at least $1/m$. The aggregation rule is also intuitive: it uses the approximate values to compute an estimated social welfare of each alternative, and picks an alternative with the highest estimated social welfare.

**Theorem 4.1.** For $t \in [m]\backslash\{1\}$ and $\ell \in \mathbb{N}$, we have

$$C(\text{PrefThreshold}_{t,\ell}) = \log \left( \binom{m}{t} \cdot (\ell + 1)^t \right) = \Theta \left( t \log \frac{m(\ell + 1)}{t} \right),$$

$$\text{dist}(\text{PrefThreshold}_{t,\ell}) = O \left( m^{1+2/\ell} / t \right).$$
Algorithm 4.1: PrefThreshold\(_{t,\ell}\), where \( t \in [m] \) and \( \ell \in \mathbb{N} \).

**Elicitation Rule:**

- If \( t > 1 \), create \( \ell + 1 \) buckets: \( B_0 = [0, 1/m^2] \) and \( B_p = (1/m^{2-2(p-1)/\ell}, 1/m^{2-2p/\ell}) \) for \( p \in [\ell] \).
- If \( t = 1 \), create \( \ell \) buckets: \( B_1 = [m^{-1}, m^{-1+1/\ell}] \) and \( B_p = (m^{-1+(p-1)/\ell}, m^{-1+p/\ell}) \) for \( p \in [\ell]\backslash{1} \).
- The query asks each voter \( i \) to identify set \( S_i^t \) of the \( t \) alternatives for which she has the highest value (breaking ties arbitrarily), and for each \( a \in S_i^t \), identify bucket index \( p_i,a \) such that \( v_i(a) \in B_{p_i,a} \).

**Aggregation Rule:**

- For each \( p \), let \( U_p \) denote the upper endpoint of bucket \( B_p \).
- For each voter \( i \in N \) and alternative \( a \in A \), define \( \hat{v}_i(a) = U_{p_i,a} \) if \( a \in S_i^t \) and \( \hat{v}_i(a) = 0 \) otherwise.
- For an alternative \( a \in A \), define the estimated social welfare as \( \hat{sw}(a) = \sum_{i \in N} \hat{v}_i(a) \).
- Return an alternative with the highest estimated social welfare, i.e., \( \hat{a} \in \arg \max_{a \in A} \hat{sw}(a) \).

**Communication Complexity:**

\[
C(\text{PrefThreshold}_{t,\ell}) = \begin{cases} 
\Theta \left( t \log \frac{m(\ell+1)}{t} \right), & \text{if } t > 1, \\
\log(m\ell), & \text{if } t = 1.
\end{cases}
\]

**Distortion:**

\[
dist(\text{PrefThreshold}_{t,\ell}) = \begin{cases} 
O(m^{1+2/t}/t), & \text{if } t > 1, \\
O(m^{1+1/\ell}), & \text{if } t = 1.
\end{cases}
\]

92
For $t = 1$ and $\ell \in \mathbb{N}$, we have

$$C(\text{PrefThreshold}_{1,\ell}) = \log(m\ell), \ dist(\text{PrefThreshold}_{1,\ell}) = O \left( m^{1+1/\ell} \right).$$

**Proof.** It is evident that the number of possible responses that a voter can provide under \text{PrefThreshold}_{t,\ell} is \(
\binom{m}{r} \cdot (\ell + 1)^t \) if $t > 1$, and $m \ell$ if $t = 1$. Taking the logarithm of this gives us the desired communication complexity.

We now establish the distortion of \text{PrefThreshold}_{t,\ell}. Let \( \bar{v} = (v_1, \ldots, v_n) \) be the underlying valuations of voters. For alternative \( a \in A \), recall that \( sw(a, \bar{v}) = \sum_{i \in N} v_i(a) \), and

$$s\hat{w}(a) = \sum_{i \in N} \hat{v}_i(a) = \sum_{i \in N,a \in S_i^t} \hat{v}_i(a) = \sum_{i \in N,a \in S_i^t} U_{p_i,a}.$$  

Let \( \hat{a} \in \arg \max_{a \in A} s\hat{w}(a) \) be the alternative chosen by the rule, and let \( a^* \in \arg \max_{a \in A} sw(a, \bar{v}) \) be an alternative maximizing social welfare.

We begin by finding an upper bound on \( sw(a^*, \bar{v}) \) in terms of \( s\hat{w}(\hat{a}) \).

$$sw(a^*, \bar{v}) = \sum_{i \in N} v_i(a^*) = \sum_{i \in N,a \in S_i^t} v_i(a^*) + \sum_{i \in N,a \neq S_i^t} v_i(a^*)$$

$$\leq \sum_{i \in N,a \neq S_i^t} v_i(a^*) + \sum_{i \in N,a \neq S_i^t} \left( \frac{\sum_{a \in S_i^t} v_i(a)}{t} \right)$$

$$\leq \sum_{i \in N,a \neq S_i^t} \hat{v}_i(a) + \sum_{i \in N,a \neq S_i^t} \frac{\sum_{a \in A \setminus \{a^*\}} \sum_{u \in A \setminus \{a, a^*\}} \hat{v}_i(a)}{t}$$

$$\leq s\hat{w}(a^*) + \frac{\sum_{a \in A \setminus \{a^*\}} s\hat{w}(a)}{t} \leq s\hat{w}(\hat{a}) + \frac{(m-1) \cdot s\hat{w}(\hat{a})}{t} = m + t - 1 \cdot s\hat{w}(\hat{a}), \quad (4.1)$$

where the third transition holds because for every \( i \in N \) with \( a^* \notin S_i^t \) and every \( a \in S_i^t \), we have \( v_i(a^*) \leq v_i(a) \); the fourth transition holds because for every \( i \in N \) and \( a \in S_i^t \), \( v_i(a) \leq \hat{v}_i(a) \); the fifth transition follows from the definition of \( s\hat{w} \); and the sixth transition holds because \( \hat{a} \) is a maximizer of \( s\hat{w} \).

We now establish the distortion for \( t > 1 \). The first step is to derive an upper bound on \( s\hat{w}(\hat{a}) \) in terms of \( sw(\hat{a}, \bar{v}) \). Our bucketing implies that for all \( i \in N \) and \( a \in S_i^t \), we have
\( v_i(a) \leq \hat{v}_i(a) \leq m^{2/\ell}v_i(a) + \frac{1}{m^2} \). Using this, we can derive the following.

\[
\hat{sw}(\hat{a}) = \sum_{i \in N : \hat{a} \in S_i^t} \hat{v}_i(\hat{a}) \leq \sum_{i \in N : \hat{a} \in S_i^t} \left( m^{2/\ell}v_i(\hat{a}) + \frac{1}{m^2} \right) \leq m^{2/\ell}sw(\hat{a}, \overline{v}) + \frac{n}{m^2}.
\] (4.2)

Next, we derive a lower bound on \( \hat{sw}(\hat{a}) \), which helps establish a lower bound on \( sw(\hat{a}, \overline{v}) \).

Note that for each voter \( i \in N \), \( \sum_{a \in S_i^t} v_i(a) \geq t/m \). Hence,

\[
\sum_{a \in A} \hat{sw}(a) - \sum_{i \in N} \sum_{a \in S_i^t} \hat{v}_i(a) - \sum_{i \in N} \sum_{a \in S_i^t} v_i(a) \geq \frac{n \cdot t}{m^2}.
\]

Because \( \hat{a} \) is a maximizer of \( \hat{sw} \), this yields \( \hat{sw}(\hat{a}) \geq n \cdot t/m^2 \). Substituting this into Equation (4.2), we get

\[
\frac{n}{m^2} + sw(\hat{a}, \overline{v}) \cdot m^{2/\ell} \geq \hat{sw}(\hat{a}) \geq \frac{n \cdot t}{m^2} \Rightarrow sw(\hat{a}, \overline{v}) \geq \frac{n \cdot (t-1)}{m^{2/\ell}} \cdot m^{-2/\ell} \geq \frac{n}{m \cdot m^{-2/\ell}}. \] (4.3)

Applying Equations (4.1), (4.2), and (4.3) in this order, we have

\[
\frac{sw(a^*, \overline{v})}{sw(\hat{a}, \overline{v})} \leq \frac{m + t - 1}{t} \cdot \frac{\hat{sw}(\hat{a})}{sw(\hat{a}, \overline{v})} \leq \frac{m + t - 1}{t} \cdot \left( \frac{m^{2/\ell}}{m^{2 \cdot sw(\hat{a}, \overline{v})}} + \frac{n}{m^{2 \cdot sw(\hat{a}, \overline{v})}} \right) \leq \frac{m + t - 1}{t} \cdot (m^{2/\ell} + m^{2/\ell}) = O(m^{1+2/\ell}/t).
\]

For \( t = 1 \), we have that for every \( i \in N \) and \( a \in S_i^t \), \( v_i(a) \leq \hat{v}_i(a) \leq m^{1/\ell}v_i(a) \). Hence, in Equation (4.2), the additive factor of \( n/m^2 \) disappears and the multiplicative factor of \( m^{2/\ell} \) becomes \( m^{1/\ell} \), yielding \( \hat{sw}(\hat{a}) \leq sw(\hat{a}, \overline{v}) \cdot m^{1/\ell} \). Similarly, Equation (4.3) becomes \( sw(\hat{a}, \overline{v}) \geq \frac{n}{m^2} \cdot m^{-1/\ell} \). Following the same line of proof as for the case of \( t > 1 \), we obtain

\[
\frac{sw(a^*, \overline{v})}{sw(\hat{a}, \overline{v})} \leq m \cdot \frac{\hat{sw}(\hat{a})}{sw(\hat{a}, \overline{v})} \leq m \cdot m^{1/\ell},
\]

which is the desired bound on distortion. \( \square \)

\( \text{PrefThreshold}_{t, \ell} \) offers a tradeoff between two parameters, \( t \) and \( \ell \). Increasing either parameter increases the communication complexity but reduces the distortion. We remark that there is no (asymptotic) benefit of choosing \( \ell > \log m \). This is because at \( \ell = \log m \), our upper bound on distortion reduces to \( O(m/t) \), and increasing \( \ell \) further does not change the bound asymptotically. Further, increasing \( \ell \) from 1 to \( \log m \) reduces the distortion by a factor of \( m^2 \) (if \( t = 1 \)), but does not (asymptotically) increase the communication complexity when
$t = \Omega(m/\log m)$ and only increases it by a sublogarithmic factor when $t = \omega(m/\log m)$. Hence, unless our goal is to make the communication complexity very small (e.g. with constant $t$ and $\ell$), it is best to set $\ell = \log m$. This gives rise to the following interesting choices of $t$ and $\ell$.

- **$t = 1, \ell = 1$**: $\text{PrefThreshold}_{1,1}$ coincides with plurality. Hence, $C(\text{PrefThreshold}_{1,1}) = \log m$ and $\text{dist}(\text{PrefThreshold}_{1,1}) = \Theta(m^2)$. We later show that $\Theta(m^2)$ distortion is asymptotically optimal for voting rules with deterministic elicitation, deterministic aggregation, and at most $\log m$ communication complexity (Theorem 4.4).

- **$t = 1, \ell = 2$**: In this case, $C(\text{PrefThreshold}_{1,2}) = \log m + 1$ and $\text{dist}(\text{PrefThreshold}_{1,2}) = \Theta(m\sqrt{m})$. This shows that eliciting just one extra bit per voter compared to plurality is sufficient for achieving subquadratic distortion.

- **$t = 1, \ell = \log m$**: In this case, we obtain $C(\text{PrefThreshold}_{1,\log m}) = \log m + \log \log m = \Theta(\log m)$ and $\text{dist}(\text{PrefThreshold}_{1,\log m}) = \Theta(m)$. Thus, asking each voter to report not only her most favorite alternative, but also her approximate value for this alternative allows achieving linear distortion with the same asymptotic communication complexity as that of plurality.

- **$t = m^{1-\gamma}, \ell = \log m$, where $\gamma \in (0,1)$ is a constant**: This achieves sublinear distortion with polynomial communication complexity. Specifically, $C(\text{PrefThreshold}_{m^{1-\gamma},\log m}) = \Theta(m^{1-\gamma}\log m)$ and $\text{dist}(\text{PrefThreshold}_{m^{1-\gamma},\log m}) = \Theta(m^{\gamma})$.

- **$t = m/\sqrt{\log m}, \ell = \log m$**: We obtain

$$C\left(\text{PrefThreshold}_{m/\sqrt{\log m},\log m}\right) = O\left(\frac{m \log \log m}{\sqrt{\log m}}\right) = o(m),$$

$$\text{dist}\left(\text{PrefThreshold}_{m/\sqrt{\log m},\log m}\right) = O\left(\sqrt{\log m}\right).$$

This choice Pareto-dominates the use of threshold approval votes, which has higher communication complexity of $m$ and results in higher distortion of $\Omega(\log m/\log \log m)$, even when randomized aggregation is allowed [Ben+17].

- **$t = m, \ell = \log m$**: In this case, each voter reports her approximate value for each alternative. We obtain $C(\text{PrefThreshold}_{m,\log m}) = \Theta(m \log \log m)$ and
dist(PrefThreshold\textsubscript{m,log\textsubscript{m}}) = O(1). By contrast, eliciting ranked preferences leads to not only higher communication complexity of Θ(m log m), but also significantly higher distortion of Θ(m²) with deterministic aggregation [Car+17] and Ω(\sqrt{m}) with randomized aggregation [Bou+15]. In other words, this choice Pareto-dominates the use of ranked preferences.

4.3.2 Randomized Elicitation, Randomized Aggregation

We now present a generic approach to designing voting rules with randomized elicitation. Given a voting rule \( f \) and an integer \( s \leq m \), instead of using \( f \) to select one alternative from \( A \) directly, we sample \( S \subseteq A \) with \( |S| = s \) at random and use \( f \) to select one alternative from \( S \).

Recall that for \( p \in \mathbb{N} \), \( C^{p}(f) \) and \( \text{dist}^{p}(f) \) denote the communication complexity and distortion of \( f \) for \( p \) alternatives, respectively.

Clearly, this approach reduces the communication complexity from \( C^{m}(f) \) to \( C^{s}(f) \). Its effect on distortion, however, is more subtle. On the one hand, selecting an alternative from \( S \) instead of \( A \) results in an inevitable loss of welfare because we can only hope to do as well as the best alternative in \( S \). On the other hand, the welfare we achieve is related to the welfare of the best alternative in \( S \) via the factor \( \text{dist}^{s}(f) \), which can be significantly lower than \( \text{dist}^{m}(f) \). We show that in some cases, this approach reduces distortion in addition to reducing communication complexity.

The key challenge in making this approach work is that we cannot apply \( f \) directly to select one alternative from \( S \). This is because \( f \) assumes that each voter has a total value of 1 for the set of alternatives under consideration. This is true when we apply \( f \) to select an alternative from \( A \), but not when we apply it to select an alternative from \( S \). We circumvent this obstacle by eliciting an approximate value of \( v_i(S) \) from each voter \( i \), making a number of copies of voter \( i \) that is approximately proportional to \( v_i(S) \) (where each copy now has a total value of 1 for alternatives in \( S \)), and running \( f \) on the resulting instance.

**Theorem 4.2.** For every voting rule \( f \) and \( s \in [m] \), we have \( C^{m}(\text{RandSubset}(f, s)) = C^{s}(f) + \log(\log(4m)) \) and \( \text{dist}^{m}(\text{RandSubset}(f, s)) \leq \frac{4m}{s} \cdot \text{dist}^{s}(f) \).
Proof. Let $\bar{v} = (v_1, \ldots, v_n)$ denote the underlying valuations of voters. First, let us consider a fixed choice of $S \subseteq A$ with $|S| = s$. Due to our bucketing, we have that for every $i \in N$,
\begin{equation}
\frac{v_i(S)}{2} - \frac{1}{4m} \leq L_{p_i} \leq v_i(S). \tag{4.4}
\end{equation}

Recall that in the input to the aggregation rule of $f$, we have $4m \cdot L_{p_i}$ copies of the response $\rho_i$ of voter $i$. Hence, the social welfare function approximated by the aggregation rule of $f$ is given by
\[\forall a \in S, \quad \widehat{sw}(a, \bar{v}) = \sum_{i \in N} 4m \cdot L_{p_i} \cdot \frac{v_i(a)}{v_i(S)} = 4m \sum_{i \in N} v_i(a) \cdot \frac{L_{p_i}}{v_i(S)}.\]
Combining this with Equation (4.4), we have that for each $a \in S$,
\begin{equation}
\widehat{sw}(a, \bar{v}) \geq 4m \sum_{i \in N} v_i(a) \cdot \left(1 - \frac{1}{4m \cdot v_i(S)}\right) = 2m \cdot sw(a, \bar{v}) - \sum_{i \in N} \frac{v_i(a)}{v_i(S)} \geq 2m \cdot sw(a, \bar{v}) - n, \tag{4.5}
\end{equation}
as well as
\begin{equation}
\widehat{sw}(a, \bar{v}) \leq 4m \sum_{i \in N} v_i(a) \cdot 1 = 4m \cdot sw(a, \bar{v}). \tag{4.6}
\end{equation}
Let \(\hat{a}\) denote the alternative chosen by our rule. Because the distortion of $f$ for choosing an
alternative from \( S \) is \( \text{dist}^t(f) \), we have that \( \mathbb{E} \left[ \left\| \hat{\text{sw}}(\hat{a}, \bar{v}) \right\| \right] \geq \max_{a \in S} \hat{\text{sw}}(a, \bar{v}) / \text{dist}^t(f) \). Note that so far, we have fixed \( S \). The expectation on the left hand side is due to the fact that even for fixed \( S \), \( \hat{a} \) can be randomized if \( f \) is randomized.

Next, we take expectation over the choice of \( S \), and use the fact that the optimal alternative \( a^* \in \arg \max_{a \in A} \text{sw}(a, \bar{v}) \) belongs to \( S \) with probability \( s/m \). We obtain

\[
\mathbb{E} \left[ \hat{\text{sw}}(\hat{a}, \bar{v}) \right] \geq \mathbb{E} \left[ \frac{\max_{a \in S} \hat{\text{sw}}(a, \bar{v})}{\text{dist}^t(f)} \right] \geq \frac{\frac{m}{n} \cdot \hat{\text{sw}}(a^*, \bar{v})}{\text{dist}^t(f)} \geq \frac{\frac{m}{n} (2m \cdot \text{sw}(a^*, \bar{v}) - n)}{\text{dist}^t(f)},
\]

where the final transition follows from Equation (4.5). On the other hand, from Equation (4.6), we have

\[
\mathbb{E} \left[ \hat{\text{sw}}(\hat{a}, \bar{v}) \right] \leq 4m \mathbb{E} \left[ \text{sw}(\hat{a}, \bar{v}) \right].
\]

Combining Equations (4.7) and (4.8), we have that

\[
\text{dist}^m(\text{RandSubset}(f, s)) = \frac{\text{sw}(a^*, \bar{v})}{\mathbb{E} \left[ \text{sw}(\hat{a}, \bar{v}) \right]} \leq \frac{\text{sw}(a^*, \bar{v})}{\text{sw}(a^*, \bar{v})} - \frac{n}{4m} \cdot \frac{m}{s} \cdot \text{dist}^t(f) \leq \frac{4m}{s} \cdot \text{dist}^t(f),
\]

where the final transition uses the fact that \( \text{sw}(a^*, \bar{v}) \geq (1/m) \cdot \sum_{a \in A} \text{sw}(a, \bar{v}) = n/m \). This establishes the desired distortion bound. Since each voter answers the query of \( f \) for \( s \) alternatives and chooses one of \( \lfloor \log(4m) \rfloor \) buckets, we get \( C^m(\text{RandSubset}(f, s)) = C^s(f) + \log(\log(4m)) \), as desired.

Using \( f = \text{PrefThreshold}_{t, \ell} \) and Theorem 4.1, we obtain that for \( s \in [m], t \in [s], \) and \( \ell \in \mathbb{N}, \)

\[
C^m(\text{RandSubset}(\text{PrefThreshold}_{t, \ell}, s)) = O \left( t \log(s(\ell + 1)/t) + \log m \right),
\]

\[
\text{dist}^m(\text{RandSubset}(\text{PrefThreshold}_{t, \ell}, s)) = O \left( m \cdot 2^{\ell/t} / t \right).
\]

Setting \( \ell = \log s \), we get \( O(m/t) \) distortion. Then, we set \( s = t \) to minimize communication complexity to \( O(t \log \log t + \log \log m) \). This is slightly better than using \( \text{PrefThreshold}_{t, \log m} \), which achieves \( O(m/t) \) distortion with \( O(t \log m \log m) \) communication complexity. In particular, for \( t = O(1) \) this reduces communication complexity by a factor of \( \log m / \log \log m \).

An interesting choice is \( t = \frac{\log m}{\log \log m} \), which leads to distortion \( O(m \log \log m / \log m) = o(m) \) and communication complexity \( O(t \log \log t + \log \log m) = o(\log m) \). Note that this rule has randomized elicitation but deterministic aggregation. By contrast, we later show that with

98
deterministic elicitation, no voting rule can achieve \( o(m) \) distortion with communication complexity at most \( \log m \), even when randomized aggregation is allowed (Theorem 4.4).

### 4.4 Direct Lower Bounds For Deterministic Elicitation

We now turn our attention to deriving lower bounds on the distortion of a voting rule given an upper bound on its communication complexity (equivalently, this gives a lower bound on the communication complexity required to achieve a given level of distortion). In this section, our focus is on deterministic elicitation. In the next section, we use tools from multi-party communication complexity to derive lower bounds for both deterministic and randomized elicitation.

Consider a voting rule \( f \) which uses deterministic elicitation and has communication complexity at most \( \log k \). Hence, the (deterministic) query of \( f \) must partition \( \Delta^m \) into at most \( k \) compartments. We argue that for deriving a lower bound on the distortion of \( f \), we can assume, without loss of generality, that it uses exactly \( k \) compartments. This is because if \( f \) uses \( k' \) compartments where \( k' < k \), then we can partition some of its compartments into smaller compartments and derive a new voting rule \( g \) which uses exactly \( k \) compartments, receives at least the information that \( f \) receives from the voters, and simulates the aggregation rule of \( f \) to achieve the same distortion. Thus, let us assume that \( f \) uses exactly \( k \) compartments.

Now, establishing a lower bound on the distortion of \( f \) requires analyzing the following game between two players, the voting rule \( f \) and the adversary.

1. The voting rule \( f \) decides the partition of \( \Delta^m \) into \( k \) compartments.
2. The adversary decides the response of each voter.
3. The voting rule \( f \) picks a winning alternative (or a distribution over winning alternatives, if its aggregation rule is randomized).
4. The adversary picks valuations of voters consistent with their responses in the second step.
We use this framework to derive lower bounds on the distortion of voting rules that use deterministic elicitation. We first focus on deterministic aggregation. Perhaps the simplest such voting rule is plurality, which has \( \log m \) communication complexity and achieves \( \Theta(m^2) \) distortion. This raises an important question: What distortion can we achieve with deterministic elicitation, deterministic aggregation, and communication complexity less than \( \log m \)?

To answer this question, we start by establishing a straightforward lemma. Recall that for a valuation \( v \in \Delta^m \), \( \text{supp}(v) \) denotes the support of \( v \).

**Lemma 4.1.** Let \( f \) be a voting rule which uses deterministic elicitation and deterministic aggregation. Let \( q^* \) be the query used by \( f \). If some compartment of \( q^* \) contains two valuations \( v^1 \) and \( v^2 \) such that \( \text{supp}(v^1) \cap \text{supp}(v^2) = \emptyset \), then the distortion of \( f \) is unbounded.

**Proof.** Suppose compartment \( P \) contains valuations \( v^1 \) and \( v^2 \) such that \( \text{supp}(v^1) \cap \text{supp}(v^2) = \emptyset \). Let \( \hat{a} \) be the alternative returned by \( f \) when all voters pick compartment \( P \). Pick \( t \in \{1, 2\} \) such that \( \hat{a} \notin \text{supp}(v^t) \). Note that \( v^t(\hat{a}) = 0 \), but there exists \( a^* \in \text{supp}(v^t) \) such that \( v^t(a^*) > 0 \).

Define voter valuations \( \vec{v} = (v_1, \ldots, v_n) \) such that \( v_i = v^t \) for each \( i \in N \). This yields \( \text{sw}(\hat{a}, \vec{v}) = 0 \) and \( \text{sw}(a^*, \vec{v}) > 0 \), which implies that \( f \) must have infinite distortion. \( \square \)

Next, we leverage this lemma to show that communication complexity less than \( \log m \) leads to unbounded distortion. For this, we need the following definition. For \( a \in A \), we say that the *unit valuation* corresponding to \( a \) is the valuation \( \nu^a \in \Delta^m \) for which \( \nu^a(a) = 1 \).

**Theorem 4.3.** Every voting rule that has deterministic elicitation, deterministic aggregation, and communication complexity strictly less than \( \log m \) has unbounded distortion.

**Proof.** Let \( f \) be a voting rule that has deterministic elicitation and deterministic aggregation, and let \( C(f) < \log m \). Hence, the query used by \( f \) must partition \( \Delta^m \) into less than \( m \) compartments.

Because there are \( m \) unit valuations, by the pigeonhole principle there must exist distinct \( a, b \in A \) such that \( \nu^a \) and \( \nu^b \) belong to the same compartment. Because \( \text{supp}(\nu^a) \cap \text{supp}(\nu^b) = \emptyset \), Lemma 4.1 implies that the distortion of \( f \) must be infinite. \( \square \)

Thus, we have established that with deterministic aggregation, we must have communication complexity at least \( \log m \) to achieve finite distortion. Plurality has communication
complexity $\log m$ and achieves $\Theta(m^2)$ distortion. Can a different voting rule achieve better distortion using only $\log m$ communication complexity? Perhaps unsurprisingly, we answer this in the negative. However, the proof of this intuitive result is surprisingly intricate.

Further, using randomized aggregation we can trivially achieve $O(m)$ distortion with zero communication complexity (by returning the uniform distribution over alternatives). One may wonder: How much information do we need from the voters to achieve sublinear distortion? It is easy to show that eliciting plurality votes is not sufficient (while this is implied by the next result, we present a much simpler proof in Appendix C.1). Surprisingly, we show that this holds for every $\log m$-bit elicitation. That is, even with randomized aggregation, eliciting $\log m$ bits per voter is asymptotically no better than blindly selecting an alternative uniformly at random!

**Theorem 4.4.** Let $f$ be a voting rule which uses deterministic elicitation and has $C(f) \leq \log m$. If $f$ uses deterministic aggregation, then $\text{dist}(f) = \Omega(m^2)$. If $f$ uses randomized aggregation, then $\text{dist}(f) = \Omega(m)$.

**Proof.** Let $f$ be a voting rule which has deterministic elicitation and $C(f) \leq \log m$. As argued above, we can assume $C(f) = \log m$ without loss of generality. Hence, the query $q^*$ used by $f$ partitions $\Delta^m$ into $m$ compartments. Let $\mathcal{P} = (P_1, \ldots, P_m)$ denote the set of compartments. If $f$ has unbounded distortion, we are done. Suppose $f$ has bounded distortion.

Due to Lemma 4.1, each of $m$ unit vectors must belong to a different compartment. Since there are $m$ compartments, we identify each compartment by the unit valuation it contains. For $a \in A$, let $P^a$ denote the compartment containing unit valuation $v^a$. Before we construct adversarial valuations, we need to define low valuations and high valuations.

**Low valuations:** We say that a valuation $v \in \Delta^m$ is a low valuation if $|\text{supp}(v)| = m/5$ and $v(a) = 5/m$ for every $a \in \text{supp}(v)$. Let $\Delta^{m,\text{low}}$ denote the set of all low valuations. Due to Lemma 4.1, we have

$$v \in \Delta^{m,\text{low}}, a \in \text{supp}(v) \Rightarrow v(a) = \frac{5}{m}. \quad (4.9)$$

Let $\mathcal{L} = \{P \in \mathcal{P} : P \cap \Delta^{m,\text{low}} \neq \emptyset\}$ be the set of compartments containing at least one low valuation, and $A^\mathcal{L} = \{a \in A : P^a \in \mathcal{L}\}$ be the set of alternatives corresponding to these compartments.
We claim that $|A^L| = |L| \geq 4m/5 + 1$. Suppose for contradiction that $|A^L| \leq 4m/5$. Then, $|A \setminus A^L| \geq m/5$. Hence, there exists a low valuation $v \in \Delta_{m,\text{low}}$ such that $\text{supp}(v) \subseteq A \setminus A^L$. Let $a \in A$ be the alternative for which $v \in P^a$. Because $P^a$ contains a low valuation, $a \in A^L$ by definition. Thus, the construction of $v$ ensures $v(a) = 0$. We have $v \in \Delta_{m,\text{low}} \cap P^a$ with $v(a) = 0$, which contradicts Equation (4.9). Hence, $|A^L| \geq 4m/5 + 1$.

**High valuations:** We say that a valuation $v \in \Delta^m$ is a high valuation if $|\text{supp}(v)| = 2$ and $v(a) = 1/2$ for each $a \in \text{supp}(v)$. Let $\Delta_{m,\text{high}}$ denote the set of high valuations. Note that $|\Delta_{m,\text{high}}| = \binom{m}{2}$.

Similarly to the case of low valuations, we can apply Lemma 4.1, and obtain that

$$v \in \Delta_{m,\text{high}} \cap P^a \Rightarrow a \in \text{supp}(v) \land v(a) = \frac{1}{2}. \quad (4.10)$$

For $a \in A$, let $H^a = \{P \in L : \exists v \in \Delta_{m,\text{high}} \cap P \text{ s.t. } a \in \text{supp}(v)\}$. In words, $H^a$ is the set of compartments from $L$ which contain at least one high valuation $v$ for which $v(a) = 1/2$. Let $A^\text{high} = \{a \in A : |H^a| \geq m/5\}$. We claim that $|A^\text{high}| \geq m/6$.

Suppose this is not true. Let $B = |A \setminus A^\text{high}|$. Then, $|B| \geq 5m/6$. Consider $a \in B$. Each of the $m - 1$ high valuations which contain $a$ in their support must belong to some compartments in $H^a \cup (P \setminus L)$. Since $|H^a| \leq m/5 - 1$ for $a \in B$ and $|P \setminus L| \leq m/5 - 1$, the $m - 1$ high valuations containing $a$ in their support are distributed across at most $2m/5 - 2$ compartments. However, due to Lemma 4.1, a compartment other than $P^a$ can contain at most one high valuation with $a$ in its support. Hence, $P^a$ must contain at least $m - 1 - (2m/5 - 3) = 3m/5 + 2$ high valuations. Thus, we have established that $|B| \geq 5m/6$ and for each $a \in B$, $P^a$ contains at least $3m/5 + 2$ high valuations. Thus, the number of high valuations is at least $(5m/6) \cdot (3m/5 + 2) > m^2/2 > \binom{m}{2}$, which is a contradiction. Thus, we have $|A^\text{high}| \geq m/6$.

We are now ready to prove the desired result for both deterministic and randomized aggregation.

**Voter responses:** When responding to the query $q^a$, suppose each compartment $P \in L$ is picked by a set $N_P$ of $n/|L|$ voters.

**Deterministic aggregation:** Let $\hat{a}$ denote the alternative picked by $f$. We claim that $\hat{a} \in A^L$. If $\hat{a} \notin A^L$, consider voter valuations $\bar{v}$ such that every voter $i$ picking compartment $P^a \in L$ has...
valuation $v_i = \nu^a$. Since $\hat{a} \notin A^\mathcal{L}$, we have $v_i(\hat{a}) = 0$ for each $i \in N$, i.e., $sw(\hat{a}, \nu) = 0$. Since $sw(a, \nu) > 0$ for some $a \in A$, $f$ has infinite distortion, which is a contradiction. Thus, we must have $\hat{a} \in A^\mathcal{L}$.

Now, let us construct the voter valuations as follows. Pick a low valuation $\hat{\nu} \in P\hat{a} \cap \Delta^{m,\text{low}}$, which exists because we have established $\hat{a} \in A^\mathcal{L}$. Note that $\hat{\nu}(\hat{a}) = 5/m$. For each $i \in N_{p\hat{a}}$, let $v_i = \hat{\nu}$. Pick $a^* \in A^{\text{high}} \setminus \{\hat{a}\}$. Let $P$ be the compartment containing the high valuation under which both $\hat{a}$ and $a^*$ have utility 1/2. For each $P \in \mathcal{H}^{a^*} \setminus \{P\hat{a}, \bar{P}\}$, and for each $i \in N_P$, let $v_i$ be the high valuation in $P$ such that $v_i(a^*) = 1/2$ and $v_i(\hat{a}) = 0$. For every other $P^a \in \mathcal{L}$ and every $i \in N_{P^a}$, let $v_i = \nu^a$.

Observe that under these valuations, $sw(\hat{a}, \nu) = \Theta(n/m^2)$, whereas, since $|\mathcal{H}^{a*}| \geq m/5$ and $|\mathcal{L}| \leq |P| = m$, $sw(a^*, \nu) = \Theta(n)$. We conclude that $\text{dist}(f) = \Omega(m^2)$.

**Randomized aggregation:** Note that $f$ must select at least one alternative $a^* \in A^{\text{high}}$ with probability at most $1/|A^{\text{high}}| \leq 6/m$. Construct voter valuations such that for every $P \in \mathcal{H}^{a*}$ and every $i \in N_P$, $v_i$ is the high valuation under which $v_i(a^*) = 1/2$. For every $P^a \in \mathcal{L} \setminus \mathcal{H}^{a*}$, and for every $i \in N_{P^a}$, let $v_i = \nu^a$. It holds that $sw(a^*, \nu) = \Theta(n)$ (as before), whereas $sw(a, \nu) = O(n/m)$ for every $a \in A \setminus \{a^*\}$. Because $f$ selects $a^*$ with probability at most $6/m$, we have $E_{\tilde{a} \sim f(\nu)} [sw(\hat{a}, \nu)] = O(n/m)$, implying $\text{dist}(f) = \Omega(m)$, as required. $\square$

For deterministic aggregation, Theorem 4.4 shows that eliciting $\log m$ bits per voter is not sufficient to achieve $o(m^2)$ distortion. By contrast, we know from Theorem 4.1 that we can achieve $O(m)$ distortion by eliciting $O(\log m)$ bits per voter. Similarly, for randomized aggregation, Theorem 4.4 shows that eliciting $\log m$ bits per voter is not sufficient to achieve $o(m)$ distortion. However, we can achieve $o(m)$ distortion if we are willing to elicit $\omega(\log m)$ bits per voter (Theorem 4.1), or if we are willing to use randomized elicitation (Theorem 4.2).

---

6For $t = \omega(1)$, PrefThreshold$_{\log m}$ achieves $O(m/t) = o(m)$ distortion and has communication complexity $O(t \log m)$. 
4.5 Lower Bounds Through Multi-Party Communication Complexity

In this section, we leverage tools from the literature on multi-party communication complexity to derive lower bounds for both deterministic and randomized elicitation. Specifically, we derive lower bounds on the communication complexity of voting rules that achieve a given level of distortion. We can equivalently interpret these results similarly to the results in the previous section, i.e., as lower bounds on the distortion given an upper bound on the communication complexity. We use the former interpretation as it allows to make a direct connection to the literature on communication complexity, which aims to derive lower bounds on the communication required to solve a problem.

We begin by reviewing existing results on multi-party communication complexity, and then derive new results, which help us prove the desired lower bounds in our voting context.

4.5.1 Setup

In multi-party communication complexity, there are $t$ computationally omnipotent players. Each player $i$ holds a private input $X_i \in \mathcal{X}_i$. The input profile is the vector $(X_1, \ldots, X_t)$. The goal is to compute the output of a function $f : \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_t \rightarrow \{0, 1\}$ on the input profile.

A shared protocol $\Pi$ specifies how the players exchange information among themselves and with the center. We use the blackboard model, in which messages written by one player are visible to all other players. For a fixed input profile $(X_1, \ldots, X_t)$, let $\Pi(X_1, \ldots, X_t)$ denote the random variable representing the message transcript obtained when players follow the protocol. Here, the randomness is due to coin tosses either by the players or in the protocol.

The communication cost of $\Pi$, denoted $|\Pi|$, is the maximum length of $\Pi(X_1, \ldots, X_t)$ over all input profiles $(X_1, \ldots, X_t)$ and all coin tosses. Given $\delta \geq 0$, we say that $\Pi$ is a $\delta$-error protocol for $f$ if there exists a function $\Pi_{\text{out}}$ such that for every input profile $(X_1, \ldots, X_t)$, we have

$$\Pr[\Pi_{\text{out}}(\Pi(X_1, \ldots, X_t)) = f(X_1, \ldots, X_t)] \geq 1 - \delta.$$ 

The $\delta$-error communication complexity of $f$, denoted $R_\delta(f)$, is the communication cost of the best
4.5.2 Multi-Party Fixed-Size Set-Disjointness

The main ingredient of our proof is the multi-party set-disjointness problem, denoted \( \text{DISJ}_{m,t} \). This is a standard problem in multi-party communication complexity. In this problem, there are \( t \) players. Each player \( i \) holds an arbitrary set \( S_i \) from a universe of size \( m \). The goal is to distinguish between two types of inputs.

- **NO inputs:** The sets are pairwise disjoint, i.e., \( S_i \cap S_j = \emptyset \) for all \( i \neq j \).
- **YES inputs:** The sets have a unique element in common, but are otherwise pairwise disjoint, i.e., there exists \( x \) such that \( S_i \cap S_j = \{x\} \) for all \( i \neq j \).

It is promised that the input will be one of these two types (in other words, the protocol is free to choose any output on an input that does not satisfy this promise). Alon, Matias, and Szegedy [AMS99] proved that \( R_\delta(\text{DISJ}_{m,t}) = \Omega(m/t^4) \). This was later improved to \( \Omega(m/t^2) \) by Bar-Yossef et al. [BY+04] and then to \( \Omega(m/(t \log t)) \) by Chakrabarti, Khot, and Sun [CKS03]. Finally, Gronemeier [Gro09] and Jayram [Jay09] established the optimal lower bound of \( \Omega(m/t) \).

We introduce a variant of this problem, which we call multi-party fixed-size set-disjointness and denote \( \text{FDISJ}_{m,s,t} \). It is almost identical to \( \text{DISJ}_{m,t} \), except that we know each player \( i \) holds a set \( S_i \) of a given size \( s \). Our goal is to still determine whether the sets are pairwise disjoint (\( S_i \cap S_j = \emptyset \) for all \( i \neq j \)) or pairwise uniquely intersecting (there exists \( x \) such that \( S_i \cap S_j = \{x\} \) for all \( i \neq j \)). We use the lower bound on \( R_\delta(\text{DISJ}_{m,t}) \) to derive the following lower bound on \( R_\delta(\text{FDISJ}_{m,s,t}) \). We do so by reducing the standard set-disjointness problem to its fixed-size variant. A detailed proof is provided in Appendix C.2.

**Theorem 4.5.** For a sufficiently small constant \( \delta > 0 \) and \( m \geq (3/2)st \), \( R_\delta(\text{FDISJ}_{m,s,t}) = \Omega(s) \).

4.5.3 Lower Bounds on the Communication Complexity of Voting Rules

We now use our lower bound on the \( \delta \)-error communication complexity of \( \text{FDISJ}_{m,s,t} \) to derive a lower bound on the communication complexity of a voting rule in terms of its distortion.
We derive different bounds depending on whether the elicitation rule of $f$ is deterministic or randomized. For randomized elicitation, our bound is weaker.

The key insight in the proof is that we can use a voting rule $f$ with $\text{dist}(f) = \frac{1}{2}$ to construct a $\delta$-error protocol for solving $\text{FDISJ}_{m,s,t}$, and hence we can use the lower bound on $R_\delta(\text{FDISJ}_{m,s,t})$ from Theorem 4.5 to derive a lower bound on $C(f)$. At a high level, consider an instance $(S_1, \ldots, S_t)$ of $\text{FDISJ}_{m,s,t}$. We ask each player $i$ to respond to the query of $f$ according to an artificial valuation function constructed using $S_i$. We then use these responses to create an input for the aggregation rule of $f$. We show that by asking each player an additional question about the alternative returned by the aggregation rule, and possibly running this process a number of times, we can solve $\text{FDISJ}_{m,s,t}$.

**Theorem 4.6.** For a voting rule $f$ with elicitation rule $\Pi_f$ and $\text{dist}(f) = d$, the following hold.

- If $\Pi_f$ is deterministic, then $C(f) \geq \Omega\left(\frac{m}{d^2}\right)$.
- If $\Pi_f$ is randomized, then $C(f) \geq \Omega\left(\frac{m}{d^3}\right)$.

**Proof.** Let $t = 2 \cdot \text{dist}(f)$ and $s = 2m/(3t)$. Note that for these parameters, we have $R_\delta(\text{FDISJ}_{m,s,t}) = \Omega(s)$ from Theorem 4.5.

Consider an input $(S_1, \ldots, S_t)$ to $\text{FDISJ}_{m,s,t}$ with a universe $U$ of size $m$. Let us create an instance of the voting problem with a set of $n$ voters $N$ and a set of $m$ alternatives $A$. Each alternative in $A$ corresponds to a unique element of $U$. Partition the set of voters $N$ into $t$ equal-size buckets $\{N_1, \ldots, N_t\}$. Here, bucket $N_i$ corresponds to player $i$, and consists of $n/t$ voters that each have valuation $v^{S_i}$ given by $v^{S_i}(a) = 1/s$ for each $a \in S_i$ and $v^{S_i}(a) = 0$ for each $a \notin S_i$. Let $\bar{v}$ denote the resulting profile of voter valuations. Note that under these valuations, $sw(a, \bar{v}) = \frac{n}{ls} \sum_{i=1}^t 1[a \in S_i]$, where 1 is the indicator variable. Due to the promise that an element either belongs to at most one set or belongs to every set, we have $sw(a, \bar{v}) \in \{0, n/(ts), n/s\}$. We say that $a$ is a “good” alternative if $sw(a, \bar{v}) = n/s$ and a “bad” alternative otherwise.

We define two processes that will help covert our voting rule $f$ into a protocol for $\text{FDISJ}_{m,s,t}$.

**Process E:** In this process, we ask each player $i$ to respond to the query posed by voting rule
f (possibly selected in a randomized manner) according to valuation $v^i$. We note that this requires a total of $t \cdot C(f)$ bits of communication from the players.

Process A: We take players’ responses from process E, create $n/t$ copies of the response of each player, and pass the resulting profile as input to the aggregation rule $\Gamma_f$ to obtain the returned alternative $\hat{a}$ (possibly selected in a randomized manner). We end the process by determining if $\hat{a}$ is a good alternative or a bad alternative. This requires eliciting 2 extra bits of information: we can ask any two players $i$ and $j$ whether their sets contain $\hat{a}$, and due to the promise of FDISJ_{m,s,t}, we know that $\hat{a}$ is good if and only if it belongs to both $S_i$ and $S_j$.

Knowing whether $\hat{a}$ is good or bad is useful for solving the given instance of FDISJ_{m,s,t} due to the following reason.

1. If $(S_1, \ldots, S_t)$ is a “NO input”, then we know that every alternative is a bad alternative. Hence, $sw(a, \bar{v}) \leq (n/t) \cdot (1/s) = n/(ts)$ for each $a \in A$. In particular, this implies $sw(\hat{a}, \bar{v}) \leq n/(ts)$ with probability 1.

2. If $(S_1, \ldots, S_t)$ is a “YES input”, then there exists a unique good alternative $a^* \in A$ with $sw(a^*, \bar{v}) = n/s$, and every other alternative $a$ is a bad alternative with $sw(a, \bar{v}) \leq n/(ts)$. Because $dist(f) = t/2$, we have that $E[sw(\hat{a}, \bar{v})] \geq n/s = 2n/2t$. This implies that $Pr[sw(\hat{a}, \bar{v}) = n/s] = Pr[\hat{a} = a^*] \geq 1/t$ because if $Pr[\hat{a} = a^*] < 1/t$, then $E[sw(\hat{a}, \bar{v})] < (1/t) \cdot (n/s) + 1 \cdot n/(ts) = 2n/(ts)$, which is a contradiction.

We are now ready to use $f$ to construct a protocol for FDISJ_{m,s,t}, and use Theorem 4.5 to derive a lower bound on $C(f)$. We consider two cases depending on whether the elicitation rule $\Pi_f$ is deterministic or randomized.

1. Deterministic elicitation: In this case, we run process E once and then run process A $t \ln(1/\delta)$ times. In a NO input, we always get a bad alternative. In a YES input, each run of process A returns a good alternative with probability at least $1/t$. Hence, the probability that we get a good alternative at least once is at least $1 - (1 - 1/t)^{t \ln(1/\delta)} \geq 1 - \delta$. Hence, this is a $\delta$-error protocol for FDISJ_{m,s,t} which requires $t \cdot C(f) + t \ln(1/\delta) \cdot 2$ bits of total communication from the players. Using Theorem 4.5, we have that $t \cdot (C(f) + 2 \ln(1/\delta)) = \Omega(s)$. Using $s = 2m/(3t)$ and $t = 2d$, we have $C(f) = \Omega(m/d^2)$. 

107
2. **Randomized elicitation:** In this case, we run E once followed by running A once. And we repeat this entire process \( t \ln(1/\delta) \) times. Note that we need to repeat process E because the elicitation is also randomized. Like in the previous case, we always get a bad alternative in a NO input, and get a good alternative with probability at least \( 1/t \) in each run in a YES input. Hence, in a YES input, we get a good alternative in at least one run with probability at least \( 1 - (1 - 1/t)^{t \ln(1/\delta)} \geq 1 - \delta \). This results in a \( \delta \)-error protocol for FDISJ\(_{m,s,t} \) which requires \( t \ln(1/\delta) \cdot (t \cdot C(f) + 2) \) bits of total communication from the players. Using Theorem 4.5, we have \( t \ln(1/\delta) \cdot (t \cdot C(f) + 2) = \Omega(s) \). Using \( s = 2m/(3t) \) and \( t = 2d \), we have \( C(f) = \Omega(m/d^3) \).

These are the desired lower bounds on \( C(f) \).

Note that we do not obtain an improved asymptotic lower bound on \( C(f) \) when \( f \) uses deterministic aggregation as compared to when it uses randomized aggregation. In case of deterministic elicitation, we could slightly improve the bound by running process A only once instead of running it \( \Theta(t) \) times as it is guaranteed to always return a good alternative. However, because process A requires much less communication than process E, this does not provide an asymptotic improvement. In the case of randomized elicitation, we need to run both processes E and A repeatedly anyway, so having deterministic aggregation does not seem to help.

Finally, let us consider the lower bounds on \( C(f) \) implied by Theorem 4.6 for interesting choices of upper bounds on \( \text{dist}(f) \), and compare them with our previous results.

- When \( f \) uses deterministic elicitation and \( \text{dist}(f) = O(\sqrt{m / \log m}) \), we have that \( C(f) = \Omega(\log m) \). Theorem 4.4 already provides a stronger result: with deterministic elicitation, even \( \text{dist}(f) = o(m) \) implies \( C(f) = \Omega(\log m) \). However, Theorems 4.3 and 4.4 fail to impose a super-logarithmic lower bound on \( C(f) \), which, as we see below, Theorem 4.6 is powerful enough to do.

- When \( f \) uses deterministic elicitation and \( \text{dist}(f) = O(m^\gamma) \) for \( \gamma \in (0, 1/2) \), we have that \( C(f) = \Omega(m^{1-2\gamma}) \). For randomized elicitation and \( \gamma \in (0, 1/3) \), we have that \( C(f) = \Omega(m^{1-3\gamma}) \). By contrast, PrefThreshold\(_{m^{1-\gamma}\log m} \), which uses deterministic elicitation and
deterministic aggregation, achieves $O(m^\gamma)$ distortion with $O(m^{1-\gamma} \log m)$ communication complexity. In particular, this shows that in order to achieve $O(m^\gamma)$ distortion for constant $\gamma < 1/3$, polynomial communication complexity is both necessary (even with randomized elicitation and aggregation) and sufficient (even with deterministic elicitation and aggregation).

- When $\text{dist}(f) = O(\log m)$, we have $C(f) = \Omega\left(m/\log^2 m\right)$ for deterministic elicitation and $C(f) = \Omega\left(m/\log^3 m\right)$ for randomized elicitation. By contrast, PrefThreshold$_{m/\log m, \log m}$, which uses deterministic elicitation and aggregation, achieves $O(\log m)$ distortion with $O(m \log \log m / \log m)$ communication complexity. Note that the upper and lower bounds on communication complexity differ by only polylogarithmic factors.

- Finally, when $\text{dist}(f) = O(1)$, we have $C(f) = \Omega(m)$ even with randomized elicitation and aggregation. By contrast, again, PrefThreshold$_{m, \log m}$ uses deterministic elicitation and aggregation to achieve $O(1)$ distortion with only $O(m \log \log m)$ communication complexity. In this case, our upper and lower bounds differ by only a sublogarithmic factor.

### 4.6 Discussion

We have gained a significant understanding of the communication-distortion tradeoff. But our work leaves open several research directions.

The most immediate direction is to improve our upper and lower bounds, and close the gap between them. Regarding our upper bounds, both families of voting rules that we introduce — PrefThreshold and RandSubset — use deterministic aggregation. Can randomized aggregation help? Also, using randomized elicitation in RandSubset, we can achieve sublinear distortion with communication complexity at most $\log m$; Theorem 4.4 shows that this is a barrier that deterministic elicitation cannot help break. This raises an elegant question: What is the best possible distortion with randomized elicitation and communication complexity at most $\log m$?
Regarding our lower bounds, in Section 4.5 we provide a general lower bound on communication complexity in terms of distortion. It is an interesting open question to derive better lower bounds. It would be especially interesting if a different problem from the literature on multi-party communication complexity could help in this regard.

Taking a broader viewpoint, it is possible to consider wilder forms of elicitation. For example, we could ask questions to which voters respond in a randomized fashion. Caragiannis and Procaccia [CP11] show that if each voter is to vote for a single alternative, but instead of picking her favorite alternative voter $i$ picks each alternative $a$ with probability $v_i(a)$, then one can achieve (roughly) $O(1)$ distortion. This only requires $\log m$ communication complexity. In this case, it makes sense to dive into the world of sublogarithmic communication complexity, which is uninteresting in our setting.

Another possibility is to study non-uniform elicitation rules, which can ask different voters different questions. In this case, Bhaskar, Dani, and Ghosh [BDG18] show that to achieve $O(1)$ distortion it is sufficient to ask $O(m^3 \log m)$ voters a single bit of information each: for a given random alternative and a given random threshold in $[0, 1]$, is your value for the alternative at least the threshold? In this case, the average number of bits required per voter vanishes even when we want to achieve constant distortion. Hence, it may make sense to instead focus on the total number of bits elicited.

Finally, one can also consider asking adaptive questions to voters based on past responses. While our model can already handle adaptive elicitation based on a voter’s own answers, it is strictly weaker than the model where one voter’s questions can be chosen based on another voter’s responses. Interestingly, our lower bounds from Section 4.5, which are derived using multi-party communication complexity techniques, apply even to adaptive elicitation rules so long as they are anonymous, that is, to elicitation rules which never ask different questions to two voters with the exact same valuation function. However, it is interesting to study what can be achieved with non-anonymous adaptive elicitation rules. Such rules have received significant attention in the computational social choice literature due to the fact that they can simulate efficient optimization methods such as stochastic gradient descent (see, e.g., [Gar+17]).

On a conceptual level, perhaps the main take-away message of our paper is that it pays off
to elicit and aggregate preferences “by any means necessary,” that is, potentially through highly nonstandard aggregation and, especially, elicitation rules. In the setting of Caragiannis and Procaccia [CP11] where voters are software agents, this is only natural. But when voters are people, it is crucial to better understand the implications of such unconventional approaches, both in terms of how communication complexity corresponds to cognitive burden, and in terms of the interpretability and transparency of aggregation rules.
Chapter 5

Causal Inference with Time-Varying Treatments

5.1 Introduction

One of the basic challenges in causal inference is to estimate a causal quantity from observational data. Often such datasets involve individuals who are subject to treatments over multiple time periods. The main goal is to estimate the effect of a treatment policy on the outcome. Consider a ride-sharing company, for example, which records several variables such as the number of trips, and trip origins and destinations, for each rider, and based on this information decides whether or not to provide monthly discounts. After running this experiment for several months, the company is interested to know whether providing discounts increases the number of trips taken. If the answer is yes, the company might also want to find a policy that would further increase the number of trips taken.

A second example comes from Acemoglu et al. [Ace+14], who consider a fundamental problem in political science: does democracy cause economic development, in relation to autocracy? The authors collect data from 184 countries over more than half a century. The outcome is GDP per capita and is measured every year. The variables recorded include whether the country was under democracy or autocracy, the population across different age brackets, and net financial inflow. The goal is to find out whether democracy increases GDP of the
countries over the periods when the country was under democracy.

There are two fundamental differences between these examples: (1) In the first example, not all users who are provided a discount end up using it, whereas in the second example the political status of a country is determined by policy. (2) The ridesharing company is perfectly aware of all the factors that go into the treatment policy, since it uses an algorithm to determine whether or not to assign a pass based on the past history of a rider. On the other hand, factors that affect the decision about the political status of a country may not be present in any data (and those decisions are far from algorithmic). Despite these differences, the same kind of question is of interest: what is the effect of a treatment policy over the subjects who are assigned the treatment? This quantity is known as the average treatment effect over the treated (ATET). There are also additional challenges. The subjects in both cases, the riders or countries, are heterogeneous, i.e., the effect of the same treatment policy can be expected to differ across subjects. Second, the number of time periods over which the policy is applied can be quite large, with treatments varying across time-steps.

*Marginal Structural Models* (MSMs) [Rob00] are widely used to estimate the causal quantity of interest when subjects receive treatments over multiple periods of time. However, MSMs have two main limitations: (a) they do not capture subject heterogeneity, and (b) they only consider fixed time intervals and do not scale gracefully with longer intervals. This latter limitation comes about because the number of parameters scales linearly with the length of the time interval, and with a fixed number of agents there is not enough data to estimate the parameters of the model.

In this work, we propose a new form of MSM to address these drawbacks. We assume that potential outcomes are generated from a three-dimensional tensor of low rank, where the dimensions correspond to the agents, time intervals, and set of possible histories. Intuitively, the rank of the tensor can be interpreted as a measure of the heterogeneity of the agents or the time periods. For example, if the rank is $r$, then each agent can be described as a linear combination of $r$ underlying groups. We assume the rank of the tensor is low, but we allow the dimensions of the tensor to increase with the number of agents and time periods.
5.1.1 Contributions

In order to estimate the outcome model, we set up a weighted tensor completion problem, and show that the solution converges to the true model. Compared to Robins [Rob00], we prove convergence for two cases – when the number of agents $N$ is fixed and the length of the time interval $T$ increases and when $T$ is fixed and $N$ increases. In particular, if the outcome at every time period depends only on the history of length $k$, then as long as $k$ is bounded by the logarithm of the increasing variable (be it $N$ or $T$), our method guarantees convergence. We solve the weighted tensor completion in two steps. First, we convert it to a weighted tensor approximation problem with an additive loss, where the loss goes to zero as either $N$ or $T$ increases. Then we turn to solving this weighted low-rank approximation problem, and provide conditions under which we can approximately solve the estimation problem in polynomial time. To the best of our knowledge, ours is the first additive approximation algorithm for the noisy weighted tensor completion that runs in polynomial time under reasonable conditions. Finally, we propose an algorithm based on projected gradient descent, which is easy to implement, and show that on a simulated dataset, it performs better than MSM and matrix completion in estimating ATET.

5.1.2 Related Work

The fundamental problem of causal inference is that for each unit we observe only one of two possible outcomes – the outcome corresponding to the treatment but not the control. A standard approach is to use the Neyman-Rubin potential outcomes framework [SNDS90; Rub74]. For each unit and each intervention (0 or 1), there are two potential outcomes $Y_0$ and $Y_1$, and we only observe one of these two outcomes. The traditional focus has been on estimating the average treatment effect (ATE), which measures the difference in average outcomes under treatment than without treatment. When the treatment policy is completely randomized, this quantity can be estimated by taking the average of the outcomes between the treatment and the control group. For observational data, this quantity can estimated through propensity score matching [RR83], which cleverly accounts for the covariates that predict treatment.
Traditionally, datasets have been too small to discover any heterogeneity in treatment effects. However, with ever-increasing data and improvements in machine learning algorithms, several recent papers have devised algorithms to discover heterogeneous treatment effects. They often involve machine learning techniques such as Bayesian nonparametrics [Hil11], random forests [WA18; AI16], and deep learning [SJS16; JSS16; YJS18]. Although we will be working with the potential outcomes framework, there has also been significant effort in using graphical models as a framework for causality [PM18], including attention to heterogeneous effects [SP12; Pea17]. However, we are not aware of work on combining these methods with the kinds of temporal settings studied here.

Epidemiologists and biostatisticians have also considered the problem of estimating the causal effect of a policy that applies treatments over multiple time periods. Robins [Rob86] proposed the marginal structural model (MSM), as a way to measure the causal effect of a time-varying treatment in the presence of time-varying confounders. Suppose, for example, that a policy applies a binary treatment over \( T \) time periods. MSM models each of the \( 2^T \) potential outcomes through a parametric model with parameter \( \beta \). Robins [Rob86] further showed that the solution to a maximum weighted likelihood correctly estimates the quantity \( \beta \). MSM has been adopted in various domains to estimate the causal effect in a longitudinal study, ranging from effect of different drugs on the mortality of HIV patients [RHB00] to the effect of loneliness on depression [Van+11].

There have been very few attempts to generalize these models to capture important aspects such as heterogeneous effects, large numbers of time-periods, or to the case when the outcome depends on a short history instead of the full history of length \( T \). This is because MSM was developed in the context of clinical trials, and most of these datasets are relatively small. Neugebauer et al. [Neu+07] define a history-adjusted MSM, which considers potential outcomes dependent on a short history instead of the full history of length \( T \). In particular, they propose a parametric model of the potential outcome conditioned on a history of treatments and covariates. Similar to Robins [Rob86], they propose an estimator based on maximum weighted likelihood, but that fails to capture heterogeneous effects over the population.

The most closely related prior work is that of Athey et al. [Ath+18], who use matrix
completion methods to estimate average treatment effects and other related causal quantities for the time-varying treatment setting. They model the potential outcomes using a matrix of low rank and provide an estimator. The rank of the underlying matrix captures different types of heterogeneous effects in the population. However, they do not consider the effect of past treatments on the outcomes. Rather, the potential outcome at each time step depends only on the current treatment. Boruvka et al. \cite{Bor+18} do consider time-varying treatments, but model treatment effect conditioned on a given history and under the same underlying policy; i.e., what would happen if treatment were switched from 1 to 0 at time $t$ and then the policy is otherwise unchanged. Since they prefer not to directly model the environment, their method cannot be used to estimate the average treatment effect or other related quantities under a different policy. Finally, Bojinov and Shephard \cite{BS17} consider estimating causal effects when a single unit is subject to time-varying treatments. As they consider only a single unit, the causal quantities they can estimate is limited. In particular, their estimands depends on the observed path and they focus on the following causal estimand. Conditioned on an observed path at time $t - 1$, what is the average causal effect of switching treatment from 1 to 0 at time $t$ on the outcome at time $t + p$.

In recent years, there have been several applications of tensor methods; e.g., for learning mixture models \cite{HK13} and learning topic models \cite{Ana+12}, and so forth. Our main optimization problem is weighted tensor completion problem, which tries to estimate the missing entries of a tensor from the observed entries. Although several algorithms have been proposed for the problem of tensor completion \cite{BM16; YZ16; MS18}, the problem of weighted tensor completion is relatively unexplored. We convert the weighted tensor completion problem into a weighted tensor approximation problem. A special case of this problem, weighted matrix completion, is intractable in general. Srebro and Jaakkola \cite{SJ03} developed an alternating minimization algorithm for this problem. Razenshteyn, Song, and Woodruff \cite{RSW16} developed a provably efficient algorithm for this problem using sketching techniques. Subsequently, Song, Woodruff, and Zhong \cite{SWZ17} generalized their methods for the weighted tensor approximation problem.
5.2 Model

For $t = 1, \ldots, T$, $A_{i,t}$ denotes the treatment assigned to subject $i$ at time $t$, and $X_{i,t}$ denotes the covariate of subject $i$ at time $t$. For $t = 1, \ldots, T$, $Y_{i,t}$ denotes the observed outcome for unit $i$ at time $t$ and depends on the history of the treatments assigned to agent $i$ at time $t$, and the history of covariates up to but excluding time $t$. We use the following notation for a sequence of treatments. $A_{i,t':t''}$ denotes the sequence of treatments from $t'$ to $t''$ i.e. $A_{i,t'}, A_{i,t'+1}, \ldots, A_{i,t''}$. A sequence of outcomes, $Y_{i,t':t''}$, and a sequence of covariates $X_{i,t':t''}$ is defined analogously. We will use lowercase variables to denote particular realizations of the random variables, e.g., $a_{i,t}$ denotes a realization of $A_{i,t}$, the random variable denoting treatment of agent $i$ at time $t$. The same notation applies for the outcomes, and the covariates.

The directed acyclic graph (Figure 5.1) represents the relationship among different variables. For each $i$ and $t$, a policy determines $A_{i,t}$, i.e., the treatment to be assigned. In general, such a policy can be randomized and dynamic, such that the action $A_{i,t}$ depends on the history up to time $t$. In such a case, we will write $\Pr[A_{i,t} = a_{i,t}|a_{i,1:t-1}, x_{i,1:t-1}, y_{i,1:t-1}]$ for the probability assigned to the treatment $a_{i,t}$ given past treatment sequence of length $t-1$, $a_{i,1:t-1}$, the covariate sequence of length $t-1$, $x_{i,1:t-1}$, and the past outcome sequence of length $t-1$, $y_{i,1:t-1}$. To

\footnote{We assume the policy is known i.e. the conditional probabilities of the treatment assignments are known. We leave the problem of estimating these probabilities from the data as future work. In particular, it will be interesting to develop a doubly robust estimator which is robust to misspecification in either the treatment model or the outcome model.}

Figure 5.1: A directed acyclic graph describing the model for agent $i$
give a concrete example, Robins, Hernan, and Brumback [RHB00] consider a clinical trial setting with HIV patients where the outcomes are the health status and the decision to give a particular drug at a time depends on the patient’s CD4 count at that time. In full generality, the outcome at any time might also depend on the entire treatment history, but we make the following assumption about the outcome for any agent, say $i$.

Assumption 1. The outcome at time $t$, $Y_{i,t}$ depends only on the past treatment history of length $k$, $A_{i,t-k+1:t}$.

5.2.1 Outcome Model

Since the outcome at time $t$ for agent $i$, $Y_{i,t}$, may depend on the past treatment history of length $k$, there are $2^k$ potential outcomes for each agent $i$ and each time $t$. This implies that there are $N \times T \times 2^k$ potential outcomes out of which we observe only $N \times T$ potential outcomes. We now define the outcome model. There is a tensor $T$ of dimension of $N \times T \times 2^k$, such that the outcome for subject $i$ at time $t$ is

$$Y_{i,t} = T[i,t, A_{i,t-k+1:t}] + \xi_{i,t},$$

where $\xi_{i,t}$ are iid Gaussian random variables with zero mean and unit variance. Equation (5.1) says that the potential outcomes are indexed by the subject $i$, time period $t$, and the treatment history of length $k$, $A_{i,t-k+1:t}$. The variable $k$ controls the dependence of the outcome on past sequence of treatments. In general, $k$ can be arbitrarily long. However, we need to assume that the $k$ is bounded from above by the larger of logarithm of $N$ and logarithm of $T$ in order to estimate the potential outcomes. Otherwise, the number of missing outcomes grows at a rate larger than the number of observed outcomes, and we are unable to estimate all the missing outcomes.

---

2In some scenarios, potential outcomes can exhibit structure, and the number of distinct potential outcomes we need to estimate can be smaller. As an example, suppose that a subject’s response at time $t$ depends only on how many times she was given the treatment in the last $k$ rounds. This implies, for each $i$ and $t$, there are only $k + 1$ distinct potential outcomes. Our algorithm need not be aware of such a structure, and the results are stated without this requirement. Introducing this assumption would only lead to improved, positive results.

3This seems reasonable in settings with time-varying treatments, e.g., the number of trips taken by a rider will depend on his coupons for the past couple of months, but not on whether she received coupons several years back.
5.2.2 Sequentially Randomized Experiment

In this paper, we restrict our attention to the case when there are no unobserved confounders. These are variables that both affect the treatment and the outcome, but are not recorded in the covariates. This assumption is true for the ride-sharing example, where the platform determines whether to give a rider a coupon or not based on the rider’s history. However, not all variables that affect a country’s GDP and political situation can be recorded, and this example has unobserved confounders.

We formalize this requirement of no unobserved confounders by time-varying generalizations of standard properties in the literature on causal inference, namely consistency (the observed outcome is the same as the potential outcome corresponding to the treatment applied), and ignorability (the treatment is independent of the potential outcomes conditioned on the covariate), and positivity of treatment assignment. Let $A_{i,t}^{obs}$ denote the observed outcome, and $A_{i,t}(\cdot)$ denote the corresponding random variable dependent on the history. The same notation holds for the outcomes and the covariates. We define the following properties:

1. Consistency: The observed data $(Y_{i,1}, A_{i,1}, X_{i,1}, Y_{i,2}, A_{i,2}, X_{i,2}, \ldots)$ is equal to the potential outcomes as follows. For every history $h_{i,t} = (a_{i,1:t}, x_{i,1:t}, y_{i,1:t-1})$, we have $Y_{i,t}^{obs} = Y_{i,t}(h_{i,t}) = Y_{i,t}(a_{i,t-k+1:t}), X_{i,t+1}^{obs} = X_{i,t+1}(h_{i,t})$, and $A_{i,t+1}^{obs} = A_{i,t+1}(h_{i,t})$.

2. Sequential Ignorability: For each $t$, the potential outcomes are independent of the treatment conditioned on the history at time $t$, i.e.,

$$Y_{i,t} \independent A_{i,t} | A_{i,1:t} = a_{i,1:t}, X_{i,1:t} = x_{i,1:t}, Y_{i,1:t-1} = y_{i,1:t-1} \quad (5.2)$$

3. Positivity: There exists a $\delta > 0$ such that for each $a_{i,1:t-1}, x_{i,1:t-1}, y_{i,1:t-1}$, we have

$$\delta < \Pr [a_{i,t} | a_{i,1:t-1}, x_{i,1:t-1}, y_{i,1:t-1}] < 1 - \delta$$

Consistency maps the observed outcomes to the potential outcomes. In particular, the outcome observed at time $t$, $Y_{i,t}^{obs}$, is completely determined by the past treatment history of length $k$, i.e., there are no additional factors such as the subject’s motivation that affects both the actual treatment assignment and the outcome. If the treatment at time $t$, $A_{i,t}$ is chosen based on the history up to time $t$, then sequential ignorability automatically holds [Bor+18]. On the other
hand, in an observational study, we must assume there are no unmeasured confounders for sequential ignorability to hold. If the policy systematically violates positivity, then it might be that some units do not get a particular treatment at all, and it would be impossible to estimate the outcome model.

5.2.3 Quantities to Estimate

The literature on causal inference has proposed various quantities to estimate in a setting with time-varying treatments. In the introduction, we talked about the average treatment effect over the treated (ATET). For any given assignment \( \{a_{i,t}\}_{i,t} \) we define ATET to be the average effect of the treatment over the units that actually received the treatment under \( \{a_{i,t}\}_{i,t} \). Formally,

\[
\text{ATET} = \frac{1}{|\{(i,t) : a_{i,t} = 1\}|} \sum_{(i,t):a_{i,t}=1} \mathbb{E}[Y_{i,t}(a_{i,t-k+1:t})] - \mathbb{E}[Y_{i,t}(a_{i,t-k+1:t-1}, 0)]
\]

According to the outcome model specified in (5.1), this becomes

\[
\text{ATET} = \frac{1}{|\{(i,t) : a_{i,t} = 1\}|} \sum_{(i,t):a_{i,t}=1} T[i, t, a_{i,t-k+1:t}] - T[i, t, (a_{i,t-k+1:t-1}, 0)], \quad (5.3)
\]

and can be computed easily once we have an estimate of the tensor \( T \). We can also generalize ATET by considering the effect of switching from one history \( h_1 \) to another history \( h_2 \) of length at most \( k \). \( \text{ATET}(h_1, h_2) \) is defined as:

\[
\frac{1}{|\{(i,t) : a_{i,t-|h_1|-1:t-1} = h_1\}|} \sum_{(i,t):a_{i,t-|h_1|-1:t-1}=h_1} \mathbb{E}[Y_{i,t}(a_{i,t-k+1:t})] - \mathbb{E}[Y_{i,t}(a_{i,t-k+1:t-|h_2|}, h_2)]
\]

and it is straightforward to write \( \text{ATET}(h_1, h_2) \) using the outcome model specified in (5.1).

Another quantity of interest is the \textit{contemporaneous effect of treatment on} \( Y_{i,t} \) [BG18b]. This quantity measures the effect of treatment \( A_{i,t} \) on \( Y_{i,t} \), i.e., what would happen if we let the treatment sequence be as it would have been under the given policy, and then switch the treatment at time \( t \) from 0 to 1:

\[
\text{CET}_{i,t} = \mathbb{E}_{A_{i,1:t-1}} \left[ Y_{i,t}(A_{i,1:t-1}, 1) \right] - \mathbb{E}_{A_{i,1:t-1}} \left[ Y_{i,t}(A_{i,1:t-1}, 0) \right] | A_{i,1:t-1}
\]

Here the sequence \( A_{i,1:t-1} \) in the outer expectation is drawn according to the given policy.
According to the outcome model specified in (5.1), the quantity $\text{CET}_{i,t}$ becomes

$$
\text{CET}_{i,t} = \sum_{a_{i,t-k+1:t-1}} \Pr[a_{i,t-k+1:t-1}] \left\{ T(i, t, (a_{i,t-k+1:t-1}, 1)) - T(i, t, (a_{i,t-k+1:t-1}, 0)) \right\}
$$

(5.4)

In general, one might be interested in distant effects of the treatment on the outcome. For example, the $j$ step lagged effect considers the effect of switching the treatment at time $t$, $A_{i,t}$ from 0 to 1 on the outcome after $j$ steps. As long as we can estimate the tensor $T$ in our outcome model, and we are aware of the treatment policy, we can estimate the various quantities of interest.

### 5.2.4 Marginal Structural Models

Our work builds on the marginal structural models, proposed by Robins, Hernan, and Brumback [RHB00]. At each time $t$, for every possible sequence of treatments $a_{i,1:t}$, MSMs define the following model of the corresponding potential outcome.

$$
E[Y_{it}(a_{i,1:t})] = g(a_{i,1:t}, \beta)
$$

(5.5)

Here $g$ is the link function, usually chosen to be either a linear function or a logistic function.

The standard maximum likelihood based estimator of $\beta$ will be biased. Robins [Rob00] showed that the parameter can be estimated in an unbiased way through an inverse probability of treatment weighting (IPTW) approach. Suppose the observed data is given as $\{a_{i,t}, x_{i,t}, y_{i,t}\}_{i,t}$. Then consider the following weight for each agent $i$ and each time period $t$:

$$
sw_{it} = \prod_{s=1}^{t} \frac{\Pr[a_{i,s} | a_{i,1:s-1}, x_{i,1:s-1}, y_{i,1:s-1}]}{\Pr[a_{i,s} | a_{i,1:s-1}]} 
$$

The denominator of each term is the probability of the corresponding treatment given the history up to that point. The numerator of each term is the marginal probability of the corresponding treatment conditioned only on the past sequence of treatments and is used to stabilize the weights. Now if we compute a maximum likelihood estimator where the observation of subject $i$ at time $t$ is weighted by $sw_{it}$, then $\beta$ can be identified. If we know the policy, we can directly compute the marginal probabilities and get the weights. When the policy is unknown the probabilities are estimated from the data and substituted to compute
5.3 Estimation

The goal is to design an unbiased and consistent estimator $\hat{T}$ of the $N \times T \times 2^k$ tensor $T$. We will assume that the tensor $T$ has low rank. $T$ has rank $r$ if there exist vectors $\{u_i\}^r_{i=1}$, $\{v_i\}^r_{i=1}$ and $\{w_i\}^r_{i=1}$ ($u_i \in \mathbb{R}^N$, $v_i \in \mathbb{R}^T$, $w_i \in \mathbb{R}^B$) such that $T = \sum_{i=1}^r u_i \otimes v_i \otimes w_i$ and $r$ is the smallest integer such that $T$ can be written in this form. Here $u_i \otimes v_i \otimes w_i$ denotes the outer-product of the three vectors $u_i$, $v_i$, and $w_i$ with entries $u_i \otimes v_i \otimes w_i(a, b, c) = u_i(a) \times v_i(b) \times w_i(c)$. Without loss of generality, we can assume that the tensor $T$ is written in the following form, where each of the vectors $u_i$, $v_i$, and $w_i$ are normalized.

$$T = \sum_{i=1}^r \lambda_i u_i \otimes v_i \otimes w_i \quad (5.6)$$

We use $\lambda_i(T) = \lambda_i$ to denote the $i$-th singular value of $T$. For $p = 1, \ldots, B$, let $O_p$ be the set of observations that lead to the realization of history corresponding to the $p$-th slice. Formally, $O_p = \{(i, t) : A_{i,t-k+1:t} = p\}$. Then we propose to solve the following optimization problem:

$$\min_{T \in \mathbb{R}^{N \times T \times 2^k}, \text{rank}(T) \leq r} \frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i, t) \in O_p} w_{i,t} (Y_{i,t} - T(i, t, p))^2. \quad (5.7)$$

The weights $w_{i,t}$ are defined as:

$$w_{i,t} = \frac{\prod_{s=t-k+1}^t \frac{\Pr[a_{i,s} | a_{i,t-2k+1:s-1}]}{\Pr[a_{i,s} | a_{i,t-2k+1:s-1}, y_{i,t-2k+1:s-1}]} }{\prod_{s=t-k+1}^t \frac{\Pr[a_{i,s} | a_{i,t-2k+1:s-1}]}{\Pr[a_{i,s} | a_{i,t-2k+1:s-1}, y_{i,t-2k+1:s-1}]} }. \quad (5.8)$$

For each term, the denominator denotes the probability of the treatment given the history from time $t - 2k + 1$ to that time. The numerator can be any marginal probability not involving the outcome variables. Note that we omit the covariates from the denominator as the outcome does not depend on them. But, why are we interested in the optimization problem eq. (5.7)? The objective function is the weighted log-likelihood given tensor $T$, and we prove next that if we could solve this problem exactly, the corresponding estimator will be consistent. We make some additional assumptions:
Asn.1 Bounded Singular Value: For each $N$ and $T$, each of the $r$ singular values of $T_{N,T}^*$ are bounded, i.e. $\|T_{N,T}^*\|_* = \max_i |\lambda_i(T)| \leq L$ for some $L$.

Asn.2 Decaying Covariance: There exists a constant $\gamma \geq 1$, such that for all $t' > t + 3k$, and for all sequences of treatments $a_{i,1:t}$ and $\tilde{a}_{i,t'-k+1:t'}$ we have

$$1 - \varepsilon \leq \frac{\Pr[\tilde{a}_{i,t'-k+1:t'}|a_{i,1:t}]}{\Pr[a_{i,t'-k+1:t'}]} \leq 1 + \varepsilon$$

for $\varepsilon = O((t' - t)^{-\gamma}).$

The first assumption implies that each entry of the tensor is bounded between $-L$ and $L$. The second assumption implies that the treatments chosen at two time periods that are far apart, are almost independent.

5.3.1 Consistency

For any $N$ and $T$, we assume that the data is generated from an underlying tensor $T_{N,T}^*$. We will also write $\hat{T}_{N,T}$ to denote the solution to eq. (5.7). Consider the weighted log-likelihood function:

$$L_{N,T}(T_{N,T}) = \frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T w_{i,t} \log \Pr[Y_{i,t}|T_{N,T}].$$

The estimate $\hat{T}_{N,T}$ maximizes $L_{N,T}(T_{N,T})$ over all possible choices of $T_{N,T}$. Our goal is to show that with high probability, $\|\hat{T}_{N,T} - T_{N,T}^*\|_2/\sqrt{NT}$ converges to zero as $N$ and $T$ increases.

We normalize the difference in norm by both $N$ and $T$. This is necessary, as with increasing $N$ and $T$ the number of parameters we are estimating also grows.

**Theorem 5.1.** Suppose $T_{N,T}^*$ exists for all $N$ and $T$. Then

- If $k \leq O\left(\log_2(1-\delta)/\delta N\right)$, then for any $\varepsilon > 0$, $\Pr\left[\frac{\|\hat{T}_{N,T} - T_{N,T}^*\|_2}{\sqrt{NT}} > \varepsilon\right] \to 0$ as $N \to \infty$.

- If Assumption Asn.2 holds, and $k \leq \max\{O\left(\log_4(T/\log T)\right), O\left(\log_2(1-\delta)/\delta T\right)\}$, then $\Pr\left[\frac{\|\hat{T}_{N,T} - T_{N,T}^*\|_2}{\sqrt{NT}} > \varepsilon\right] \to 0$ as $T \to \infty$.

The full proof is given in section D.1.1. Here we sketch the main challenges. The proof follows the ideas presented in Newey and McFadden [NM94], but there are some subtle
differences. Unlike the traditional maximum likelihood estimation, we are not estimating a fixed parameter. As either \( N \) or \( T \) increases, we are estimating a sequence of tensors increasing in either \( N \) or \( T \). This is why we prove that the normalized distance between \( T^*_{N,T} \) and \( \hat{T}_{N,T} \) goes to zero, instead of the actual L2 distance. There are two more challenges in the proof. First, the parameter space \( \Theta_{N,T} = \{ T \in \mathbb{R}^{N \times T \times 2^k} : \text{rank}(T) \leq r \} \) need not be a closed set, as we can have a sequence of rank \( r \) tensors converging to a rank \( r + 1 \) tensor. However, the concavity of the log-likelihood function in \( T \) helps us to circumvent this problem. Second, the standard way to prove the consistency of the maximum likelihood estimation is to consider a neighborhood around the true parameter, say \( B \). Then there will be a gap of \( \varepsilon \) between the maximum over \( B \) and the maximum outside of \( B \), and for large number of samples the gap between the objective value of the true parameter and the estimate will be less than \( \varepsilon \), and the estimate will be inside the neighborhood \( B \). However, in our case, the gap \( \varepsilon \) is also changing with \( N \) and \( T \) as the entire parameter space is changing, and it might be possible that this gap goes to zero with increasing \( N \) and \( T \). However, we can provide a lower bound on the gap in terms of the radius of the neighborhood and the parameters \( N \) and \( T \), and this helps to complete the proof.

5.3.2 Solving Tensor Completion

In this section, we focus on solving the weighted tensor completion to estimate the underlying tensor \( T^*_{N,T} \). We proceed in two steps. First, we convert the weighted tensor completion problem to a weighted tensor approximation problem with an additive error that goes to zero as either the number of units \( N \) or the number of time intervals \( T \) increases to infinity. Then we provide a \((1 + \varepsilon)\)-approximation to the weighted tensor approximation problem under reasonable assumptions on the policy generating the treatment assignment. A combination of these two steps gives us an approximate solution to the original objective function defined in eq. (5.7).

Recall that \( O_p \) refers to all observations for which we observe the counterfactual outcome corresponding to the \( p \)-th slice, i.e., \( O_p = \{(i,t) : A_{i,t-k+1:t} = p\} \). Consider the objective function
defined in (5.7):

\[
\frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t} (Y_{i,t} - T(i,t,p))^2
\]

\[
= \frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t} (Y_{i,t}^2 - 2Y_{i,t}T(i,t,p) + T(i,t,p)^2).
\]

Since we are optimizing over the tensor \(T\), we can drop the first term above, and consider the following objective:

\[
\frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t} T(i,t,p)^2.
\]

The main idea to convert this objective into a tensor approximation problem is to replace the second term by its population variant and define a weight tensor so that the first sum is defined over all the entries in \(T\). Let \(\Pr[(i,t) \in O_p]\) be the marginal probability that the underlying policy selects slice \(p\) for agent \(i\) at time \(t\). The supplementary material proves that the expected value of the second term is

\[
\frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{p=1}^{2^k} \Pr[(i,t) \in O_p] T^2(i,t,p),
\]

a weighted norm of \(T\). So we replace the second term above by its corresponding population variant,

\[
\frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} -2w_{i,t} Y_{i,t} T(i,t,p) + \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{p=1}^{2^k} \Pr[(i,t) \in O_p] T^2(i,t,p).
\]

Now let us define the following tensor:

\[
Y_w(i,t,p) = \begin{cases} 
\frac{w_{i,t} Y_{i,t}}{\Pr[(i,t) \in O_p]} & \text{if } (i,t) \in O_p \\
0 & \text{otherwise}
\end{cases}
\]

and the “weight” tensor, \(W(i,t,p) = \sqrt{\Pr[(i,t) \in O_p]}\). This leads to the following form of objective (5.11):

\[
- \frac{2}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{p=1}^{2^k} (W(i,t,p))^2 Y_w(i,t,p) T(i,t,p) + \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{p=1}^{2^k} \Pr[(i,t) \in O_p] T^2(i,t,p).
\]

Finally, we include additional terms involving the tensor \(Y_w\) to make the objective function (5.12)
a square:
\[
\frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{p=1}^{2^k} (W(i, t, p))^2 \{ Y_w(i, t, p) - T(i, t, p) \}^2 = \frac{1}{NT} \| Y_w - T \|_W^2. \tag{5.13}
\]

This leads us to the following tensor approximation problem, instead of the tensor completion problem in (5.7):
\[
\min_{T \in \mathbb{R}^{N \times T \times 2^k}, \text{rank}(T) \leq r, \| T \|_i \leq L} \frac{1}{NT} \| Y_w - T \|_W^2. \tag{5.13}
\]

Objective (5.13) computes a weighted low rank approximation of \( Y_w \). Let \( \tilde{T}_{N,T} \) be the solution to (5.13). We first show that this estimator approximately optimizes the original objective (5.7). Let \( \text{OPT} \) be the optimal value of (5.7).

**Lemma 5.1.** Suppose \( T^*_N, T \) exists for all \( N \) and \( T \). Then

- If \( k \leq O \left( \log(1 - \delta)/\delta \right) N \), then
  \[
  \frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t} \left( Y_{i,t} - \tilde{T}_{N,T}(i, t, p) \right)^2 \lesssim \text{OPT} + O \left( \frac{L^2}{NT^4} \right)
  \]
  w.p. at least \( 1 - \exp \left( -N^{1/4} \right) \).

- If \( k \leq O \left( \log(1 - \delta)/\delta \right) T \) and assumption Asn.2 holds, then
  \[
  \frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t} \left( Y_{i,t} - \tilde{T}_{N,T}(i, t, p) \right)^2 \lesssim \text{OPT} + O \left( \frac{L^2}{NT^4} \right)
  \]
  w.p. at least \( 1 - O \left( \log T/NT \right) \).

See D.1.2 for a proof. When \( N \) is fixed and \( T \) is increasing, the additive error holds with probability \( 1 - O \left( \log T/NT \right) \) instead of probability \( 1 - \exp \left( -T \right) \). This is because we do not have independence in the treatments for different values of \( t \) for a given subject \( i \). However, assumption Asn.2 helps us to bound the variance of the estimated norm of the tensor and thereby bound the probability of failure by \( O \left( \log T/NT \right) \).

### 5.3.3 Solving Low-Rank Tensor Approximation

Now we focus on solving problem (5.13). Although weighted low-rank approximation of a tensor is in general intractable, we first provide an efficient approximation algorithm for problem 5.13 using techniques derived by [SWZ17].

---

\(^5\)For a tensor \( T \), let \( \| T \|_W^2 \) denote the weighted Euclidean norm, i.e.
\[
\| T \|_W^2 = \sum_{i,j,k} W^2(i,j,k) T^2(i,j,k).
\]
Song, Woodruff, and Zhong [SWZ17] show that there is an algorithm that takes as input a tensor $A \in \mathbb{R}^{n \times n \times n}$, a weight tensor $W \in \mathbb{R}^{n \times n \times n}$, and outputs a tensor $A'$ of rank $r$ such that 

$$\|A - A'\|_W \leq (1 + \varepsilon) \min_{\text{rank}(B) \leq r} \|A - B\|_W^2.$$ 

The authors consider the case when the weight tensor $W$ has $s$ distinct faces in two dimensions (e.g. $s$ distinct rows, and columns). Then their algorithm runs in time $\text{nnz}(A) + \text{nnz}(W) + n2^d(s^2r^2/\varepsilon)$ time, where $\text{nnz}(A)$ is the number of nonzero entries in $A$. The algorithm works by choosing a sketching matrix for each of the three unfoldings of the tensor. The sketching matrices project the rows, columns, and tubes of the tensor to a low-dimensional space. This allows to convert the tensor approximation problem to a polynomial system verification problem in a low-dimensional space.

We want to find a rank $r$ approximation of tensor $Y_w \in \mathbb{R}^{N \times T \times 2^k}$. There are two challenges. First, the algorithm proposed in Song, Woodruff, and Zhong [SWZ17] works with tensors whose dimensions across the three axes are the same. However, the algorithm can be easily generalized so that it works for tensors of arbitrary dimensions by choosing the sketching matrices to be of appropriate dimensions. Second, we want to enforce an additional constraint that the singular values are bounded between $-L$ and $L$. This can be handled by introducing $r$ additional constraints in the polynomial system verifier of the algorithm in Song, Woodruff, and Zhong [SWZ17]. Algorithm 5.1 details the full procedure. It takes as input a tensor $T \in \mathbb{R}^{N \times T \times 2^k}$, a weight tensor $W \in \mathbb{R}^{N \times T \times 2^k}$ and approximately solves $\min_{\text{rank}(V) \leq r} \|T - V\|_W^2$. Since we are guaranteed that $W$ has $s$ distinct rows and $s$ distinct columns, the number of distinct tubes of $s$ is at most $S = 2^{O(s \log s)}$.

Algorithm 5.1 closely follows algorithm G.4 in [SWZ17] with modifications to handle asymmetric tensors and additional constraint on the bound for the largest singular value. It chooses three sketching matrices of appropriate dimension to solve the original low-rank approximation problem in a low-dimensional space. The main idea is that the entries of $\hat{U}_1$ can be represented as polynomials of the variables for $i = 1$ to $s$ (line 10). This is possible because the weight matrix has $s$ distinct rows and columns, which implies that it’s flatenning along the rows has $s$ distinct faces. The same holds for $\hat{U}_2$. However, this need not be true for $\hat{U}_3$, so they are represented through $S$ distinct denominators (line 16). With this setup [SWZ17] shows that the number of variables in the polynomial system verifier is $O\left(\frac{r^2s}{\varepsilon}\right)$ and the number
Algorithm 5.1: Weighted Low Rank Tensor Approximation

Input: $T \in \mathbb{R}^{NT \times T \times 2^k}$, weight tensor $W \in \mathbb{R}^{NT \times T \times 2^k}$, rank $r$, rank of weight tensor $s$, and $\varepsilon$.
Output: Tensor $V$ of rank $k$ such that
\[
\|T - T\|_W^2 \leq (1 + \varepsilon) \min_{V: \text{rank}(V) \leq r, \|V\|_2 \leq L} \|T - V\|_W^2.
\]

1. for $j = 1$ to 3 do
   2. $s_j \leftarrow O(r/\varepsilon)$
3. Choose three sketching matrices $S_1 \in \mathbb{R}^{T2^k \times s_1}, S_2 \in \mathbb{R}^{N2^k \times s_2}$, and $S_3 \in \mathbb{R}^{NT \times s_3}$
4. for $j = 1$ to 2 do
   5. /* Omitting the third dimension */
   6. for $i = 1$ to $s_j$ do
   7. Set $\hat{U}_j^i = T_j^i D_j W_j S_j P_j^T (P_j P_j^T)^{-1}$
8. for $i = 1$ to $S$ do
   9. /* Representing the third dimension */
   10. Set $\hat{U}_3^i = T_3^i D_3 W_3 S_3 P_{3,j}^T (P_{3,j} P_{3,j}^T)^{-1}$
11. Form $\|W \cdot (\hat{U}_1 \otimes \hat{U}_2 \otimes \hat{U}_3 - T)\|_F^2$
12. for $i = 1$ to $r$ do
   13. Add constraint $\|\hat{U}_1^i\|_2, \|\hat{U}_2^i\|_2, \|\hat{U}_3^i\|_2 \leq L$
14. return $U_1 \otimes U_2 \otimes U_3$

of constraints is $2s + S$. In line 18, we add additional $r$ constraints. So the total number of constraints is $2s + r + 2^{O(s \log s)}$ and the total number of variables is $O(r^2 s / \varepsilon)$. Moreover, the degree of the new constraints in line 18 is at most $\text{poly}(r, s, S)$. A polynomial system can be verified in time

\[
\text{(\# max degree of any polynomial)} \cdot \text{\# number of variables} \cdot \text{poly(input size of max coefficient)}
\]

In our case, this takes time

\[
\left(\text{poly}(r, s) \text{poly}(2^{O(s \log s)})\right)^{O(r^2 s / \varepsilon)} \cdot \text{poly(max}(N, T, 2^k), L)
= \left(\text{poly}(r, s)2^{O(s \log s)}\right)^{O(r^2 s / \varepsilon)} = 2^{\tilde{O}(r^2 s / \varepsilon)}
\]

Recall that we want to compute a low-rank approximation of the tensor $Y_w \in \mathbb{R}^{NT \times T \times 2^k}$. Although $\text{nnz}(Y_w) = NT$, positivity implies that the number of nonzero entries in $W$ is $\text{nnz}(W) = NT 2^k$. Therefore, the resulting algorithm runs in time $O\left(NT 2^k + \text{max}\{N, T, 2^k\}2^{\tilde{O}(s^2 / \varepsilon)}\right)$
and outputs a tensor $\tilde{T}_{N,T}$ such that $\|Y_w - \tilde{T}_{N,T}\|_W^2 \leq (1 + \varepsilon) \min_{T \in \mathbb{R}^{N \times T \times 2^k}, \text{rank}(T) \leq r} \|Y_w - T\|_W^2$ with probability at least $9/10$. The next lemma shows that $\tilde{T}_{N,T}$ approximately optimizes our original objective.

**Lemma 5.2.**

- If $k \leq O\left(\log(1-\delta)/\delta\right) N$, then $\frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t} \left( Y_{i,t} - \tilde{T}_{N,T}(i,t,p) \right)^2 \leq (1 + \varepsilon) \text{OPT} + O\left(\frac{L^2}{N^2}\right)$ with probability at least $4/5$.

- If $k \leq O\left(\log_{T/(1-\delta)} T\right)$ and assumption 5.9 holds, then $\frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t} \left( Y_{i,t} - \tilde{T}_{N,T}(i,t,p) \right)^2 \leq (1 + \varepsilon) \text{OPT} + O\left(\frac{L^2}{T^2}\right)$ with probability at least $4/5$.

**Distinct Faces of the Weight Matrix**

Recall that we need the weight matrix $W$ to have $s$ distinct faces in two dimensions, where the weight matrix $W$ is defined as $W(i,t,p) = \sqrt{\Pr[(i,t) \in O_p]}$. If the underlying policy satisfies the following two assumptions, then the matrix $W$ has $s$ distinct faces along the two dimensions.

1. There are $s$ groups of subjects such that the policy treats all the subjects in a group identically.

2. There are $s$ groups of time periods such that for any two time $t$ and $t'$ belonging to the same group we have the same marginal probabilities across all the subjects $(\Pr[(i,t) \in O_p] = \Pr[(i,t') \in O_p]) \forall i, p$.

These two assumptions together imply that $W$ has $s$ distinct faces in two dimensions, and allows an efficient $(1 + \varepsilon)$-multiplicative approximation of problem 5.13.

**Projected Gradient Descent**

We now provide a simple algorithm for the weighted tensor approximation problem (5.13) based on projected gradient descent. Algorithm 5.2 repeatedly applies two steps. Line 5 computes a gradient step to compute the new tensor $T_u$. However, the tensor $T_u$ might not be of rank $r$, so line 6 computes a projection of tensor $T_u$ into the space of tensors of rank $r$. As
the projection step is a standard rank $r$ approximation of a tensor, we use the parafac method from the TensorLy package [Kos+18] for this step.

**Algorithm 5.2: Weighted Low Rank Tensor Approximation**

- **Input:** Tensor $S \in \mathbb{R}^{N \times T \times k}$, weight tensor $W \in \mathbb{R}^{N \times T \times k}$, rank $r$, and number of iterations $Iter$.
- **Output:** Tensor $T \in \mathbb{R}^{N \times T \times k}$ of rank $r$.

1. Initialize $T$
2. for $j = 1$ to $Iter$ do
   3. $T_u \leftarrow T + \lambda W^2 (S - T)$
   4. $T \leftarrow $ Project($T_u, r$)
   5. if Relative Change in Loss $\leq \varepsilon$ then
      6. return $T$
3. return $T$

**Simulation**

We now evaluate the effectiveness of Algorithm 5.2 through a simulation. We consider three worlds, *square* ($N = 50$ and $T = 50$), *fat* ($N = 20$ and $T = 100$), and *thin* ($N = 100$ and $T = 20$). We consider $k = 4$ i.e. the outcome at any point in time $t$ depends on the treatment history of length 4.

For each world, we first fix a tensor $T$ of rank 6 by two steps. First, we choose the vectors $\{v_i\}_{i=1}^T, \{v_i\}_{i=1}^T$ and $\{w_i\}_{i=1}^T$ by selecting each entry uniformly at random from the interval $[0, 1]$ and then normalizing the vectors. Second, we select the singular values $\{\lambda_i\}_{i=1}^T$ uniformly at random from the interval $[10, 20]$. Having fixed this tensor, we generate the data i.e. the treatment assignment $\{A_{i,t}\}_{i,t}$ and the outcome $\{Y_{i,t}\}_{i,t}$ according to four policies.

1. **Simple:** The treatment at every period is either 0 or 1 with equal probability.
2. **Complex-2:** The policy counts the number of ones in the previous two rounds. If the count is zero then $A_{i,t}$ is 1 with probability 0.75. If it is one (resp. two) then $A_{i,t}$ is 1 with probability 0.5 (resp. 0.25).
3. **Complex-3:** This policy is same as Complex-2 except that now we count the number of ones in the previous three rounds. The probability that $A_{i,t}$ is 1 is adjusted accordingly.
4. **Complex-4:** We now count number of ones in the previous four rounds and the probabilities are adjusted accordingly.

Since algorithm 5.2 is oblivious of the true parameters $r$ and $k$, we aim to determine how sensitive it is to the choice of the assumed rank parameter $r$ and the assumed length of the history $k$. Figure 5.2 plots the estimation error for various choices of assumed $r$ values for the four different types of policies. We observe that the algorithm is less sensitive to change in the assumed rank as the amount of change in error is bounded by at most 0.03. Figure 5.3 plots the estimation error for various choices of assumed $k$ values for the four different types of policies. Note that when $k = 1$, our algorithm is basically weighted matrix completion. And as $k$ increases the performance improves across all the worlds and different types of policies, since it allows for estimation of more counterfactual outcomes.

Finally, we fit traditional MSM [Rob00] at every time period. Since the outcome distribution
Figure 5.3: Average absolute error in estimating ATET through weighted tensor approximation (Algorithm 5.2), with true tensor model generated with rank $r = 6$ and temporal dimension $k = 4$, and four different policies. Error bars show standard errors from repeating each simulation i.e. regenerating the treatment assignment and outcomes 20 times.

is normal, we use a linear function as the link function, i.e., $g(a_{i;1:t}, \beta) = \sum_{t'} a_{i,t'} \beta_{t'}$ in eq. 5.5. Table 5.1 compares the marginal structural models with our algorithm 5.2. Out of 12 possible combinations of world and policy, our method beats MSM on six combinations. MSM consistently does better on world “thin”. This is because MSM performs well when there is not enough heterogeneity in the world ($r = 6$) and $N$ is large compared to $T$. In this case, the approximation error in tensor approximation becomes significant. However, as proven in lemma 5.1, this error vanishes with large $N$ and and even in the “thin” world, algorithm 5.2 will beat MSM with large $N$. 

132
### Table 5.1: Comparison of Marginal Structural Models and algorithm 5.2 across three different worlds and four different policies. For each combination of world and policy, the top row shows the absolute error for estimating ATET by MSM and the bottom row shows the absolute error in estimating ATET for algorithm 5.2. Algorithm 5.2 beats MSM on six out of twelve cases, but MSM consistently does better on world “thin”. This is because MSM performs well when there is not enough heterogeneity in the world (r = 6) and N is large compared to T. In this case, the approximation error of algorithm 5.2 becomes significant, but it should vanish with large N.

<table>
<thead>
<tr>
<th>World</th>
<th>Fat</th>
<th>Square</th>
<th>Thin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>0.12</td>
<td>0.37</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>0.12</td>
<td>0.20</td>
<td>0.12</td>
</tr>
<tr>
<td>Complex-2</td>
<td>0.12</td>
<td>0.22</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>0.13</td>
<td>0.20</td>
<td>0.12</td>
</tr>
<tr>
<td>Complex-3</td>
<td>0.18</td>
<td>0.51</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>0.13</td>
<td>0.20</td>
<td>0.12</td>
</tr>
<tr>
<td>Complex-4</td>
<td>0.39</td>
<td>0.33</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>0.13</td>
<td>0.21</td>
<td>0.12</td>
</tr>
</tbody>
</table>

#### 5.4 Generalized Model

In this section, we generalize our outcome model so that the potential outcomes are also affected by the baseline covariates \( \{Z_i\}_{i=1}^N \). We propose the following outcome model. There is a tensor \( T \) of rank \( r \), dimension \( N \times T \times 2^k \) and a vector \( \beta^* \in \mathbb{R}^d \) such that the outcome for subject \( i \) at time \( t \) is

\[
Y_{i,t} = T[i, t, A_{i,t-k+1:t}] + Z_i^t \beta^* + \xi_{i,t},
\]

where \( \xi_{i,t} \) are iid Gaussian random variables with zero mean and unit variance. We add a linear function of the time-varying covariates to the outcome model proposed in eq. (5.1). Although we carry out the analysis under linearity, our method easily generalizes as long as the effect of baseline covariates is additive and we are aware of the functional form of the effect. We need two additional assumptions:

#### 5.4.1 Estimation

We propose to solve the following optimization problem:

\[
\min_{\beta \in \mathbb{R}^d, \mathbb{T} \in \mathbb{R}^{N \times T \times 2^k}, \text{rank} (\mathbb{T}) \leq r} \frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t} (Y_{i,t} - T(i, t, p) - Z_i^t \beta)^2. \tag{5.15}
\]
The weights $w_{i,t}$ are defined as:

$$w_{i,t} = \prod_{s=t-k+1}^{t} \frac{\Pr[a_{i,s} | a_{i,t-2k+1:s-1}]}{\Pr[a_{i,s} | a_{i,t-2k+1:s-1}, x_{i,t-2k+1:s-1}, y_{i,t-2k+1:s-1}]}.$$  

(5.16)

For each term, the denominator denotes the probability of the treatment given the history from time $t - 2k + 1$ to that time. The numerator can be any marginal probability not involving the outcome variables. Note that we now include the covariates in the denominator as the outcome depends on them. The objective function 5.15 is the weighted log-likelihood given tensor $\mathbf{T}$ and the parameter $\beta$ and we next prove that if we solve this problem exactly, the corresponding estimator will be consistent. However, we need two additional assumptions.

**Asn.3 Decaying Covariance**: There exists a constant $\gamma \geq 1$, such that for all $t' > t + 3k$, and for any history $H_{i,1:t} = (a_{i,1:t}, x_{i,1:t}, y_{i,1:t})$ and any sequence of length $k$ treatment $\tilde{a}_{i,t'-k+1:t'}$ and covariates $\tilde{x}_{i,t'-k+1:t'}$ we have,

$$1 - \varepsilon \leq \frac{\Pr[\tilde{a}_{i,t'-k+1:t'}, \tilde{x}_{i,t'-k+1:t'} | H_{i,1:t}]}{\Pr[\tilde{a}_{i,t'-k+1:t'}, \tilde{x}_{i,t'-k+1:t'}]} \leq 1 + \varepsilon$$

(5.17)

for $\varepsilon = O ((t' - t)^{-\gamma})$.

**Asn.4** There exist constants $\ell > 0$ and $0 \leq \mu \leq 1$ such that $\Pr[\|Z_i\|_2 \geq \ell] \geq \mu \forall i, t$.

Assumption **Asn.3** generalizes assumption **Asn.2** for the generalized model. And assumption **Asn.4** ensures that the covariates $Z_i$ has considerable effect on the outcome so that the parameter $\beta^*$ can be identified.

For any $N$ and $T$, we assume that the data is generated from an underlying tensor $\mathbf{T}_{N,T}$ and parameter $\beta^*$. We will also write $\hat{\mathbf{T}}_{N,T}$ to denote the solution to eq. (5.15). Consider the weighted log-likelihood function$^6$:

$$L_{N,T}(\mathbf{T}_{N,T}, \beta_{N,T}) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} w_{i,t} \log \Pr[Y_{i,t} | \mathbf{T}_{N,T}, \beta_{N,T}].$$

(5.18)

The estimate $\hat{\mathbf{T}}_{N,T}$ and $\hat{\beta}_{N,T}$ maximizes $L_{N,T}(\mathbf{T}_{N,T})$ over all possible choices of $\mathbf{T}_{N,T}$ and $\beta$.

Then the following holds.

---

$^6$The weighted log-likelihood function is also conditioned on the covariates, but we are omitting them for ease of notation.
Theorem 5.2. Suppose $T^*_N, T$ exists for all $N$ and $T$. Then

- If assumption Asn.4 holds and $k \leq O \left( \log_2(1-\delta)/\delta \right) N$, then for any $\epsilon > 0$,
  \[
  \Pr \left[ \frac{\|T_{N,T} - T^*_N\|_2}{\sqrt{NT}} > \epsilon \right] \rightarrow 0 \quad \text{and} \quad \hat{\beta}_{N,T} \xrightarrow{p} \beta^* \quad \text{as} \quad N \rightarrow \infty.
  \]

- If assumptions Asn.3 and Asn.4 hold, and $k \leq \max\{O \left( \log_4(T/\log T) \right), O \left( \log_2(1-\delta)/\delta \right) T \}$, then
  \[
  \Pr \left[ \frac{\|T_{N,T} - T^*_N\|_2}{\sqrt{NT}} > \epsilon \right] \rightarrow 0 \quad \text{and} \quad \hat{\beta}_{N,T} \xrightarrow{p} \beta^* \quad \text{as} \quad T \rightarrow \infty.
  \]

See D.1.3 for a proof.

5.4.2 Solving Tensor Completion

We now focus on solving the weighted tensor completion in eq. (5.15) to estimate the underlying tensor $T^*_N, T$ and parameter $\beta^*$. We proceed in two steps. First, we follow our previous approach and try to convert it to a weighted tensor approximation problem. However, because of the terms involving the covariates we don’t get an exact low rank approximation problem like eq. (5.13), but we will end up with an objective similar to eq. (5.13). Then we provide an algorithm for the new objective function using projected gradient descent. Although, we don’t have any guarantee on the performance of this algorithm, we find that the algorithm is easy to implement and fast compared to the sketching based algorithm 5.1 and we use the projected gradient descent based algorithm for experiments.

Recall that $O_p$ refers to all observations for which we observe the counterfactual outcome corresponding to the $p$-th slice, i.e., $O_p = \{(i,t) : A_{i,t-k+1:t} = p\}$. Consider the objective function defined in (5.15):

\[
\frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t} \left( Y_{i,t} - T(i,t,p) - Z_i'\beta \right)^2
\]

\[= \frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t} \left( Y_{i,t}^2 - 2Y_{i,t}T(i,t,p) + T(i,t,p)^2 - 2Y_{i,t}Z_i'\beta + (Z_i'\beta)^2 + 2T(i,t,p)Z_i'\beta \right).\]

Since we are optimizing over the tensor $T$ and $\beta$, we can drop the first term above, and consider the three following terms.
Terms only involving \( T \):

\[
\frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} -2w_{i,t}Y_{i,t}T(i,t,p) + \frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t}T(i,t,p)^2 \tag{5.19}
\]

Terms only involving \( \beta \):

\[
\frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} -2w_{i,t}Y_{i,t}Z'_{i}\beta + \frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t}(Z'_{i}\beta)^2 \tag{5.20}
\]

Terms involving both \( T \) and \( \beta \):

\[
\frac{1}{NT} \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} 2w_{i,t}T(i,t,p)Z'_{i}\beta \tag{5.21}
\]

We now proceed as we did in subsection 5.3.2. Recall that we defined the weight tensor \( W \) as

\[
W_{p_i,t} = \frac{w_{i,t}Y_{i,t}}{\Pr[(i,t) \in O_p]} \quad \text{if } (i,t) \in O_p
\]

\[
0 \quad \text{otherwise}
\]

With these two definitions, the first term (5.19) can be replaced by \( \frac{1}{NT} \|Y_{w} - T\|_W \). Consider the second term (5.20). This term simplifies to the following:

\[
\frac{1}{NT} \sum_{i,t} -2w_{i,t}Y_{i,t}X'_{i}\beta + \frac{1}{NT} \sum_{i,t} w_{i,t}(Z'_{i}\beta)^2
\]

Now we add the square terms involving \( Y_{i,t} \) to complete square.

\[
\frac{1}{NT} \sum_{i,t} w_{i,t} (Y_{i,t} - Z'_{i}\beta)^2
\]

Now consider the third term. Using Cauchy-Schwarz inequality we bound it by

\[
\frac{2}{NT} \left\{ \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t}T^2(i,t,p) \right\}^{1/2} \left\{ \sum_{p=1}^{2^k} \sum_{(i,t) \in O_p} w_{i,t}(Z'_{i}\beta)^2 \right\}^{1/2}
\]

The first term in the product above can be approximated by it’s population variant \( \|T\|_W \). The second term can be upper bounded by \( \|\beta\|_2 \left( \sum_{i,t} w_{i,t}Z_i \|Z_i\|^2 \right)^{1/2} \). This gives us the following
overall bound on the third term:

\[
\frac{2}{NT} \|T\|_W \|\beta\|_2 \left\{ \sum_{i,t} w_{i,t} \|Z_i\|_2^2 \right\}^{1/2}
\]

Combining the three new terms we get the following objective:

\[
\min_{\beta \in \mathbb{R}^d, T \in \mathbb{R}^{N \times T \times 2^k}, \text{rank}(T) \leq r, \|T\|_*, \leq L} \frac{1}{NT} \left[ \|Y_w - T\|_W^2 + \sum_{i,t} w_{i,t} (Y_{i,t} - Z_i^t \beta)^2 + 2\|T\|_W \|\beta\|_2 \left\{ \sum_{i,t} w_{i,t} \|Z_i\|_2^2 \right\}^{1/2} \right].
\]

(5.22)

Projected Gradient Descent

We now provide a simple algorithm for the optimization problem (5.22) based on projected gradient descent. Algorithm 5.3 repeatedly applies three steps. Line 6 computes a gradient step to compute the new tensor \( T_u \). However, the tensor \( T_u \) might not be of rank \( r \), so line 4 computes a projection of tensor \( T_u \) into the space of tensors of rank \( r \). As the projection step is a standard rank \( r \) approximation of a tensor, we use the parafac method from the TensorLy package [Kos+18] for this step. Given new value of \( T \), line 5 computes the best possible value of the parameter \( \beta \). Note that this step is a convex optimization problem and it’s solution can be computed in polynomial time. We leave empirical validation of algorithm 5.3 as a future work.

Algorithm 5.3: Weighted Low Rank Tensor Approximation with Covariates

**Input:** Tensor \( Y_w \in \mathbb{R}^{N \times T \times 2^k} \), weight tensor \( W \in \mathbb{R}^{N \times T \times 2^k} \), baseline covariates \( Z_i \in \mathbb{R}^d \) for each \( i \), weight matrix \( w \in \mathbb{R}^{N \times T} \), rank \( r \), and number of iterations \( \text{Iter} \).

**Output:** Tensor \( T \in \mathbb{R}^{N \times T \times 2^k} \) of rank \( r \) and \( \beta \in \mathbb{R}^d \).

1. Initialize \( T \) and \( \beta \).
2. for \( j = 1 \) to \( \text{Iter} \) do
   3. \( T_u \leftarrow T - \lambda \left[ -2(Y_w - T) \cdot W^2 + 2 \frac{T}{\|T\|_W} \|\beta\|_2 \left\{ \sum_{i,t} w_{i,t} \|Z_i\|_2^2 \right\}^{1/2} \right] \)
   4. \( T \leftarrow \text{Project}(T_u, r) \)
   5. \( \beta \leftarrow \arg \min_{\alpha \in \mathbb{R}^d} \left[ \sum_{i,t} w_{i,t} (Y_{i,t} - Z_i^t \alpha)^2 + 2\|T\|_W \|\alpha\|_2 \left\{ \sum_{i,t} w_{i,t} \|Z_i\|_2^2 \right\}^{1/2} \right] \)
   6. if Relative Change in Loss \( \leq \epsilon \) then
      7. return \( T \) and \( \beta \).
8. return \( T \) and \( \beta \).
5.5 Conclusion

In this work, we introduced a new form of marginal structural models. We used tensors to model the potential outcomes for time-varying treatments and showed how to efficiently estimate the parameters of the model. There are several directions for future work including handling of time-varying unobserved confounders, and developing a doubly robust estimator which works when either the outcome model or the treatment model is correctly specified.
Chapter 6

Conclusion

In this thesis, I have discussed three main challenges in the design of robust AI systems –

1. *Elicitation* to obtain high quality feedback on training instances

2. *Aggregation* to understand and summarize the trade-offs arising from decisions across society, and

3. *Causal Inference* to estimate the impact of any system before deploying.

Chapters 1 and 2 discussed the peer prediction method to elicit feedback on training instances where responses cannot be verified. I considered both the heterogeneous agents and heterogeneous tasks setting, and showed that the mechanisms developed have better incentive guarantees, both in theory and through empirical evaluation. Chapters 3 and 4 considered theoretical aspects of voting rules motivated by anticipated uses of AI systems in the context of societal decision making. In particular, Chapter 4 provides a unified view of voting by considering elicitation and aggregation together, and provides a sharp characterization of the performance of such rules. Finally, Chapter 5 contributes a new methodological approach, making use of tensor decomposition, for estimating the impact of a policy that applies treatments over a sequence of rounds. I show that the proposed estimator is consistent, propose an algorithm to efficiently solve the estimation problem, and through simulation show that it has better performance than existing methods for causal inference under time-varying
treatments. Moving forward, there are many open directions in regard to each of the three challenge areas, but here I will highlight only one important problem from each domain:

**Elicitation:** As we start using AI systems for various societal decision making settings, loan approval, and bail decisions, for example, then we need to think of such AI as being part of our democratic systems, where people’s inputs are used to regulate and otherwise determine the way in which these AI systems function. In fact, several recent papers [Rah; Lee+18] have advocated for a unifying system for algorithmic governance. However, there are several challenges in eliciting users’ preferences in these contexts, and this is necessary for the right kinds of representation. Often, the trade-off that results from a change in policy is not a priori clear. Imagine for example an algorithm that is used to decide whether a person should be given a loan or not, and we want to elicit people’s preferences about a change to the way the algorithm makes decisions. However, the change will substitute one set of users with another set of users in terms of giving loan. The trade-off from such a change is not immediately clear to people as this kind of decision has long-term implications. Moreover, we might also need to build different kinds of elicitation formats for different kinds of users. Experts, for example, will have a detailed opinion about how the system should work, whereas others may prefer to specify simple preferences over the outcomes. It seems that in order to make AI systems part of governance, we will need to develop methods of elicitation that support diverse forms of inputs and also help people learn the trade-offs of different approaches to making decisions.

**Aggregation:** We have discussed in Chapters 3 and 4 the aggregation of preferences over a set of alternatives. However, for many scenarios we will need to aggregate various models for decision-making. For example, imagine building a system that would decide whether or not to give loan to an individual. We cannot just use past data to build such a system as we don’t observe certain counterfactual outcomes – what would have happened if a person who was denied loan, were actually given loan. This requires us to build a causal model of the world that can describe the relationship between different input variables and how the input variables affect the outcome variables. One way to build causal models is to ask experts’ opinions, and then aggregate their diverse opinions [ACH18]. However, these algorithms for developing causal insights are limited in the sense that they can aggregate causal models only
over the observed variables. We may not have access to the right set of variables in many interesting settings, healthcare or recommender systems for example, and for this reason there can be unobserved confounders. Moreover, some experts might have insights about only a part of the system. Therefore, we would need to develop methods to aggregate partial causal models, or causal models with possibly different unobserved confounders across different experts. Even if the aggregation method comes up with a causal model that is incomplete, but correctly specifies the causal relationships among some variables, that would greatly reduce the potential number of interventions we would need to perform and could be very helpful in developing causal AI systems.

**Causal Inference:** Consider the following problem. Suppose we have satellite images of each of the 50 states of the US over the past several years and we want to determine the impact of a policy, reducing carbon emission for example. For this question, it is crucial that we use all of the available images for our analysis. Although a recurrent convolutional neural network (RCNN) [PC14] will be able to handle such a sequence of images, such systems cannot be used for valid causal inference. In Chapter 5, we saw how we can perform causal inference with time-varying treatments by carefully modeling the counterfactual outcomes, and avoiding the combinatorial explosion of the counterfactual outcomes with time. However, this approach uses covariates in a very limited manner. It would be ideal to use RCNN for our setting, but performing valid inference with such models seems challenging. I believe that we will need to make significant progress in terms of understanding the uncertainty of the predictions of neural networks and developing associated methods of high dimensional statistical inference before we can use the predictive powers of such networks to perform better causal inference in the time-varying setting.

To conclude, AI systems have made enormous progress in their predictive performance over the past decade. However, in order to build AI systems that are robust and can be reliably used in societal contexts, we need to make progress in three areas – elicitation, aggregation, and causal inference. But I do believe that with continued progress, AI systems will benefit society in the long run.
References

[Ace+14] Daron Acemoglu, Suresh Naidu, Pascual Restrepo, and James A Robinson. 


[Wu+16] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. “Google’s Neural Machine Translation System: Bridging the Gap Between


Appendix A

Appendix to Chapter 1

A.1 Missing Proofs

A.1.1 Proof of Lemma 1.1

The fact that $\sum_{r_1, r_2 \in [n]} \psi_{r_1} \hat{S}(r_1, r_2) \phi_{r_2} \geq 0$ follows easily from the fact that $\psi_{r_1} \geq 0$, $\phi_{r_2} \geq 0$ and $\hat{S}(r_1, r_2) \geq 0$ for all $r_1$ and $r_2$. The other direction follows from the following.

$$
\sum_{r_1, r_2 \in [n]} \psi_{r_1} \hat{S}(r_1, r_2) \phi_{r_2} = \sum_{r_1 \in [n]} \psi_{r_1} \sum_{r_2 \in [n]} \hat{S}(r_1, r_2) \phi_{r_2}
$$

$$
\leq \sum_{r_1 \in [n]} \psi_{r_1} \sum_{r_2 \in [n]} 1 \cdot \phi_{r_2} \quad (\hat{S}(r_1, r_2) \leq 1)
$$

$$
= \sum_{r_1 \in [n]} \psi_{r_1} \cdot 1 \quad (\sum_{r_2 \in [n]} \phi_{r_2} = 1)
$$

$$
= 1 \quad (\sum_{r_1 \in [n]} \psi_{r_1} = 1)
$$

A.1.2 Proof of Lemma 1.2

We have that

$$
\sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F_{ir_p}^{p} F_{jr_q}^{q} \hat{S}_{p,q}(r_p, r_q) = \sum_{(i,j): \Delta_{p,q}(i,j) > 0} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F_{ir_p}^{p} F_{jr_q}^{q} \hat{S}_{p,q}(r_p, r_q)
$$

$$
+ \sum_{(i,j): \Delta_{p,q}(i,j) \leq 0} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F_{ir_p}^{p} F_{jr_q}^{q} \hat{S}_{p,q}(r_p, r_q) \quad (A.1)
$$

160
Now we make two observations. Firstly,

\[ \sum_{i,j: \Delta_{pq}(i,j) > 0} \Delta_{pq}(i,j) = \sum_{i,j: \Delta_{pq}(i,j) > 0} \Delta_{pq}(i,j) \sum_{r_p, r_q} F_{ir_p}^P F_{jr_q}^q \hat{S}_{pq}(r_p, r_q), \]

which follows from Lemma 1.1 as \( \sum_{r_p, r_q} F_{ir_p}^P F_{jr_q}^q \hat{S}_{pq}(r_p, r_q) \leq 1 \). Secondly,

\[ \sum_{(i,j): \Delta_{pq}(i,j) \leq 0} \Delta_{pq}(i,j) \sum_{r_p, r_q} F_{ir_p}^P F_{jr_q}^q \hat{S}_{pq}(r_p, r_q) \leq 0, \]

which again follows from Lemma 1.1 as \( \sum_{r_p, r_q} F_{ir_p}^P F_{jr_q}^q \hat{S}_{pq}(r_p, r_q) \geq 0 \).

Now, the desired bound follows from Equation A.1 and the two observations above.

A.1.3 Proof of Lemma 1.3

Let \( 1[\cdot] \) denote the indicator function. Then the utility of the truthful strategy profile \( \{1, \{1\}_{q \neq p}\} \) is given by

\[
u^*_p(1, \{1\}_{q \neq p}) = \frac{1}{\ell - 1} \sum_{q \in P\setminus\{p\}} \sum_{i, j} \Delta_{pq}(i, j) \sum_{r_p, r_q} 1[i = r_p] \cdot 1[j = r_q] \cdot S_{p,q}^*(r_p, r_q)
\]

\[
= \frac{1}{\ell - 1} \sum_{q \in P\setminus\{p\}} \sum_{i, j} \Delta_{pq}(i, j) \cdot S_{p,q}^*(i, j)
\]

\[
= \frac{1}{\ell - 1} \sum_{q \in P\setminus\{p\}} \sum_{i, j: \Delta_{pq}(i,j) > 0} \Delta_{pq}(i, j)
\]

The utility of any other strategy profile \( \{F^p, \{F^q\}_{q \neq p}\} \) is given by

\[
u^*_p(F^p, \{F^q\}_{q \neq p}) = \frac{1}{\ell - 1} \sum_{q \in P\setminus\{p\}} \sum_{i, j} \Delta_{pq}(i, j) \sum_{r_p, r_q} F_{ir_p}^P F_{jr_q}^q S_{p,q}^*(r_p, r_q).
\]

From Lemma 1.2 we then have

\[
u^*_p(1, \{1\}_{q \neq p}) \geq \nu^*_p(F^p, \{F^q\}_{q \neq p}).
\]

For an uninformed strategy \( F^p \) such that all the rows of \( F^p \) are the same, i.e. \( F^p_i = \psi \) for all \( i \)
where $\psi$ is a probability distribution, we have

$$
\begin{align*}
\ell^*_{p}(F^p, \{F^q\}_{q \neq p}) &= \frac{1}{\ell - 1} \sum_{q \neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F^p_{f_{r_p}} F^q_{j_{r_q}} S^*_{p,q}(r_p, r_q) \\
&= \frac{1}{\ell - 1} \sum_{q \neq p} \sum_{i,j} \psi_{r_p} F^q_{j_{r_q}} S^*_{p,q}(r_p, r_q) \\
&= \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{f_{r_p}, j_{r_q}} \psi_{r_p} F^q_{j_{r_q}} S^*_{p,q}(r_p, r_q) \left( \sum_i \Delta_{p,q}(i,j) \right) = 0
\end{align*}
$$

The last equality follows since the row / column sum of delta matrices is zero. On the other hand, $\ell^*_{p}(\mathbb{I}, \{\mathbb{I}\}_{q \neq p})$, being a sum of only positive entries, is strictly greater than 0.

**A.1.4 Proof of Lemma 1.6**

Let $\Delta_{p,G(q)} = \frac{1}{|G(q)|} \sum_{r \in G(q)} \Delta_{p,r}$, then using the property of clusters we have

$$
\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 = \left\| \frac{1}{|G(p)|} \sum_{r \in G(p), \sigma \in G(q)} \Delta_{u,v} - \frac{1}{|G(p)||G(q)|} \sum_{u \in G(p), v \in G(q)} \Delta_{u,v} \right\|_1
\leq \frac{1}{|G(p)||G(q)|} \sum_{u \in G(p), v \in G(q)} \|\Delta_{p,q} - \Delta_{u,v}\|_1
\leq \frac{1}{|G(p)||G(q)|} \sum_{u \in G(p), v \in G(q)} \|\Delta_{p,q} - \Delta_{a,q}\|_1 + \|\Delta_{a,q} - \Delta_{u,v}\|_1
\leq \frac{1}{|G(p)||G(q)|} \sum_{u \in G(p), v \in G(q)} 2 \max_{a,b,c \in P : G(a) = G(b)} \|\Delta_{a,c} - \Delta_{b,c}\|_1
= 2 \max_{a,b,c \in P : G(a) = G(b)} \|\Delta_{a,c} - \Delta_{b,c}\|_1,
$$

as required.

**A.1.5 Proof of Lemma 1.7**

Lemma 1.4 shows that $\|\bar{D}_{p,q} - D_{p,q}\|_1 \leq \epsilon' \Rightarrow \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 \leq 3\epsilon'$.

Now,

$$
\|\bar{\Delta}_{p,q} - \Delta_{G_0,G_i}\|_1 \leq \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 + \|\Delta_{p,q} - \Delta_{G_0,G_i}\|_1 \leq 3\epsilon' + 2\varepsilon.
$$

162
The last inequality uses Lemma 1.6.

A.1.6 Proof of Lemma 1.9

\[ \Delta_{G_a,G_b}(i,j) = \frac{1}{|G_a||G_b|} \sum_{p \in G_a,q \in G_b} \Delta_{p,q}(i,j) = \frac{1}{|G_a||G_b|} \sum_{p \in G_a,q \in G_b} D_{p,q}(i,j) - D_p(i)D_q(j) \]

\[ = \frac{1}{|G_a||G_b|} \sum_{p \in G_a,q \in G_b} \sum_k \tau_k C_k^p C_k^q - \sum_k \tau_k C_k^p \sum_{k} C_k^q \]

\[ = \sum_k \tau_k \left( \frac{1}{|G_a|} \sum_{p \in G_a} C_k^p \right) \left( \frac{1}{|G_b|} \sum_{q \in G_b} C_k^q \right) \]

\[ - \sum_k \tau_k \left( \frac{1}{|G_a|} \sum_{p \in G_a} C_k^p \sum_k \tau_k \left( \frac{1}{|G_b|} \sum_{q \in G_b} C_k^q \right) \right) \]

\[ = \sum_k \tau_k C_k^a C_k^b – \sum_k \tau_k C_k^a \sum_k \tau_k C_k^b \]

Now

\[ \| \Delta_{G_a,G_b} - \Delta_{G_a,G_b} \|_1 = \sum_{i,j} | \Delta_{G_a,G_b}(i,j) - \Delta_{G_a,G_b}(i,j) | \]

\[ = \sum_{i,j} \left| \sum_k \tau_k \tilde{C}_k^a \tilde{C}_k^b \right| - \sum_{i,j} \left| \sum_k \tau_k \tilde{C}_k^a \tilde{C}_k^b \right| + \sum_{i,j} \left| \sum_k \tau_k \tilde{C}_k^a \tilde{C}_k^b \right| - \sum_{i,j} \left| \sum_k \tau_k \tilde{C}_k^a \tilde{C}_k^b \right| \]

\[ = \sum_k \tau_k \sum_i \tilde{C}_k^b = C_k^b + \sum_k \tau_k \sum_i \tilde{C}_k^a = C_k^a \]

\[ + \sum_{i,j} \left| \sum_k \tau_k \tilde{C}_k^a \sum_k \tau_k \tilde{C}_k^b \right| - \sum_k \tau_k \tilde{C}_k^a \sum_k \tau_k \tilde{C}_k^b - \sum_k \tau_k \tilde{C}_k^a \sum_k \tau_k \tilde{C}_k^b \]

\[ = \sum_k \tau_k \sum_j \tilde{C}_k^b = C_k^b + \sum_k \tau_k \sum_i \tilde{C}_k^a = C_k^a \]

\[ + \sum_k \tau_k \sum_i \tilde{C}_k^a \sum_k \tau_k \tilde{C}_k^b \]

\[ \leq 2\| C^a - C^b \|_1 + 2\| C^a - C^b \|_1 \leq 4 \varepsilon/4 = \varepsilon' \]
A.1.7 Proof of Lemma 1.11

1. First moments of $\{\tilde{R}_g\}_{g=1}^K$ and $\{R_g\}_{g=1}^K$ are equal:

$$E[\tilde{R}_{aj}] = \frac{1}{|G_a|} \sum_{p \in G_a} E[r_{pj}] = E[R_{aj}]$$

2. Second order cross-moments of $\{\tilde{R}_g\}_{g=1}^K$ and $\{R_g\}_{g=1}^K$ are equal:

$$E[\tilde{R}_{aj} \otimes \tilde{R}_{bj}] = \sum_k \tau_k E[\tilde{R}_{aj} \otimes \tilde{R}_{bj} | y_j = k] = \sum_k \tau_k E[\tilde{R}_{aj} | y_j = k] \otimes E[\tilde{R}_{bj} | y_j = k]$$

$$= \sum_k \tau_k \left( \frac{1}{|G_a|} \sum_{p \in G_a} C^p_k \right) \otimes \left( \frac{1}{|G_b|} \sum_{q \in G_b} C^q_k \right) = \sum_k \tau_k C^a_k \otimes C^b_k = E[R_{aj} \otimes R_{bj}]$$

3. Third order cross-moments of $\{\tilde{R}_g\}_{g=1}^K$ and $\{R_g\}_{g=1}^K$ are equal:

$$E[\tilde{R}_{aj} \otimes \tilde{R}_{bj} \otimes \tilde{R}_{cj}] = \sum_k \tau_k E[\tilde{R}_{aj} \otimes \tilde{R}_{bj} \otimes \tilde{R}_{cj} | y_j = k]$$

$$= \sum_k \tau_k E[\tilde{R}_{aj} | y_j = k] \otimes E[\tilde{R}_{bj} | y_j = k] \otimes E[\tilde{R}_{cj} | y_j = k]$$

$$= \sum_k \tau_k \left( \frac{1}{|G_a|} \sum_{p \in G_a} C^p_k \right) \otimes \left( \frac{1}{|G_b|} \sum_{q \in G_b} C^q_k \right) \otimes \left( \frac{1}{|G_c|} \sum_{r \in G_c} C^r_k \right)$$

$$= \sum_k \tau_k C^a_k \otimes C^b_k \otimes C^c_k = E[R_{aj} \otimes R_{bj} \otimes R_{cj}]$$
A.1.8 Proof of Lemma 1.14

We have

\[
\|\Delta_{p,q} - \Delta_{p,r}\|_1 = \sum_{i,j} |\Delta_{p,q}(i,j) - \Delta_{p,r}(i,j)|
\]

\[
= \sum_{i,j} |D_{p,q}(i,j) - D_p(i)D_q(j) - D_{p,r}(i,j) + D_p(i)D_r(j)|
\]

\[
= \sum_{i,j} |D_{p,q}(i,j) - D_{p,r}(i,j) - D_p(i)(D_q(j) - D_r(j))|
\]

\[
= \sum_{i,j} \left| \sum_k \pi_k C_{kij}^p C_{kij}^q - \sum_k \pi_k C_{kij}^p C_{kij}^r - \sum_k \pi_k C_{kij}^p \left(\sum_l \pi_l C_{lij}^q - \sum_l \pi_l C_{lij}^r\right)\right|
\]

\[
= \sum_{i,j} \left| \sum_k \pi_k C_{kij}^p \left(C_{kij}^q - C_{kij}^r\right) - \sum_k \pi_k C_{kij}^p \left(\sum_l \pi_l \left(C_{lij}^q - C_{lij}^r\right)\right)\right|
\]

\[
\leq \sum_j \sum_k \pi_k \left|C_{kij}^q - C_{kij}^r\right| \sum_i C_{kij}^p + \sum_j \sum_k \pi_k \sum_i \pi_l \left|C_{lij}^q - C_{lij}^r\right| \sum_i C_{kij}^p
\]

\[
= \sum_j \sum_k \pi_k \left|C_{kij}^q - C_{kij}^r\right| + \sum_j \sum_k \pi_k \sum_i \pi_l \left|C_{lij}^q - C_{lij}^r\right| \quad [\text{Using } \sum_i C_{kij}^p = 1]
\]

\[
\leq \sum_k \pi_k \sum_j \left|C_{kij}^q - C_{kij}^r\right| + \sum_k \pi_k \sum_l \pi_l \sum_j \left|C_{lij}^q - C_{lij}^r\right|
\]

\[
= \sum_k \pi_k \sum_j \left|C_{kij}^q - C_{kij}^r\right| + \sum_k \pi_l \sum_j \left|C_{lij}^q - C_{lij}^r\right| \quad [\text{Using } \sum_k \pi_k = 1]
\]

\[
= 2 \cdot \sum_k \pi_k \sum_j \left|C_{kij}^q - C_{kij}^r\right|
\]

as required.
Appendix B

Appendix to Chapter 2

B.1 Missing Proofs

B.1.1 Proof of Lemma 2.1

\[
\sum_i \sum_j \Delta_b(i, j) = \sum_i \sum_j \left\{ P_b(i, j) - \frac{1}{(m-1)(m-2)} \sum_{l' \in [m], l' \neq [m]} \sum_{l'' \in [m], \{b'\}} P_{l'}(i) P_{l''}(j) \right\}
\]

\[
= \sum_i \left\{ P_b(i) - \frac{1}{(m-1)(m-2)} \sum_{l' \in [m], l' \neq [m]} \sum_{l'' \in [m], \{b'\}} P_{l'}(i) \right\}
\]

\[
= 1 - 1 = 0
\]

B.1.2 Proof of Theorem 2.3

We will write \( E[T, F, G] \) to denote the average expected score under strategies \( F \) and \( G \) when using the score matrix \( T = \{T_b\}_{b=1}^m \). Suppose \( S = \{S_b\}_{b=1}^m \) is the true scoring matrix and \( \hat{S} = \{\hat{S}_b\}_{b=1}^m \) is the scoring matrix estimated from the data. Then

\[
E[\hat{S}, F, G] = \frac{1}{m} \sum_{b=1}^m \sum_{i, j} \Delta_b(i, j) \hat{S}_b(F_i, G_j) \leq \frac{1}{m} \sum_{b=1}^m \sum_{i, j : \Delta_b(i, j) > 0} \Delta_b(i, j) = E[S, \mathbb{I}, \mathbb{I}]
\]

Therefore, in order to show \( E[\hat{S}, \mathbb{I}, \mathbb{I}] \geq E[\hat{S}, F, G] - \epsilon \) it is enough to show that \( E[\hat{S}, \mathbb{I}, \mathbb{I}] \geq \)
$$E[S, I, I] - \epsilon.$$ 

$$\left| \frac{1}{m} E[\tilde{S}, I, I] - \frac{1}{m} E[S, I, I] \right|$$ 

$$= \left| \frac{1}{m} \sum_{b=1}^{m} \sum_{i,j} \Delta_b(i, j) \left( \hat{S}_b(i, j) - S_b(i, j) \right) \right|$$ 

$$= \left| \frac{1}{m} \sum_{b=1}^{m} \sum_{i,j} \Delta_b(i, j) \left( \text{sign} \left( \hat{\Delta}_b(i, j) \right) - \text{sign} \left( \Delta_b(i, j) \right) \right) \right|$$ 

$$\leq \left| \frac{1}{m} \sum_{b=1}^{m} \sum_{i,j} \left( \hat{\Delta}_b(i, j) - \Delta_b(i, j) \right) \right|$$ 

$$= \left| \frac{1}{m} \sum_{b=1}^{m} \sum_{i,j} P_b(i, j) - T_b(i, j) - \frac{1}{(m-1)(m-2)} \sum_{b=1}^{m} \sum_{i,j, b' \neq b} \left( P_{b'}(i) P_{v'}(j) - T_{b'}(i) T_{v'}(j) \right) \right|$$ 

$$\leq \frac{1}{m} \sum_{b=1}^{m} \sum_{i,j} |P_b(i, j) - T_b(i, j)| + \frac{1}{m(m-1)(m-2)} \sum_{b=1}^{m} \sum_{i,j, b' \neq b} |P_{b'}(i) P_{v'}(j) - T_{b'}(i) T_{v'}(j)|$$ 

$$= \frac{1}{m} \sum_{b=1}^{m} \sum_{i,j} |P_b(i, j) - T_b(i, j)| + \frac{1}{m(m-1)(m-2)} \sum_{b=1}^{m} \sum_{i,j} \sum_{b' \neq b} \left( P_{b'}(i) (P_{v'}(j) - T_{v'}(j)) \right)$$ 

$$+ |T_{v'}(j) (P_{b'}(i) - T_b(i))|$$ 

$$\leq \frac{1}{m} \sum_{b=1}^{m} \sum_{i,j} |P_b(i, j) - T_b(i, j)| + \frac{1}{m(m-1)(m-2)} \sum_{b=1}^{m} \sum_{i,j} \sum_{b' \neq b} \left\{ \sum_{j} |P_{v'}(j) - T_{v'}(j)| \sum_{i} P_{b'}(i) \right. $$ 

$$+ \left. \sum_{i} |P_{b'}(i) - T_{b'}(i)| \sum_{j} T_{v'}(j) \right\}$$ 

$$= \frac{1}{m} \sum_{b=1}^{m} \sum_{i,j} |P_b(i, j) - T_b(i, j)| + \frac{1}{m(m-1)(m-2)} \sum_{b=1}^{m} \sum_{i,j} \sum_{b' \neq b} \left\{ \sum_{j} |P_{v'}(j) - T_{v'}(j)| \right.$$

$$+ \left. \sum_{i} |P_{b'}(i) - T_{b'}(i)| \right\}$$ 

(B.1) 

Now if we have $O \left( \frac{n^2}{\epsilon^2} \log \left( \frac{m}{\epsilon} \right) \right)$ samples from each joint distribution $P_b$ (where $n$ is the
number of signals) and \( O \left( \frac{n}{\epsilon^2} \log \left( \frac{m}{\delta} \right) \right) \) from each marginal distribution \( P_b \), we can ensure that with probability at least \( 1 - \delta \), for all \( b = 1, 2, \ldots, m \) the following results hold (see [DL12] for a proof)

\[
\sum_{i,j} |P_b(i,j) - T_b(i,j)| \leq \frac{\epsilon}{3} \quad \text{and} \quad \sum_{i} |P_b(i) - T_b(i)| \leq \frac{\epsilon}{3}.
\] (B.2)

Note: If we just had \( O(n/\epsilon^2 \log(1/\delta)) \) samples for each task, then we can guarantee (B.2) for each task separately with probability at least \( 1 - \delta \). By the union bound, this would give a success probability of \( 1 - m\delta \) over all tasks. So in order to have a \( 1 - \delta \) confidence bound, we need a \( \log(m/\delta) \) factor in the sample complexity. Substituting the bounds from eq. (B.2) in eq. (B.1) and simplifying gives us \( |E \left[ \hat{S}, I, I \right] - E [S, I, I] | \leq \epsilon \). Since there are \( q \) agents providing reviews for each task, we get \( q^2 \) samples from each joint distribution and \( q \) samples from each marginal distribution. So as long as \( q = \Omega \left( \frac{n}{\epsilon^2} \log \left( \frac{m}{\delta} \right) \right) \) we have enough number of samples and we are done.
B.2 Additional Plots

Figure B.1: Histograms for the 204 (region, business type) pairs of expected benefit (averaged across questions) from truthful behavior vs. some other strategy, when fraction \( p \) is truthful and fraction \( 1 - p \) adopt the same, non-truthful strategy for \( p = 0.1, 0.5, 0.7, 0.9 \).
Appendix C

Appendix to Chapter 4

C.1 Lower Bound for Plurality Votes

In this section, we show that eliciting plurality votes (whereby each voter picks her most favorite alternative) results in $\Omega(m)$ distortion, even with randomized aggregation. This is implied by Theorem 4.4, which proves this for any elicitation that has at most $\log m$ communication complexity. However, for the special case of plurality votes, we can provide a much simpler proof.

**Theorem C.1.** Every voting rule which elicits plurality votes incurs $\Omega(m)$ distortion.

**Proof.** For simplicity, let the number of voters $n$ be divisible by the number of alternatives $m$. Consider an input profile in which the set of voters $N$ is partitioned into equal-size sets $\{N_a\}_{a \in A}$ such that for each $a \in A$, $a$ is the most favorite alternative of every voter in $N_a$.

Take any voting rule $f$. It must return some alternative $a^* \in A$ with probability at most $1/m$. Now, construct adversarial valuations of voters $\tilde{v}$ as follows.

- For all $i \in N_{a^*}$, $v_i(a^*) = 1$ and $v_i(a) = 0$ for all $a \in A\{a^*\}$.
- For all $\hat{a} \in A\{a^*\}$ and $i \in N_{\hat{a}}$, $v_i(\hat{a}) = v_i(a^*) = 1/2$ and $v_i(a) = 0$ for all $a \in A\{a^*,\hat{a}\}$.

Under these valuations, we have $sw(a^*, \tilde{v}) \geq n/2$, while $sw(a, \tilde{v}) = (n/m) \cdot (1/2)$ for every
a \in A\setminus\{a^*\}. Hence, the distortion of f is

$$\text{dist}(f) \geq \frac{\text{sw}(a^*, \bar{\vartheta})}{m \text{sw}(a^*, \bar{\vartheta}) + \frac{m-1}{m} \frac{n}{2m}} = \Omega(m),$$

where the final transition holds when substituting \(\text{sw}(a^*, \bar{\vartheta}) \geq n/2\).

\[ \square \]

## C.2 Lower Bound on the Communication Complexity of FDISJ\(_{m,s,t}\)

In this section, we prove a lower bound on the communication complexity of multi-party fixed-size set-disjointness. Let us recall Theorem 4.5.

**Theorem 4.5.** For a sufficiently small constant \(\delta > 0\) and \(m \geq (3/2)st\), \(R_\delta(\text{FDISJ}_{m,s,t}) = \Omega(s)\).

**Proof.** Suppose there is a \(\delta\)-error protocol \(\Pi\) for FDISJ\(_{m,s,t}\). We use it to construct a \(2\delta\)-error protocol \(\Pi'\) for DISJ\(_{m',t'}\), where \(m' = st/2\) and \(t' = 2t\).

Consider an instance \((S'_1, \ldots, S'_{t'})\) of DISJ\(_{m',t'}\). Due to the promise that the sets are either pairwise disjoint or pairwise uniquely intersecting, we have that at most one of the \(m'\) elements can appear in multiple sets. Hence, \(\sum_{i=1}^{t'} |S'_i| \leq m' - 1 + t'\). Due to the pigeonhole principle, there must exist at least \(t'/2 = t\) sets of size at most \(2(m' + t' - 1)/t'\). Note that

$$\frac{2(m' + t' - 1)}{t'} = \frac{st/2 + 2t - 1}{t} = \frac{s}{2} + 2 - \frac{1}{t} \leq s.$$

The final transition holds when \(s \geq 4\). When \(s < 4\), the lower bound of \(\Omega(s)\) is trivial.

Consider a set of \(t\) players \(\{i_1, \ldots, i_t\}\) such that \(|S'_{i_k}| \leq s\) for each \(k \in [t]\). Suppose that each such player \(i_k\) adds \(s - |S'_{i_k}|\) unique elements to \(S'_{i_k}\) and creates a set \(S_{i_k}\) with \(|S_{i_k}| = s\). The number of unique elements required is at most \(st\). Hence, the total number of elements used in sets \(S_{i_1}, \ldots, S_{i_t}\) is at most \(m' + st = (3/2)st \leq m\). In other words, these sets can be created using the \(m\)-element universe of FDISJ\(_{m,s,t}\). Further, it is easy to check that sets \(S_{i_1}, \ldots, S_{i_t}\) are pairwise disjoint (resp. pairwise uniquely intersecting) if and only if sets \(S'_1, \ldots, S'_{t'}\) are pairwise disjoint (resp. pairwise uniquely intersecting). Thus, \((S_{i_1}, \ldots, S_{i_t})\) is a valid instance of FDISJ\(_{m,s,t}\) and has the same solution as the instance \((S'_1, \ldots, S'_{t'})\) of DISJ\(_{m',t'}\).

Our goal is to construct a \(2\delta\)-error protocol \(\Pi'\) for DISJ\(_{m',t'}\) that solves \((S'_1, \ldots, S'_{t'})\) by effectively running the given \(\delta\)-error protocol \(\Pi\) for FDISJ\(_{m,s,t}\) on \((S'_1, \ldots, S'_{t'})\). We could ask
each player $i$ to report a single bit indicating whether $|S'_i| \leq s$, determine $t$ players for which this holds, and then run $\Pi$ on them. However, this would add a $t'$-bit overhead. Instead, we would like to bound the overhead in terms of the communication cost of $\Pi$, denoted $|\Pi|$, which could be significantly smaller.

This is achieved as follows. We first order the players according to a uniformly random permutation $\sigma$. Then, we simulate $\Pi$. Every time $\Pi$ wants to interact with a new player, we ask players that we have not interacted with so far, in the order in which they appear in $\sigma$, whether their sets have size at most $s$, until we find one such player. Then, we let $\Pi$ interact with this player. Protocol $\Pi'$ terminates naturally when protocol $\Pi$ terminates (and returns the same answer), but terminates abruptly if, at any point, it has interacted with more than $2|\Pi|/\delta$ players (and returns an arbitrary answer).

Note that $|\Pi|$ is also an upper bound with the number of players that $\Pi$ needs to interact with. Let $X$ be the smallest index such that there are at least $|\Pi|$ players having sets of size at most $s$ among the first $X$ players in $\sigma$. Then, because at least half of the players have sets of size at most $s$, we have $\mathbb{E}[X] \leq 2 \cdot |\Pi|$. Due to Markov’s inequality, we have that $\Pr[X > 2|\Pi|/\delta] \leq \delta$. Hence, the probability that $\Pi'$ terminates abruptly is at most $\delta$. When it does not terminate abruptly, it returns the wrong answer with probability at most $\delta$ (as $\Pi$ is a $\delta$-error protocol). Hence, due to the union bound, we conclude that $\Pi'$ is a $2\delta$-error protocol for $\text{DISJ}_{m',t'}$.

Finally, we have that $|\Pi'| \leq 2|\Pi|/\delta + |\Pi| = |\Pi|(1 + 2/\delta)$. When $\delta$ is sufficiently small, Gronemeier [Gro09] showed that $|\Pi'| \geq R_{2\delta}(\text{DISJ}_{m',t'}) = \Omega(m'/t') = \Omega(s)$. Hence, we have that $|\Pi| = \Omega(s)$. Since this holds for every $\delta$-error protocol $\Pi$ for $\text{FDISJ}_{m,s,t}$, we have $R_{\delta}(\text{FDISJ}_{m,s,t}) = \Omega(s)$, as desired. $\square$
Appendix D

Appendix to Chapter 5

D.1 Missing Proofs

D.1.1 Proof of Theorem 5.1

The weighted log-likelihood function with respect to a tensor $T_{N,T}$ is given as:

$$L_{N,T}(T_{N,T}) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} w_{it} \log \Pr[Y_{i,t}|T_{N,T}]$$

First we compute the expected value of the weighted log-likelihood with respect to the policy $\mathcal{P}$ (i.e. the random variables $\{Y_{i,t}\}_{i=1}^{N}$, $\{A_{i,t}\}_{i=1}^{N}$, $\{X_{i,t}\}_{i=1}^{N}$ and the true underlying tensor $T_{N,T}^*$. We write $\ell_{N,T}^{\ast}(T_{N,T})$ to denote this quantity as it only depends on the tensor $T_{N,T}$.

$$\ell_{N,T}^{\ast}(T_{N,T}) = E_{\mathcal{P},T_{N,T}^{\ast}} [L_{N,T}(T_{N,T})]$$

$$= \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} E_{A_{i,t},Y_{i,t},X_{i,t}} [w_{it} \log \Pr[Y_{i,t}|T_{N,T}]]$$

$$= \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} 2^k \sum_{a_{i,t-k+1:t}} \Pr[a_{i,t-k+1:t}] \times$$

$$\int \log \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T_{N,T}] \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T_{N,T}^*] dY_{i,t} \quad (D.1)$$

The last line follows from lemma D.1. We want to show that $\|\hat{T}_{N,T} - T_{N,T}^{\ast}\|$ becomes small as either $N$ or $T$ increases. Our proof is based on the proof of the consistency of the maximum likelihood given in [NM94]. We write $\Theta_{N,T}$ to denote the parameter space $\{T \in \mathbb{R}^{T \times N \times B} :$
rank($T$) ≤ $r$, $|T|_{\infty} \leq L$. $\Theta_{N,T}$ is bounded but need not be closed because of issues with border tensor. It is known that there might exist a sequence of rank $r$ tensors whose limit is a rank $r + 1$ tensor [Bin86]. However, we can exploit the concavity of the log-likelihood function to overcome this problem.

First consider a neighborhood $B$ of radius $d$ centered at $T_{N,T}^*$ and contained within the interior of $\Theta_{N,T}$.

$$B = \{T \in \mathbb{R}^{N \times T \times B} : ||T - T_{N,T}^*||_{2/\sqrt{NT}} \leq d\}$$

Lemma D.9 proves that $L_{N,T}(\cdot)$ is concave over $\Theta_{N,T}$. Since a concave function is continuous over the interior of its domain, $L_{N,T}(\cdot)$ is continuous over $B$. Moreover, unlike $\Theta_{N,T}$, set $B$ is a compact set. This implies that there exists a maximizer for $L_{N,T}(\cdot)$ over $B$. Suppose $\hat{T}_{N,T}$ be the maximizer of $L_{N,T}(\cdot)$ over $B$. Consider any $T \in \Theta_{N,T} \setminus B$. Then there exists $\lambda < 1$ such that $T' = \lambda \hat{T}_{N,T} + (1 - \lambda)T$ and $T' \in B$. This gives us the following:

$$L_{N,T}(\hat{T}_{N,T}) \geq L_{N,T}(T') = L_{N,T}(\lambda \hat{T}_{N,T} + (1 - \lambda)T) \geq \lambda L_{N,T}(\hat{T}_{N,T}) + (1 - \lambda)L_{N,T}(T)$$

$$\Rightarrow L_{N,T}(\hat{T}_{N,T}) \geq L_{N,T}(T)$$

This first line uses the concavity of $L_{N,T}(\cdot)$ (lemma D.9). This proves that $\hat{T}_{N,T}$ is actually the maximizer of $L_{N,T}(\cdot)$ over the entire parameter space $\Theta_{N,T}$. Moreover, any other maximizer $\tilde{T}_{N,T}$ of $L_{N,T}(\cdot)$ must be inside $B$. Otherwise, suppose $\hat{T}_{N,T}$ maximizes $L_{N,T}(\cdot)$ and $\tilde{T}_{N,T} \in \Theta_{N,T} \setminus B$. Then for $\varepsilon = (2\delta)^k d^2$ we have with probability at least $1 - O(1/\varepsilon^2 N^p)$,

$$L_{N,T}(\hat{T}_{N,T}) + \varepsilon/3 > \ell^*_N(\hat{T}_{N,T}) > \ell^*_N(\tilde{T}_{N,T}) + \varepsilon > L_{N,T}(\tilde{T}_{N,T}) + 2\varepsilon/3$$

The first and the third inequality uses lemma D.4 and the second inequality uses lemma D.16. Therefore, with probability at least $1 - O(1/(4^k d^2 \delta^{2k} N^p))$ all the maximizers of $L_{N,T}(\cdot)$ must be inside the ball $B$. This proves that for any $d$ we can choose $N$ large enough such that with high probability $1 - 1/poly(N)$ the maximizer of $L_{N,T}(\cdot)$ lies within a $d$ neighborhood of $T_{N,T}^*$. This proves the consistency of the estimate when $N$ increases to infinity. The proof of consistency when the number of time periods $T$ increases to infinity is similar.
D.1.2 Proof of Lemma 5.1

Lemma D.12 proves that
\[
\Pr \left[ \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} |T(i,t,p)|^2 \notin \left[ \|T\|^2_W - \varepsilon, \|T\|^2_W + \varepsilon \right] \right] \leq O \left( \exp \left( - \frac{2N^2}{T^4 \left( \frac{1-\delta}{T} \right)^2} \right) \right). \tag{5.7}
\]
Suppose \( \hat{T}_{N,T} \) solves 5.7 and \( \hat{\hat{T}}_{N,T} \) solves 5.13, then we get the following bound with probability at least \( 1 - \exp \left( - \frac{2N^2}{T^4 \left( \frac{1-\delta}{T} \right)^2} \right) : \)
\[
\frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} \left( Y_{i,t} - \hat{T}_{N,T}(i,t,p) \right)^2
\]
\[
= \frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} Y_{i,t}^2 \left( 2 \frac{1}{NT} \right) \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} Y_{i,t} \hat{T}_{N,T}(i,t,p) + \frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} \left( \hat{\hat{T}}_{N,T}(i,t,p) \right)^2
\]
\[
\leq \frac{1}{NT} \left( \|Y_W\|_W^2 + \varepsilon \right) - 2 \sum_{i,t,p} W(i,t,p) Y_{w}(i,t,p) \hat{T}_{N,T}(i,t,p) + \frac{1}{NT} \left( \|\hat{\hat{T}}_{N,T}\|_W^2 + \varepsilon \right)
\]
\[
= \frac{1}{NT} \|Y_W - \hat{T}_{N,T}\|_W^2 + \frac{2\varepsilon}{NT}
\]
\[
\leq \frac{1}{NT} \|Y_W - \hat{T}_{N,T}\|_W^2 + \frac{2\varepsilon}{NT}
\]
\[
= \frac{1}{NT} \left[ \|Y_W\|_W^2 - 2 \sum_{i,t,p} W(i,t,p) Y_{w}(i,t,p) \hat{T}_{N,T}(i,t,p) + \|\hat{\hat{T}}_{N,T}\|_W^2 \right] + \frac{2\varepsilon}{NT}
\]
\[
\leq \frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w(i,t) \left( Y_{i,t} - \hat{T}_{N,T}(i,t,p) \right)^2 + \frac{4\varepsilon}{NT}
\]

The first and the third inequality use lemma D.12 and the second inequality uses the fact that \( \hat{T}_{N,T} \) is the optimal solution to 5.13. Now if we substitute \( k = 1/8 \log_{(1-\delta)/\delta} N \) and \( \varepsilon = O \left( \frac{t^2}{N^{7/4}} \right) \), we get the first result.

Now suppose \( N \) is fixed. If assumption Asn.2 holds, then using lemma D.13 and a similar argument, we get with probability at least \( 1 - O \left( \frac{L^4 \left( \frac{1-\delta}{T} \right)^2}{T^2 \log T} \right) \),
\[
\frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} \left( Y_{i,t} - \hat{T}_{N,T}(i,t,p) \right)^2 \leq \text{OPT} + \frac{4\varepsilon}{NT}
\]
If we substitute \( k = 1/8 \log_{(1-\delta)/\delta} T \) and \( \varepsilon = O \left( \frac{t^2}{T^{1/8}} \right) \) we get the second result.
D.1.3 Proof of Theorem 5.2

We proceed same as theorem 5.1. The weighted log-likelihood function with respect to the tensor $T_{N,T}$ and parameter $\beta_{N,T}$ is given as:

$$L_{N,T}(T_{N,T}) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} w_{i,t} \log \Pr [Y_{i,t}|T_{N,T}, \beta_{N,T}]$$

The expected value of the weighted log-likelihood with respect to the policy $\mathcal{P}$, the true underlying tensor $T^*_N$, and the parameter $\beta^*$ is

$$E_{\mathcal{P},T^*_N,\beta^*} [L_{N,T}(T_{N,T}, \beta_{N,T})]$$

$$= \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} E_{\mathcal{A}_{i,1:T},Y_{i,1:T},X_{i,1:T},Z_{i}} [w_{i,t} \log \Pr [Y_{i,t}|T_{N,T}, \beta_{N,T}]]$$

$$= \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{z_i} \Pr [z_i] \Pr [a_{i,t-k+1:t}] \times$$

$$\int \log \Pr [Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t},z_i)|T, \beta] \Pr [Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t},z_i)|T^*_N, \beta^*] dY_{i,t}$$  \hspace{1cm} (D.2)

The last line follows from lemma D.2. As before, we write $\Theta_{N,T}$ to denote the parameter space \{ $T \in \mathbb{R}^{T \times N \times 2^k}, \beta \in \mathbb{R}^d$ : rank($T$) $\leq r$, $\|T\|_{\infty} \leq L$ \}. Consider a neighborhood $B$ centered around $T^*_N, \beta^*$ and contained within the interior of $\Theta_{N,T}$.

$$B = \{ T \in \mathbb{R}^{N \times T \times \mathbb{B}}, \beta \in \mathbb{R}^d : \|T - T^*_N\|_2 / \sqrt{NT} \leq d_1, \|\beta - \beta^*\|_2 \leq d_2 \}$$

Lemma D.10 proves that $L_{N,T}(\cdot)$ is concave over $\Theta_{N,T}$. Since a concave function is continuous over the interior of its domain, $L_{N,T}(\cdot)$ is continuous over $B$. Moreover, since the set $B$ is a compact set, that there exists a maximizer for $L_{N,T}(\cdot)$ over $B$. Let $(\tilde{T}, \tilde{\beta})$ be the maximizer. Consider any $(T, \beta) \in \Theta_{N,T} \setminus B$. We consider three cases.

Case 1: $\|T^*_N - T\| / \sqrt{NT} \leq d_1$ and $\|\beta^* - \beta\|_2 > d_2$. In this case, there exists $\lambda < 1$ such that $\beta' = \lambda \tilde{\beta} + (1 - \lambda) \beta$ and $(T, \beta') \in B$. Then,

$$L_{N,T}(\tilde{T}, \tilde{\beta}) \geq L_{N,T}(T, \tilde{\beta}) \geq L_{N,T}(T, \beta') = L_{N,T}(T, \lambda \tilde{\beta} + (1 - \lambda) \beta)$$

$$\geq \lambda L_{N,T}(T, \tilde{\beta}) + (1 - \lambda)L_{N,T}(T, \beta)$$

176
The first two inequalities follow since \((\hat{T}, \hat{\beta})\) is the maximizer over \(B\). The above set of inequalities gives us \(L_{N,T}(\hat{T}, \hat{\beta}) \geq L_{N,T}(T, \hat{\beta}) \geq L_{N,T}(T, \beta)\).

**Case 2**: \(\|T^*_N, T\|/\sqrt{N_T} > d_1\) and \(\|\beta^* - \beta\|_2 \leq d_2\). The argument for this case is similar to the first case.

**Case 3**: \(\|T^*_N, T\|/\sqrt{N_T} > d_1\) and \(\|\beta^* - \beta\|_2 > d_2\). The argument in this case is similar to the proof of theorem 5.1.

This proves that \((\hat{T}, \hat{\beta})\) is actually the maximizer of \(L_{N,T}(\cdot)\) over the entire parameter space \(\Theta_{N,T}\). Moreover, any other maximizer \((\hat{T}_{N,T}, \hat{\beta}_{N,T})\) of \(L_{N,T}(\cdot)\) must be inside \(B\). Otherwise, suppose \((\hat{T}_{N,T}, \hat{\beta}_{N,T})\) maximizes \(L_{N,T}(\cdot)\) and \((\hat{T}_{N,T}, \hat{\beta}_{N,T}) \in \Theta_{N,T}\backslash B\). Then for \(\varepsilon = \min((2\delta)^k d_1^2, 2^k \ell^2 \mu d_2^2)\) we have with probability at least \(1 - O(1/\varepsilon^2 N^p)\),

\[
L_{N,T}(\hat{T}_{N,T}, \hat{\beta}_{N,T}) + \varepsilon/3 > \ell_N^{\ast}(\hat{T}_{N,T}, \hat{\beta}_{N,T}) > \ell_N^{\ast}(\hat{T}_{N,T}, \hat{\beta}_{N,T}) + \varepsilon > L_{N,T}(\hat{T}_{N,T}, \hat{\beta}_{N,T}) + 2\varepsilon/3
\]

The first and the third inequality uses lemma D.5 and the second inequality uses lemma D.17. Therefore, with probability at least \(1 - O(1/(\varepsilon^2 N^p))\) all the maximizers of \(L_{N,T}(\cdot)\) must be inside the ball \(B\). This proves that for any pair of distances \(d_1, d_2\) we can choose \(N\) large enough such that with high probability \(1 - 1/\text{poly}(N)\) the maximizer of \(L_{N,T}(\cdot)\) lies within a \((d_1, d_2)\) neighborhood of \((T^*_N, \beta^*)\). This proves the consistency of the estimate when \(N\) increases to infinity. The proof of consistency when the number of time periods \(T\) increases to infinity is similar.

### D.1.4 Additional Lemmata

**Lemma D.1.**

\[
\mathbb{E}_{V_{i,t}, Y_{i,t}}[w_{i,t} \log \Pr \{Y_{i,t} \mid T\}] = 2^k \sum_{a_{i,t-k-1:t}} \Pr \{a_{i,t-k-1:t} \} \times \\
\int \log \Pr \{Y_{i,t} = Y_{i,t}(a_{i,t-k-1:t}) \mid T\} \Pr \{Y_{i,t} = Y_{i,t}(a_{i,t-k-1:t}) \mid T^*_N \} \, dY_{i,t}
\]
Proof.

\[ \mathbb{E}_{V_{i,t}, Y_{i,t}} \left[ w_{i,t} \log \Pr \left[ Y_{i,t} | T \right] \right] \]
\[ = \mathbb{E}_{V_{i,t}} \left[ \mathbb{E}_{Y_{i,t}} \left[ w_{i,t} \log \Pr \left[ Y_{i,t} | T \right] \right] \right] \]
\[ = \mathbb{E}_{V_{i,t}} \left[ w_{i,t} \int \log \Pr \left[ Y_{i,t} | T \right] \Pr \left[ Y_{i,t} | T_{N,T}^* \right] dY_{i,t} \right] \]
\[ = \int \sum_{y_{i,t-1}} \sum_{a_{i,t-1}} \Pr \left[ a_{i,t-1}, x_{i,t-1}, y_{i,1:t-1} \right] w_{i,t}(a_{i,1:t}, y_{i,1:t-1}) \times \]
\[ \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T_{N,T}^* \right] dY_{i,t} dY_{i,1:t-1} \]
\[ = \int \sum_{y_{i,t-2k+1:t-1}} \sum_{a_{i,t-2k+1:t}} \Pr \left[ a_{i,t-2k+1:t}, y_{i,t-2k+1:t-1} \right] w_{i,t}(a_{i,1:t}, y_{i,1:t-1}) \times \]
\[ \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T_{N,T}^* \right] dY_{i,t} dY_{i,1:t-1} \]

The last step marginalizes out the history from time 1 to time \( t - 2k \) and the covariates from time \( t - 2k + 1 \) to \( t \).

\[ \int \sum_{y_{i,t-2k+1:t-1}} \sum_{a_{i,t-2k+1:t}} \Pr \left[ a_{i,t-2k+1:t}, y_{i,t-2k+1:t-1} \right] \left[ \prod_{s=t-k+1}^{t} \Pr \left[ a_{i,s} | a_{i,t-2k+1:s-1} \right] \right] \times \]
\[ \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T_{N,T}^* \right] dY_{i,t} dY_{i,1:t-1} \]
\[ = \sum_{a_{i,t-k+1:t}} \Pr \left[ a_{i,t-k+1:t} \right] \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T_{N,T}^* \right] dY_{i,t} \]
\[ = 2^k \sum_{a_{i,t-k+1:t}} \Pr \left[ a_{i,t-k+1:t} \right] \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T_{N,T}^* \right] dY_{i,t} \]

\[ \square \]
Lemma D.2.

\[ EV_{i,t} \left[ w_{i,t} \log \Pr \left[ Y_{i,t} | T, \beta \right] \right] = \sum_{z_i} \Pr \left[ z_i \right] \sum_{a_{i,t-k+1:t}} \Pr \left[ a_{i,t-k+1:t} \right] \]

\[ \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i) | T, \beta \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i) | T_{N,T}^*, \beta^n \right] dY_{i,t} \]

Proof.

\[ EV_{i,t} \left[ w_{i,t} \log \Pr \left[ Y_{i,t} | T, \beta \right] \right] = EV_{i,t} \left[ EV_{Y_{i,t}} \left[ w_{i,t} \log \Pr \left[ Y_{i,t} | T, \beta \right] \right] \right] \]

\[ = EV_{i,t} \left[ w_{i,t} \int \log \Pr \left[ Y_{i,t} | T, \beta \right] \Pr \left[ Y_{i,t} | T_{N,T}^*, \beta^n \right] dY_{i,t} \right] \]

\[ = \sum_{z_i} \Pr \left[ z_i \right] \int_{y_{i,t-1}} \sum_{x_{i,t}} \sum_{a_{i,t}} \Pr \left[ a_{i,1:t}, x_{i,1:t}, y_{i,1:t-1} \right] w_{i,t}(a_{i,1:t}, x_{i,1:t}, y_{i,1:t-1}) \times \]

\[ \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i) | T, \beta \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i) | T_{N,T}^*, \beta^n \right] dY_{i,t} dY_{i,t-1} \]

\[ = \sum_{z_i} \Pr \left[ z_i \right] \int_{y_{i,t-2k+1:t-1}} \sum_{x_{i,t-2k+1:t}} \sum_{a_{i,t-2k+1:t}} \Pr \left[ a_{i,t-2k+1:t}, x_{i,t-2k+1:t}, y_{i,t-2k+1:t-1} \right] \times \]

\[ w_{i,t}(a_{i,t-2k+1:t}, x_{i,t-2k+1:t}, y_{i,t-2k+1:t-1}) \times \]

\[ \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i) | T, \beta \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i) | T_{N,T}^*, \beta^n \right] dY_{i,t} dY_{i,t-2k+1:t-1} \]
The last step marginalizes out the history from time $1$ to time $t-2k$.

$$
\sum_{z_i} \Pr[z_i] \int y_{i,t-2k+1:t-1} \sum_{z_{i,t-2k+1:t}} \sum_{a_{i,t-2k+1:t}} \Pr[a_{i,t-2k+1:t}, x_{i,t-2k+1:t}, y_{i,t-2k+1:t}] \times \\
\prod_{s=t-k+1}^{t} \Pr[a_{i,s}|a_{i,t-2k+1:s-1}] \times \\
Pr[a_{i,t-k+1:t}|a_{i,t-2k+1:t-k}, x_{i,t-2k+1:t}, Y_{i,t-2k+1:t-1}] \times \\
\int \log \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i)|T, \beta] \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i)|T_{N,T}^*, \beta^*] dY_{i,t} dY_{i,t-2k+1:t-1} \\
= \sum_{z_i} \Pr[z_i] \int y_{i,t-2k+1:t-1} \sum_{z_{i,t-2k+1:t}} \sum_{a_{i,t-2k+1:t}} \Pr[a_{i,t-2k+1:t-k}, x_{i,t-2k+1:t}, y_{i,t-2k+1:t}] \times \\
\Pr[a_{i,t-k+1:t}|a_{i,t-2k+1:t-k}] \times \\
\int \log \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i)|T, \beta] \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i)|T_{N,T}^*, \beta^*] dY_{i,t} \\
= \sum_{z_i} \Pr[z_i] \sum_{a_{i,t-k+1:t}} \Pr[a_{i,t-k+1:t}] \int \log \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i)|T, \beta] \times \\
\Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}, z_i)|T_{N,T}^*, \beta^*] dY_{i,t}
\]

\[ \square \]

**Lemma D.3.** $\ell_{N,T}^*(T_{N,T}) \leq \ell_{N,T}^*(T_{N,T}^*)$ for any tensor $T_{N,T}$.

**Proof.** Fix $i, t$ and $a_{i,t-k+1:t}$. Then conditioned on $T_{N,T}^*$, $Y_{i,t}$ is distributed according to a normal distribution with mean $T_{N,T}^*(i, t, a_{i,t-k+1:t}) (= \mu^* \text{ say})$ and variance $1$. And conditioned on $T_{N,T}$, $Y_{i,t}$ is distributed according to a normal distribution with mean $T_{N,T}(i, t, a_{i,t-k+1:t}) (= \mu \text{ say})$.
and variance 1. Now consider the following term:

\[
\int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T} \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T}^* \right] dY_{i,t} \\
- \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T}^* \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T} \right] dY_{i,t} \\
= \int \log \left( \frac{\Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T} \right]}{\Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T}^* \right]} \right) \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T}^* \right] dY_{i,t} \\
\leq \int \left( \frac{\Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T} \right]}{\Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T}^* \right]} - 1 \right) \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T} \right] dY_{i,t} \\
= \int \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T} \right] dY_{i,t} - \int \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T}^* \right] dY_{i,t} = 1 - 1 = 0
\]

This proves that

\[
\int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T} \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T}^* \right] dY_{i,t} \\
\leq \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T}^* \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \middle| T_{N,T} \right] dY_{i,t}
\]

Since the above holds for any \( i, t \) and \( a_{i,t-k+1:t} \) the claim follows from the expression of \( \ell_{N,T}^*(T_{N,T}) \) given in eq. (D.1).

Lemma D.4. Suppose \( T_{N,T}^* \) exists for all \( N \) and \( T \). Then,

1. If \( k \leq O \left( \log_{2(1-\delta)/\delta} N \right) \), \( \Pr \left[ \left| L_{N,T}(T_{N,T}) - \ell^*(T_{N,T}) \right| > \varepsilon \right] \to 0 \) as \( N \to \infty \).

2. If assumption Assn.2 holds and \( k \leq \max \left\{ O \left( \log_4 (T/\log T) \right), O \left( \log_{2(1-\delta)/\delta} T \right) \right\} \),

\( \Pr \left[ \left| L_{N,T}(T_{N,T}) - \ell^*(T_{N,T}) \right| > \varepsilon \right] \to 0 \) as \( T \to \infty \).
Proof.

\[
\Pr \left[ |L_{N,T}(\mathbf{T}_{N,T}) - \ell^*(\mathbf{T}_{N,T})| > \epsilon \right] \leq \frac{\Var (L_{N,T}(\mathbf{T}_{N,T}))}{\epsilon^2} = \frac{1}{\epsilon^2 N^2 T^2} \Var \left( \sum_{i,t} w_{i,t} \log \Pr [Y_{i,t}|\mathbf{T}_{N,T}] \right)
\]

\[
= \frac{1}{\epsilon^2 N^2 T^2} \left[ \sum_{i,t} \Var (w_{i,t} \log \Pr [Y_{i,t}|\mathbf{T}_{N,T}]) + 2 \sum_{i,j<t} \cov (w_{i,t} \log \Pr [Y_{i,t}|\mathbf{T}_{N,T}], w_{i,j} \log \Pr [Y_{i,j}|\mathbf{T}_{N,T}]) \right] +
\]

\[
\leq \frac{1}{\epsilon^2 N^2 T^2} \left[ \sum_{i,t} \mathbb{E} \left[ w_{i,t}^2 \log^2 \Pr [Y_{i,t}|\mathbf{T}_{N,T}] \right] + 2 \sum_{i,j<t} \mathbb{E} [w_{i,t} w_{i,j} \log \Pr [Y_{i,t}|\mathbf{T}_{N,T}] \log \Pr [Y_{i,j}|\mathbf{T}_{N,T}]] +
\]

Now we bound each term in the summation. Fix any \( a_{i,t-k+1:t} = p \). This fixes the distribution of \( Y_{i,t} \). Then we have,

\[
\mathbb{E}_{Y_{i,t}} \left[ \log^2 \Pr [Y_{i,t}|\mathbf{T}_{N,T}] \right] = \int \log^2 \Pr [Y_{i,t}|\mathbf{T}_{N,T}] \Pr [Y_{i,t}|\mathbf{T}^*_N] \, dY_{i,t}
\]

\[
= \log^2 (\sqrt{2\pi}) + \mathbb{E}_{Y_{i,t} \sim N(\mathbf{T}^*_N(i,t,p),1)} \left[ (Y_{i,t} - \mathbf{T}_{N,T}(i,t,p))^4 / 4 \right]
\]

Since both \( \mathbf{T}^*_N(i,t,p) \) and \( \mathbf{T}_{N,T}(i,t,p) \) are bounded by \( L \), there exists a constant \( L_1 \) such that

\[
\mathbb{E}_{Y_{i,t}} \left[ \log^2 \Pr [Y_{i,t}|\mathbf{T}_{N,T}] \right] \leq L_1.
\]

This gives the following bound.

\[
\mathbb{E}_{Y_{i,t}} \left[ w_{i,t}^2 E_{Y_{i,t}} \left[ \log^2 \Pr [Y_{i,t}|\mathbf{T}_{N,T}] \right] \right] \leq \left( \frac{1-\delta}{\delta} \right)^k \mathbb{E}_{Y_{i,t}} \left[ w_{i,t} E_{Y_{i,t}} \left[ \log^2 \Pr [Y_{i,t}|\mathbf{T}_{N,T}] \right] \right]
\]

\[
\leq \left( \frac{1-\delta}{\delta} \right)^k 2^k \sum_{a_{i,t-k+1:t}} \Pr [a_{i,t-k+1:t}] \times \int \log^2 \Pr [Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|\mathbf{T}_{N,T}] \Pr [Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|\mathbf{T}^*_N] \, dY_{i,t}
\]

\[
\leq L_1 \left( \frac{2(1-\delta)}{\delta} \right)^k \sum_{a_{i,t-k+1:t}} \Pr [a_{i,t-k+1:t}] = L_1 \left( \frac{2(1-\delta)}{\delta} \right)^k
\]

We use lemma \( \text{D.6} \) for \( \circledast \) and the derivation of equation \( \circledast \) is similar to lemma \( \text{D.1} \).
We now consider the remaining terms. There are two cases.

**Case 1**: Suppose \( t < t' \leq t + 2k \).

\[
E[w_{i,t}w_{i,t'} \log \Pr[Y_{i,t}|T_{N,T}] \log \Pr[Y_{i,t'}|T_{N,T}]] \\
\leq E[w_{i,t}w_{i,t'} | \log \Pr[Y_{i,t}|T_{N,T}] | \log \Pr[Y_{i,t'}|T_{N,T}]] \\
\leq \left( \frac{1 - \delta}{\delta} \right)^k E[w_{i,t} | \log \Pr[Y_{i,t}|T_{N,T}] | \log \Pr[Y_{i,t'}|T_{N,T}]]
\]

By an argument same as before, for any realization of \( a_{i,t',k+1:t'} \) we can bound 
\( E_{Y_{i,t'}}[\log \Pr[Y_{i,t'}]] \) by a constant, say \( L_2 \). This gives us the following bound.

\[
L_2 \left( \frac{1 - \delta}{\delta} \right)^k E[w_{i,t} | \log \Pr[Y_{i,t}|T_{N,T}]] \\
= L_2 \left( \frac{1 - \delta}{\delta} \right)^k \sum_{a_{i,t-k+1:t}} \Pr[a_{i,t-k+1:t}] \times \\
\int \log \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T_{N,T}] \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T^*_{N,T}] \, dY_{i,t} \\
\leq L_2^2 \left( \frac{2(1 - \delta)}{\delta} \right)^k
\]

**Case 2**: Suppose \( t' > t + 2k \). We can bound the corresponding covariance term by 
\( L_2^2 \left( \frac{2(1 - \delta)}{\delta} \right)^k \) by an argument same as before. This gives us the following bound:

\[
\Pr[[L_{N,T}(T_{N,T}) - \ell^*(T_{N,T})] > \epsilon] \leq \frac{1}{\epsilon^2 N^2 T^2} \left[ NTL_1 \left( \frac{2(1 - \delta)}{\delta} \right)^k + 2NT^2L_2^2 \left( \frac{2(1 - \delta)}{\delta} \right)^k \right] \\
= O \left( \frac{1}{\epsilon^2 N^p} \right)
\]

for any \( 0 < \epsilon < 1 \) as long as \( k = O \left( (1 - p) \log_{2(1 - \delta)/\delta} N \right) \). This gives us the first result.

Now suppose assumption 5.9 holds. Then, lemma D.14 gives us a bound of
The proof of the first part is same as lemma D.4 and we omit the proof here.

Lemma D.6.

\[ \text{for any } 0 < p < 1 \text{ as long as } k = O \left( (1 - p) \log_{2(1 - \delta)/\delta} T \right) \text{ and } 4^k \log T \leq T^{1-p}. \]

Lemma D.5. Suppose \( T^*_{N,T} \) exists for all \( N \) and \( T \). Then,

1. If \( k \leq O \left( \log_{2(1 - \delta)/\delta} N \right) \), \( \Pr \left[ |L_{N,T}(T_{N,T} \text{, } \ell^*(T_{N,T})), \beta_{N,T}) - \ell^*(T_{N,T}, \beta_{N,T})| > \varepsilon \right] \to 0 \) as \( N \to \infty \).

2. If assumption Asn.3 holds and \( k \leq \max \left\{ O \left( \log_4 (T / \log T) \right), O \left( \log_{2(1 - \delta)/\delta} T \right) \right\} \),
\[
\Pr \left[ |L_{N,T}(T_{N,T}) - \ell^*(T_{N,T})| > \varepsilon \right] \to 0 \text{ as } T \to \infty.
\]

Proof. Proceeding similarly as lemma D.4 we can first establish

\[
\Pr \left[ |L_{N,T}(T_{N,T}, \beta_{N,T}) - \ell^*(T_{N,T}, \beta_{N,T})| > \varepsilon \right] \leq \frac{1}{\varepsilon^2 N^2 T^2} \sum_{i,t} \mathbb{E} \left[ w_{i,t}^2 \log^2 \Pr [Y_{i,t} \mid T_{N,T}, \beta_{N,T}] \right] + 2 \sum_{i,t, t' : i + t < i + 2k} \mathbb{E} \left[ w_{i,t} w_{i,t'} \log \Pr [Y_{i,t} \mid T_{N,T}, \beta_{N,T}] \log \Pr [Y_{i,t'} \mid T_{N,T}, \beta_{N,T}] \right] + 2 \sum_{i,t, t' : i + 2k < t'} \text{cov} \left( w_{i,t} \log \Pr [Y_{i,t} \mid T_{N,T}, \beta_{N,T}], w_{i,t'} \log \Pr [Y_{i,t'} \mid T_{N,T}, \beta_{N,T}] \right)
\]

The proof of the first part is same as lemma D.4 and we omit the proof here.

Now suppose assumption Asn.3 holds. Then, lemma D.15 gives us a bound of \( O \left( 2^k (t' - t)^{-\gamma} \right) \) on the covariance term, for some constant \( \gamma > 0 \). The rest of the proof is similar to the proof of lemma D.14.

Lemma D.6. \( w_{i,t} \leq \left( \frac{1 - \delta}{\delta} \right)^k \)
Proof.
\[ w_{i,t} = \prod_{s=t-k+1}^{t} \Pr[A_{i,s} | A_{i,t-2k+1:s-1}] / \Pr[A_{i,s} | A_{i,t-2k+1:s-1}, Y_{i,t-2k+1:s-1}] \]

Recall that the given policy satisfies positivity with constant \( \delta \) i.e. for each \( a_{i,1:t-1}, x_{i,1:t-1}, y_{i,1:t-1} \), we have
\[ \delta < \Pr[a_{i,t}|a_{i,1:t-1}, x_{i,1:t-1}, y_{i,1:t-1}] < 1 - \delta \]

Positivity implies that
\[ \Pr[A_{i,s} | A_{i,t-2k+1:s-1}, Y_{i,t-2k+1:s-1}] = \Pr[A_{i,s} | A_{i,t-2k+1:s-1}, X_{i,t-2k+1:s-1}, Y_{i,t-2k+1:s-1}] \Pr[A_{i,1:t-2k}, X_{i,1:s-1}, Y_{i,1:s-1}] \leq 1 - \delta. \] Now consider the term in the denominator.

\[ \Pr[A_{i,s} | A_{i,t-2k+1:s-1}, Y_{i,t-2k+1:s-1}] = \sum_{x_{i,t-2k+1:s-1}} \Pr[A_{i,s} | A_{i,t-2k+1:s-1}, x_{i,t-2k+1:s-1}, Y_{i,t-2k+1:s-1}] \times \Pr[x_{i,t-2k+1:s-1} | A_{i,t-2k+1:s-1}, Y_{i,t-2k+1:s-1}] \geq \delta \sum_{x_{i,t-2k+1:s-1}} \Pr[x_{i,t-2k+1:s-1} | A_{i,t-2k+1:s-1}, Y_{i,t-2k+1:s-1}] = \delta \]

These two results imply that each term in the product of \( w_{i,t} \) is bounded by \((1 - \delta)/\delta\) and we get the desired bound on \( w_{i,t} \).

\[ \square \]

**Lemma D.7.** Let \( S(i, t) \) be the random variable denoting the slice chosen by the policy for user \( i \) at time \( t \). If assumption 5.9 holds, then \( \text{cov}(w_{i,t}T^2(i, t, S(i, t)), w_{i,t'}T^2(i, t', S(i, t'))) \leq O \left( L^2 \left( \frac{1 - \delta}{\delta} \right)^2 (t' - t)^{-\gamma} \right) \)

**Proof.** The proof is analogous to the proof of lemma D.9 and is omitted. \( \square \)

**Lemma D.8.** \( \ell_{N,T}^*(T) \) is concave in \( T \).

**Proof.** From equation D.1 we get
\[
\ell_{N,T}^*(T) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} 2^k \sum_{a_{i,t-k+1:t}} \Pr[a_{i,t-k+1:t}] \int \log \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T] \times \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) | T_{N,T}] dY_{i,t}
\]

185
Now fix any value of \( a_{i,t-1:t} \). Then \( \log \Pr \{ Y_{i,t} = Y_{i,t}(a_{i,t-1:t}) | T \} \) is concave in \( T \). Then
\[
\int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-1:t}) | \lambda T_1 + (1 - \lambda) T_2 \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-1:t}) | T_{N,T}^* \right] dY_{i,t}
\geq \int \{ \lambda \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-1:t}) | T_1 \right] + (1 - \lambda) \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-1:t}) | T_2 \right] \} \times
\Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-1:t}) | T_{N,T}^* \right] dY_{i,t}
= \lambda \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-1:t}) | T_1 \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-1:t}) | T_{N,T}^* \right] dY_{i,t}
+ (1 - \lambda) \int \log \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-1:t}) | T_2 \right] \Pr \left[ Y_{i,t} = Y_{i,t}(a_{i,t-1:t}) | T_{N,T}^* \right] dY_{i,t}
\]
This proves that the function inside the integral is concave. Since \( \ell_{N,T}^*(T) \) is just a non-negative weighted sum of such functions, it is also concave. \( \square \)

Lemma D.9. \( L_{N,T}(T_{N,T}) \) is concave in \( T_{N,T} \).

Proof.
\[
L_{N,T}(T_{N,T}) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} w_{i,t} \log \Pr \left[ Y_{i,t} | T_{N,T} \right]
= -\frac{1}{2NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} (Y_{i,t} - T_{N,T}(i,t,p))^2 + \text{const}
\]
Each term inside the summation i.e. \( -(Y_{i,t} - T_{N,T}(i,t,p))^2 \) is a concave function. The likelihood function is a non-negative weighted sum of concave functions and is also concave. \( \square \)

Lemma D.10. \( L_{N,T}(T_{N,T}, \beta_{N,T}) \) is jointly concave in \( T_{N,T}, \beta_{N,T} \).

Proof.
\[
L_{N,T}(T_{N,T}, \beta_{N,T}) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} w_{i,t} \log \Pr \left[ Y_{i,t} | T_{N,T}, \beta_{N,T} \right]
= -\frac{1}{2NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} (Y_{i,t} - T_{N,T}(i,t,p) - Z_i' \beta_{N,T})^2 + \text{const}
\]
Each term inside the summation i.e. \( -(Y_{i,t} - T_{N,T}(i,t,p) - Z_i' \beta_{N,T})^2 \) is a concave function in \( T_{N,T}(i,t,p) + Z_i' \beta_{N,T} \). This implies that \( -(Y_{i,t} - T_{N,T}(i,t,p) - Z_i' \beta_{N,T})^2 \) is jointly concave in \( T_{N,T}(i,t,p), \beta_{N,T} \). The likelihood function is a non-negative weighted sum of concave functions and is also concave. \( \square \)
Lemma D.11.

\[
E \left[ \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} T^2(i, t, p) \right] = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{p=1}^{B} \Pr [(i, t) \in O_p] T^2(i, t, p)
\]

Proof. Let \( S(i, t) \) be the random variable denoting the slice selected by the policy for subject \( i \) at time \( t \). From the linearity of expectation, we have

\[
E \left[ \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} T^2(i, t, p) \right] = \sum_{i=1}^{N} \sum_{t=1}^{T} E \left[ w_{i,t} T^2(i, t, S(i, t)) \right]
\]

We now consider each term inside the summation.

\[
E \left[ w_{i,t} T^2(i, t, S(i, t)) \right] = \int_{y_{i,t}} \sum_{s(i,t)} \sum_{a_{i,t,t}} \Pr [a_{i,1:t}, x_{i,1:t}, y_{i,1:t-1}] w_{i,t}(a_{i,1:t}, x_{i,1:t}, y_{i,1:t-1}) T^2(i, t, s(i, t))
\]

\[
= \int_{y_{i,t}} \sum_{x_{i,t}} \sum_{a_{i,t},t} \Pr [a_{i,t-2k+1:t}, x_{i,t-2k+1:t}, y_{i,t-2k+1:t-1}] \times
w_{i,t}(a_{i,t-2k+1:t}, x_{i,t-2k+1:t}, y_{i,t-2k+1:t-1}) T^2(i, t, s(i, t))
\]

\[
= \int_{y_{i,t}} \sum_{x_{i,t}} \sum_{a_{i,t},t} \Pr [a_{i,t-2k+1:t}, x_{i,t-2k+1:t}, y_{i,t-2k+1:t-1}] \times
\frac{\Pr [a_{i,t-k+1:t} | a_{i,t-2k+1:t-k}] T^2(i, t, s(i, t))}{\Pr [a_{i,t-k+1:t} | a_{i,t-2k+1:t-k}, x_{i,t-2k+1:t}, y_{i,t-2k+1:t-1}]} \times
\]

\[
= \int_{y_{i,t}} \sum_{x_{i,t}} \sum_{a_{i,t},t} \Pr [a_{i,t-2k+1:t-k}, x_{i,t-2k+1:t}, y_{i,t-2k+1:t-1}] \times
\Pr [a_{i,t-k+1:t} | a_{i,t-2k+1:t-k}] T^2(i, t, s(i, t))
\]

\[
= \sum_{a_{i,t-2k+1:t}} \Pr [a_{i,t-2k+1:t-k}] \Pr [a_{i,t-k+1:t} | a_{i,t-2k+1:t-k}] T^2(i, t, s(i, t))
\]

\[
= \sum_{a_{i,t-2k+1:t-k}} \Pr [a_{i,t-2k+1:t-k}] T^2(i, t, s(i, t))
\]

\[
= \sum_{p=1}^{B} \Pr [S(i, t) = p] T^2(i, t, p)
\]

\[\square\]

Lemma D.12.

\[
\Pr \left[ \frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} T^2(i, t, p) - \frac{1}{NT} \| T \|_W^2 \geq \varepsilon \right] \leq 2 \exp \left( - \frac{2N\varepsilon^2}{L^4 \left( \frac{1-\gamma}{\alpha} \right)^{2k}} \right)
\]

187
Proof. Suppose $S(i, t)$ be the slice selected by the policy for agent $i$ at time $t$. Then we have
\[
\frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} T^2(i, t, p) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} w_{i,t} T^2(i, t, S(i, t)).
\]
Observe that for each $i$,
\[
\frac{1}{T} \sum_{t=1}^{T} w_{i,t} T^2(i, t, S(i, t)) \in \left[ 0, L^2 \left( \frac{1}{\delta} \right)^k \right].
\]
Now we apply the Hoeffding inequality considering the random variables
\[
\left\{ \frac{1}{T} \sum_{t=1}^{T} w_{i,t} T^2(i, t, S(i, t)) \right\}_{i=1}^{N}
\]
as independent random variables and get the following bound.
\[
\Pr \left[ \left| \frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} T^2(i, t, p) - \frac{1}{NT} \| T \|^2_W \right| \geq \varepsilon \right] \leq 2 \exp \left( - \frac{2N\epsilon^2}{L^4 \left( \frac{1}{\delta} \right)^{2k}} \right)
\]
(D.3)

Lemma D.13.
\[
\Pr \left[ \left| \frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} T^2(i, t, p) - \frac{1}{NT} \| T \|^2_W \right| \geq \varepsilon \right] \leq O \left( \frac{L^4 \left( \frac{1}{\delta} \right)^{2k}}{\epsilon^2T/\log T} \right)
\]

Proof. Suppose $S(i, t)$ be the slice selected by the policy for agent $i$ at time $t$. Then we have
\[
\frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} T^2(i, t, p) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} w_{i,t} T^2(i, t, S(i, t)).
\]
Observe that for each $i$,
\[
\frac{1}{N} \sum_{i=1}^{N} w_{i,t} T^2(i, t, S(i, t)) \in \left[ 0, L^2 \left( \frac{1}{\delta} \right)^k \right].
\]
Now we apply the Chebyshev inequality considering the random variables

188
\[
\left\{ \frac{1}{N} \sum_{i=1}^{N} w_{i,t} T^2(i, t, S(i, t)) \right\}_{i=1}^{T}.
\]

\[
\Pr \left[ \left| \frac{1}{NT} \sum_{p=1}^{R} \sum_{(i,t) \in O_p} w_{i,t} T^2(i, t, p) - \frac{1}{NT} \|T\|_W \right| \geq \varepsilon \right] \leq \frac{\Var \left( \frac{1}{NT} \sum_{i=1}^{T} \sum_{i=1}^{N} w_{i,t} T^2(i, t, S(i, t)) \right)}{\varepsilon^2}
\]

\[
= \frac{1}{\varepsilon^2 T^2} \left\{ \sum_{t=1}^{T} \Var \left( \frac{1}{N} \sum_{i=1}^{N} w_{i,t} T^2(i, t, S(i, t)) \right) + 2 \sum_{t' - t < 2k} \Cov \left( \frac{1}{N} \sum_{i=1}^{N} w_{i,t} T^2(i, t, S(i, t)), \frac{1}{N} \sum_{i=1}^{N} w_{i,t'} T^2(i, t', S(i, t')) \right) \right\}
\]

\[
\leq \frac{1}{\varepsilon^2 T^2} \left\{ TL^4 \left( \frac{1-\delta}{\delta} \right)^2 + 2TL^4 \left( \frac{1-\delta}{\delta} \right)^2 \sum_{t' - t \geq 2k} L^4 \left( \frac{1-\delta}{\delta} \right)^2 (t' - t)^{-\gamma} \right\}
\]

\[
\leq \frac{1}{\varepsilon^2 T^2} \left\{ TL^4 \left( \frac{1-\delta}{\delta} \right)^2 + 2TL^4 \left( \frac{1-\delta}{\delta} \right)^2 + 2T \log TL^4 \left( \frac{1-\delta}{\delta} \right)^2 \right\} = O \left( \frac{L^4 \left( \frac{1-\delta}{\delta} \right)^2}{\varepsilon^2 T^2} \right)
\]

\lemma{D.14.} Suppose \( t' > t + 2k \) and assumption \textbf{Asn.2} holds. Then the following is true.

\[
|\Cov (w_{i,t} \log \Pr [Y_{i,t} | T_{N,T}], w_{i,t'} \log \Pr [Y_{i,t'} | T_{N,T}])| \leq O \left( 4^k (t' - t)^{-\gamma} \right)
\]

\proof{} Let us write \( \mathcal{H}_{i,1:t} = (a_{i,1:t}, x_{i,1:t}, y_{i,1:t-1}) \) to denote the history upto time \( t \) excluding the
outcome at time $t$.

$$
\begin{align*}
|\text{cov} (w_{i,t} \log \Pr [Y_{i,t}|T_{N,T}], w_{i,t'} \log \Pr [Y_{i,t'}|T_{N,T}])|
&= |E [w_{i,t} \log \Pr [Y_{i,t}|T_{N,T}] w_{i,t'} \log \Pr [Y_{i,t'}|T_{N,T}]]
- E [w_{i,t} \log \Pr [Y_{i,t}|T_{N,T}]] E [w_{i,t'} \log \Pr [Y_{i,t'}|T_{N,T}]]
- |E [w_{i,t} w_{i,t'} | \log \Pr [Y_{i,t}|T_{N,T}] | \log \Pr [Y_{i,t'}|T_{N,T}]]
- E [w_{i,t} | \log \Pr [Y_{i,t}|T_{N,T}] |] E [w_{i,t'} | \log \Pr [Y_{i,t'}|T_{N,T}] |]
\end{align*}
$$

$$
\begin{align*}
&= \left| \sum_{a_{i,t}, x_{i,t}, a_{i,t'}} \int_{y_{i,t}} \Pr [\mathcal{H}_{i,1:t'}] w_{i,t}(\mathcal{H}_{i,1:t'}) | \log \Pr [Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T_{N,T}] \\
&\times w_{i,t'}(\mathcal{H}_{i,1:t'}) | \log \Pr [Y_{i,t'} = Y_{i,t'}(a_{i,t'-k+1:t'})|T_{N,T}] | \Pr [Y_{i,t'} = Y_{i,t'}(a_{i,t'-k+1:t'})|T_{N,T}] \right| dY_{i,1:t'} \\
&\quad - \sum_{a_{i,t}, x_{i,t}, a_{i,t'}} \int_{y_{i,t}} \Pr [\mathcal{H}_{i,1:t}] w_{i,t}(\mathcal{H}_{i,1:t}) | \log \Pr [Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T_{N,T}] \\
&\times \sum_{a_{i,t}, x_{i,t}, a_{i,t'}} \int_{y_{i,1:t'}} \Pr [\mathcal{H}_{i,1:t'}] w_{i,t'}(\mathcal{H}_{i,1:t'}) | \log \Pr [Y_{i,t'} = Y_{i,t'}(a_{i,t'-k+1:t'})|T_{N,T}] | \Pr [Y_{i,t'} = Y_{i,t'}(a_{i,t'-k+1:t'})|T_{N,T}] \right| dY_{i,1:t'}
\end{align*}
$$

Since logarithm of the probabilities are negative, the second equality changes them to their absolute values. The next equality just expands the individual terms. Integrating out the
covariates we get,

\[
\sum_{a_{i,t}} \int_{y_{1:t}} \Pr[a_{i,t}, y_{1:t-1}] w_{i,t}(a_{i,t}, y_{1:t-1}) \log \Pr[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T^*_{N,T}] \times \\
\Pr\left[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T^*_{N,T}\right] dY_{i,t}
\]

\[
\times \sum_{\tilde{a}_{i,t+1:t'}} \int_{\tilde{y}_{i,t+1:t'}} \Pr[\tilde{a}_{i,t+1:t'}, \tilde{y}_{i,t+1:t'-1}|a_{i,t}, y_{1:t}] w_{i,t'}(\tilde{a}_{i,t+1:t'}, \tilde{y}_{i,t+1:t'-1})
\]

\[
\times \log \Pr\left[\tilde{Y}_{i,t'} = Y_{i,t'}(a_{i,t'-k+1:t'})|T^*_{N,T}\right] \Pr\left[\tilde{Y}_{i,t'} = Y_{i,t'}(a_{i,t'-k+1:t'})|T^*_{N,T}\right] dY_{i,t+1:t'}
\]

\[
- \sum_{a_{i,t}} \int_{y_{1:t}} \Pr[a_{i,t}, y_{1:t-1}] w_{i,t}(a_{i,1:t}, y_{1:t-1}) \log \Pr\left[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T^*_{N,T}\right]
\]

\[
\times \Pr\left[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T^*_{N,T}\right] dY_{i,t}
\]

\[
\times \sum_{\tilde{a}_{i,t+1:t'}} \int_{\tilde{y}_{i,t+1:t'}} \Pr[\tilde{a}_{i,t+1:t'}, \tilde{y}_{i,t+1:t'-1}] w_{i,t'}(\tilde{a}_{i,t+1:t'}, \tilde{y}_{i,t+1:t'-1}) \log \Pr\left[\tilde{Y}_{i,t'} = Y_{i,t'}(\tilde{a}_{i,t'-k+1:t'})|T^*_{N,T}\right]
\]

\[
\Pr\left[\tilde{Y}_{i,t'} = Y_{i,t'}(\tilde{a}_{i,t'-k+1:t'})|T^*_{N,T}\right] dY_{i,t+1:t'}
\]

= \sum_{a_{i,1:t}, y_{1:t-1}} \Pr[a_{i,1:t}, y_{1:t-1}] w_{i,t}(a_{i,1:t}, y_{1:t-1}) \log \Pr\left[Y_{i,t} = Y_{i,t}(a_{i,1:t})|T^*_{N,T}\right]

\times \Pr\left[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t})|T^*_{N,T}\right] dY_{i,t}

\times \sum_{\tilde{a}_{i,t+1:t'}} \int_{\tilde{y}_{i,t+1:t'}} \Pr[\tilde{a}_{i,t+1:t'}, \tilde{y}_{i,t+1:t'-1}|a_{i,t}, y_{1:t}] - \Pr[\tilde{a}_{i,t+1:t'}, \tilde{y}_{i,t+1:t'-1}] w_{i,t'}(\tilde{a}_{i,t+1:t'}, \tilde{y}_{i,t+1:t'-1})

\times \log \Pr\left[\tilde{Y}_{i,t'} = Y_{i,t'}(\tilde{a}_{i,t'-k+1:t'})|T^*_{N,T}\right] \Pr\left[\tilde{Y}_{i,t'} = Y_{i,t'}(\tilde{a}_{i,t'-k+1:t'})|T^*_{N,T}\right] d\tilde{Y}_{i,t+1:t'}

Now we marginalize the history from time \( t+1 \) to \( t'-2k \) from the last summation and
integration. This gives us the following bound on the covariance:

\[
\sum_{a_{i,t}, j} \int_{Y_{i,t}} \Pr \left[ a_{i,1:t} y_{i,1:t-1} \right] w_{i,t}(a_{i,1:t}, y_{i,1:t-1}) \left| \log \Pr \left[ Y_{i,t} = Y_{i,t} (a_{i,t-k+1:t}) \right| T_{N,T} \right] \\
\times \Pr \left[ Y_{i,t} = Y_{i,t} (a_{i,t-k+1:t}) \right| T_{N,T} \right] dY_{i,t} \times \sum_{\tilde{a}_{i,t'} - 2k+1:t'} \int_{\tilde{y}_{i,t'} - 2k+1:t'} \left( \Pr \left[ \tilde{a}_{i,t'} - 2k+1:t', \tilde{y}_{i,t'} - 2k+1:t' - 1 \right| a_{i,1:t}, y_{i,1:t} \right] - \Pr \left[ \tilde{a}_{i,t'} - 2k+1:t', \tilde{y}_{i,t'} - 2k+1:t' - 1 \right] \right) \\
\times \log \Pr \left[ \tilde{Y}_{i,t'} = Y_{i,t'} (\tilde{a}_{i,t'} - k+1:t') \right| T_{N,T} \right] d\tilde{Y}_{i,t'} - 2k+1:t'
\]

We now consider the following term:

\[
\sum_{\tilde{a}_{i,t'} - 2k+1:t'} \int_{\tilde{y}_{i,t'} - 2k+1:t'} \left( \Pr \left[ \tilde{a}_{i,t'} - 2k+1:t', \tilde{y}_{i,t'} - 2k+1:t' - 1 \right| a_{i,1:t}, y_{i,1:t} \right] - \Pr \left[ \tilde{a}_{i,t'} - 2k+1:t', \tilde{y}_{i,t'} - 2k+1:t' - 1 \right] \right) \\
\times \log \Pr \left[ \tilde{Y}_{i,t'} = Y_{i,t'} (\tilde{a}_{i,t'} - k+1:t') \right| T_{N,T} \right] d\tilde{Y}_{i,t'} - 2k+1:t'
\]

We simplify two terms.

\[
\Pr \left[ \tilde{a}_{i,t'} - 2k+1:t', \tilde{y}_{i,t'} - 2k+1:t' - 1 \right] \times \Pr \left[ \tilde{a}_{i,t'} - k+1:t' \right| \tilde{a}_{i,t'} - 2k+1:t' - k \right] \\
= \Pr \left[ \tilde{a}_{i,t'} - k+1:t' \right| \tilde{a}_{i,t'} - 2k+1:t' - k \right] \Pr \left[ \tilde{a}_{i,t'} - 2k+1:t' - k \right] \\
= \Pr \left[ \tilde{a}_{i,t'} - 2k+1:t' \right] \Pr \left[ \tilde{y}_{i,t'} - 2k+1:t' - 1 \right| \tilde{a}_{i,t'} - 2k+1:t' - k \right] \\
\]
And,

\[
\begin{align*}
\Pr \left[ a_{t',t' - 2k + 1} | a_{i,1:t', y_{1:i}} \right] & \times \frac{\Pr \left[ \tilde{a}_{i,t' - k + 1}, \tilde{y}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right]}{\Pr \left[ \tilde{a}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right]} \\
= & \Pr \left[ \tilde{y}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right] \Pr \left[ \tilde{a}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right] \\
& \times \Pr \left[ \tilde{y}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right] \Pr \left[ \tilde{a}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right]
\end{align*}
\]

Substituting the two results above in eq. D.5 and integrating out \( \tilde{y}_{i,t' - 2k + 1} \) we get,

\[
\begin{align*}
\sum_{\tilde{a}_{i,t' - 2k + 1}} \int_{\tilde{y}_{i,t' - k + 1}} \left( \Pr \left[ \tilde{a}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right] \Pr \left[ \tilde{y}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right] \Pr \left[ \tilde{y}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right] \right) \\
\times \left( \Pr \left[ \tilde{a}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right] \Pr \left[ \tilde{y}_{i,t' - k + 1}; a_{i,1:t', y_{1:i}} \right] \right) \\
\times \log \Pr \left[ \tilde{y}_{i,t'} = Y_{i,t'}(\tilde{a}_{i,t' - k + 1}) | T_{N,T} \right] \Pr \left[ \tilde{y}_{i,t'} = Y_{i,t'}(\tilde{a}_{i,t' - k + 1}) | T_{N,T} \right] d\tilde{y}_{i,t'}
\end{align*}
\]

Now integrating out \( \tilde{y}_{i,t' - k + 1} \) we get,

\[
\begin{align*}
\sum_{\tilde{a}_{i,t' - 2k + 1}} \int_{\tilde{y}_{i,t'}} \left( \Pr \left[ \tilde{a}_{i,t' - 2k + 1}; a_{i,1:t', y_{1:i}} \right] \Pr \left[ \tilde{y}_{i,t'} = Y_{i,t'}(\tilde{a}_{i,t' - k + 1}) | T_{N,T} \right] \Pr \left[ \tilde{y}_{i,t'} = Y_{i,t'}(\tilde{a}_{i,t' - k + 1}) | T_{N,T} \right] \\
\times \log \Pr \left[ \tilde{y}_{i,t'} = Y_{i,t'}(\tilde{a}_{i,t' - k + 1}) | T_{N,T} \right] \Pr \left[ \tilde{y}_{i,t'} = Y_{i,t'}(\tilde{a}_{i,t' - k + 1}) | T_{N,T} \right] d\tilde{y}_{i,t'}
\end{align*}
\]

Now observe that \( \Pr \left[ \tilde{a}_{i,t' - 2k + 1}; a_{i,1:t', y_{1:i}} \right] = \frac{\Pr \left[ \tilde{a}_{i,t' - 2k + 1}, a_{i,1:t', y_{1:i}} \right]}{\Pr \left[ a_{i,1:t', y_{1:i}} \right]} \)
\[
= \frac{\Pr[y_{i,t} | a_{i,t}] \Pr[\hat{a}_{i,t'} - 2k + 1:t'|a_{i,t}] \Pr[a_{i,t}]}{\Pr[\hat{a}_{i,t'} - 2k + 1:t' | y_{i,t}]} = \Pr[\hat{a}_{i,t'} - 2k + 1:t' | a_{i,t}].
\]
Substituting this result we get
\[
\sum_{\hat{a}_{i,t'} - 2k + 1:t'} \int \ldots \Pr[\hat{a}_{i,t'} - 2k + 1:t' | a_{i,t}] - \Pr[\hat{a}_{i,t'} - 2k + 1:t'] \Pr[\hat{a}_{i,t'} - 2k + 1:t']
\]
\[
\times \log \Pr[\tilde{Y}_{i,t'} = Y_{i,t'}(\hat{a}_{i,t'} - k + 1:t') | T_{N,T}] \Pr[\tilde{Y}_{i,t'} = Y_{i,t'}(\hat{a}_{i,t'} - k + 1:t') | T_{N,T}] d\tilde{Y}_{i,t'}
\]
\[
= \sum_{\hat{a}_{i,t'} - k + 1:t'} \int \ldots \Pr[\hat{a}_{i,t'} - k + 1:t' | a_{i,t}] - \Pr[\hat{a}_{i,t'} - k + 1:t'] \Pr[\hat{a}_{i,t'} - k + 1:t']
\]
\[
\times \log \Pr[\tilde{Y}_{i,t'} = Y_{i,t'}(\hat{a}_{i,t'} - k + 1:t') | T_{N,T}] \Pr[\tilde{Y}_{i,t'} = Y_{i,t'}(\hat{a}_{i,t'} - k + 1:t') | T_{N,T}] d\tilde{Y}_{i,t'}
\]
Using assumption \textbf{Asn.2} we can bound the difference \((\Pr[\hat{a}_{i,t'} - k + 1:t' | a_{i,t}] - \Pr[\hat{a}_{i,t'} - k + 1:t']]) by \(c(t' - t)^{-\gamma}\) for some constant \(c > 0\). This gives us the following bound on the previous term.
\[
c(t' - t)^{-\gamma} \sum_{\hat{a}_{i,t'} - k + 1:t'} \int \ldots \log \Pr[\tilde{Y}_{i,t'} = Y_{i,t'}(\hat{a}_{i,t'} - k + 1:t') | T_{N,T}] \Pr[\tilde{Y}_{i,t'} = Y_{i,t'}(\hat{a}_{i,t'} - k + 1:t') | T_{N,T}] d\tilde{Y}_{i,t'}
\]
\[
= c(t' - t)^{-\gamma} \sum_{\hat{a}_{i,t'} - k + 1:t'} E[\log \Pr[Y_{i,t'} | T_{N,T}] | \hat{a}_{i,t'} - k + 1:t']
\]
Since both \(T_{N,T}(i, t, p)\) and \(T_{N,T}(i, t, p)\) are bounded by \(L\) there exists a constant \(L_3 > 0\) such that \(E[\log \Pr[Y_{i,t} | T_{N,T}] | \hat{a}_{i,t'} - k + 1:t'] \leq L_3\). This gives us a bound of \(c(t' - t)^{-\gamma} 2^k L_3\). Substituting
As we argued before, we can bound \( E \) proceeding same as lemma D.14 we get, Lemma D.15.

\[
\begin{align*}
&\left(c(t' - t)^{-\gamma}2^k L_3 \sum_{a_{i,t}} \int_{y_{i,t}} \Pr\left[a_{i,t}, y_{i,t} \right] w_{i,t}(a_{i,t}, y_{i,t}) \right| \log \Pr\left[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \right| T_{N,T}] | \log \Pr\left[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \right| T^*_{N,T}] dY_{i,t} \\
&\times \Pr\left[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \right| T_{N,T}] | \log \Pr\left[Y_{i,t} = Y_{i,t}(a_{i,t-k+1:t}) \right| T^*_{N,T}] dY_{i,t} \\
&= \left(c(t' - t)^{-\gamma}2^k L_3 \sum_{a_{i,t}} \int_{y_{i,t}} \Pr\left[y_{i,t} \right| a_{i,t-k+1:t}] \Pr\left[a_{i,t-k+1:t} \right| a_{i,t-2k+1:t-k} \right] dY_{i,t} \\
&\times \Pr\left[y_{i,t} \right| a_{i,t-k+1:t}] \Pr\left[a_{i,t-k+1:t} \right| a_{i,t-2k+1:t-k} \right] \\
&= \left(c(t' - t)^{-\gamma}2^k L_3 \sum_{a_{i,t}} \int_{y_{i,t}} \Pr\left[y_{i,t} \right| a_{i,t-k+1:t}] \Pr\left[a_{i,t-k+1:t} \right| a_{i,t-2k+1:t-k} \right] dY_{i,t} \\
&\times \Pr\left[y_{i,t} \right| a_{i,t-k+1:t}] dY_{i,t} \\
&= \left(c(t' - t)^{-\gamma}2^k L_3 \sum_{a_{i,t}} \Pr[a_{i,t-k+1:t}] E [\left|\log \Pr\left[Y_{i,t} \right| T_{N,T}] \right| a_{i,t-k+1:t}] \right)
\end{align*}
\]

As we argued before, we can bound \( E [\left|\log \Pr\left[Y_{i,t} \right| T_{N,T}] \right| a_{i,t-k+1:t}] \) by \( L_3 \) for each choice of \( a_{i,t-k+1:t} \). This gives an overall bound of \( c(t' - t)^{-\gamma}4^k L_3^2 \) on the covariance. \( \square \)

**Lemma D.15.** Suppose \( t' > t + 2k \) and assumption \textbf{Asn.3} holds. Then the following is true.

\[
|\text{cov}(w_{i,t} \log \Pr\left[Y_{i,t} \right| T_{N,T}, \beta_{N,T}], w_{i,t} \log \Pr\left[Y_{i,t'} \right| T_{N,T}, \beta_{N,T}]|) \leq O \left(2^k(t' - t)^{-\gamma}\right)
\]

**Proof.** Proceeding same as lemma D.14 we get,

\[
|\text{cov}(w_{i,t} \log \Pr\left[Y_{i,t} \right| T_{N,T}, \beta_{N,T}], w_{i,t} \log \Pr\left[Y_{i,t'} \right| T_{N,T}, \beta_{N,T}]|) \\
= \left|\int_{H_{i,t}} \Pr\left[H_{i,t} \right| w_{i}(H_{i,t}) \log \Pr\left[y_{i,t} \right| T_{N,T}, \beta_{N,T}, z_{i}] \right) dY_{i,t} \\
\times \int_{\tilde{H}_{i,t'-2k+1:t'}} \Pr\left[\tilde{H}_{i,t'-2k+1:t'} \right| H_{i,t}] - \Pr\left[\tilde{H}_{i,t'-2k+1:t'} \right] dY_{i,t'-2k+1:t'} \\
\times w_{i,t'}(\tilde{H}_{i,t'-2k+1:t'}) \log \Pr\left[y_{i,t'} \right| T_{N,T}, \beta_{N,T}, z_{i}] \right) dY_{i,t'-2k+1:t'}
\]

195
As before, we simplify $Pr \left[ \tilde{H}_{i,t'} - 2k+1:t' \right] w_{it'}(H_{i,t'} - 2k+1:t')$ as

$$Pr \left[ \tilde{a}_{i,t'} - 2k+1:t' \right] Pr \left[ \tilde{x}_{i,t'} | \tilde{a}_{i,t'} - 2k+1:t', \tilde{x}_{i,t'} - 2k+1:t' - 1 \right] Pr \left[ \tilde{y}_{i,t'} | \tilde{x}_{i,t'}, \tilde{a}_{i,t'} - k+1:t' \right]$$

$$\times Pr \left[ \tilde{x}_{i,t'} - 2k+1:t' - 1 | \tilde{a}_{i,t'} - 2k+1:t' - k \right] Pr \left[ \tilde{y}_{i,t'} - 2k+1:t' - 1 | \tilde{x}_{i,t'} - 2k+1:t' - 1, \tilde{a}_{i,t'} - 2k+1:t' - k \right].$$

We simplify $Pr \left[ \tilde{H}_{i,t'} - 2k+1:t' | H_{i,1:t} \right] w_{it'}(H_{i,t'} - 2k+1:t')$ as

$$Pr \left[ \tilde{a}_{i,t'} - 2k+1:t' | H_{i,1:t} \right] Pr \left[ \tilde{x}_{i,t'} | \tilde{a}_{i,t'} - 2k+1:t', \tilde{x}_{i,t'} - 2k+1:t' - 1, H_{i,1:t} \right] Pr \left[ \tilde{y}_{i,t'} | \tilde{x}_{i,t'}, \tilde{a}_{i,t'} - k+1:t' \right]$$

$$\times Pr \left[ \tilde{x}_{i,t'} - 2k+1:t' - 1 | \tilde{a}_{i,t'} - 2k+1:t' - k \right] Pr \left[ \tilde{y}_{i,t'} - 2k+1:t' - 1 | \tilde{x}_{i,t'} - 2k+1:t' - 1, \tilde{a}_{i,t'} - 2k+1:t' - k \right].$$

Substituting these two results and simplifying we get the following upper bound on covariance.

$$\left| \int_{H_{i,1:t}} Pr \left[ H_{i,1:t} \right] w_{it}(H_{i,1:t}) |\log Pr \left[ y_{it} | T_{N,T}, \beta_{N,T}, z_i \right] | dY_{i,1:t} \right|$$

$$\times \sum_{\tilde{x}_{i,t'} - 2k+1:t', \tilde{a}_{i,t'} - 2k+1:t'} \sum_{\tilde{y}_{i,t'}} Pr \left[ \tilde{a}_{i,t'} - 2k+1:t' \right] Pr \left[ \tilde{y}_{i,t'} | \tilde{x}_{i,t'}, \tilde{a}_{i,t'} - k+1:t' \right] Pr \left[ \tilde{x}_{i,t'} - 2k+1:t' | \tilde{a}_{i,t'} - 2k+1:t' \right]$$

$$\times \left( \frac{Pr \left[ \tilde{a}_{i,t'} - 2k+1:t' | H_{i,1:t} \right] Pr \left[ \tilde{x}_{i,t'} - 2k+1:t' | \tilde{a}_{i,t'} - 2k+1:t', H_{i,1:t} \right]}{Pr \left[ \tilde{x}_{i,t'} - 2k+1:t' | \tilde{a}_{i,t'} - 2k+1:t' \right]} - 1 \right)$$

$$\times |\log Pr \left[ y_{it} | T_{N,T}, \beta_{N,T}, z_i \right] | dY_{i,t'} |$$

Now we use assumption Asn.3 and proceed same as lemma D.14 to get the following bound.

$$c(t' - t)^{-\gamma} L_5 \left| \int_{H_{i,1:t}} Pr \left[ H_{i,1:t} \right] w_{it}(H_{i,1:t}) |\log Pr \left[ y_{it} | T_{N,T}, \beta_{N,T}, z_i \right] | dY_{i,1:t} \right|$$

Now using lemma D.2 we can bound the absolute value above by $2^k L_3$ and this gives the desired result.

**Lemma D.16.** Let $N$ be a $d$-neighborhood of $T_{N,T}^*$ i.e. $N = \{ T : \| T_{N,T}^* - T \| / \sqrt{NT} \leq d \}$. Then for any $T' \notin N$ we have $\ell_{N,T}^*(T_{N,T}^*) > \ell_{N,T}^*(T') + (2\delta)^k d^2$. 

196
Proof. Fix \(a_i, t_{k+1} = p\). Then

\[
\int \log \left( \frac{\Pr[Y_{it} = Y_{it}(p)|T_{N,T}^*]}{\Pr[Y_{it} = Y_{it}(p)|T']} \right) \Pr[Y_{it} = Y_{it}(p)|T_{N,T}^*] \, dY_{it} \\
= -\frac{1}{2} \left\{ (Y_{it} - T_{N,T}^*(i, t, p))^2 - (Y_{it} - T'(i, t, p))^2 \right\} \Pr[Y_{it} = Y_{it}(p)|T_{N,T}^*] \, dY_{it} \\
= \frac{1}{2} \left( (T_{N,T}^*(i, t, p) - T'(i, t, p)) - \frac{1}{2} \left( (T_{N,T}^*(i, t, p))^2 - (T'(i, t, p))^2 \right) \right) \Pr[Y_{it} = Y_{it}(p)|T_{N,T}^*] \, dY_{it} \\
= \frac{1}{2} (T_{N,T}^*(i, t, p) - T'(i, t, p))^2 \\
= \frac{1}{2} \left( (T_{N,T}^*(i, t, p) - T'(i, t, p))^2 \right)
\]

This gives us the following bound on the difference in log-likelihood

\[
\ell_{N,T}^*(T_{N,T}^*) - \ell_{N,T}^*(T') = \frac{1}{NT} \sum_{i,t} 2^k \sum_{p} \Pr[p] \frac{1}{2} (T_{N,T}^*(i, t, p) - T'(i, t, p))^2 \\
\geq \frac{(2\delta)^k}{NT} \|T_{N,T}^* - T'\|^2_2 > (2\delta)^k d^2
\]

Lemma D.17. Let \(\mathcal{N}\) be a \((d_1, d_2)\)-neighborhood of \((T_{N,T}^*, \beta^*)\) i.e. \(\mathcal{N} = \{(T, \beta) : \|T_{N,T} - T\|/\sqrt{NT} \leq d_1, \|\beta^* - \beta\| \leq d_2\}\) and assumption Asn.4 holds. Then for any \((T', \beta') \not\in \mathcal{N}\) we have \(\ell_{N,T}^*(T_{N,T}^*, \beta^*) + \ell_{N,T}^*(T', \beta') > \ell_{N,T}^*(T', \beta') + \min ((2\delta)^k d_1^2, 2^k \ell^2 \mu d_2^2)\).
Proof. Fix $a_{i,t-k+1:t} = p$ and $z_i$. Then

$$
\int \log \left( \frac{\Pr [Y_{it} = Y_{it}(p,z_i) | T_{N,T}^*, \beta^*]}{\Pr [Y_{it} = Y_{it}(p,z_i) | T', \beta']} \right) \Pr [Y_{it} = Y_{it}(p,z_i) | T_{N,T}^*, \beta^*] dY_{it}
$$

$$
= - \int \frac{1}{2} \left\{ (Y_{it} - T_{N,T}^*(i,t,p) - z_i' \beta^*)^2 - (Y_{it} - T'(i,t,p) - z_i' \beta')^2 \right\} \Pr [Y_{it} = Y_{it}(p,z_i) | T_{N,T}^*, \beta^*] dY_{it}
$$

This gives us the following bound on the difference in log-likelihood

$$
\ell_{N,T}^*(T_{N,T}^*) - \ell_{N,T}^*(T') = \frac{1}{NT} \sum_{i,t} \sum_{z_i} \sum_{a_{i,t-k+1:t}} \Pr [z_i] \Pr [a_{i,t-k+1:t}] \times 
$$

$$
\frac{1}{2} \left( (T_{N,T}^*(i,t,p) + z_i' \beta^*) - (T'(i,t,p) + z_i' \beta') \right)^2
$$

$$
= \frac{1}{NT} \sum_{i,t} \sum_{z_i} \sum_{a_{i,t-k+1:t}} \Pr [z_i] \Pr [a_{i,t-k+1:t}] \times 
$$

$$
\frac{1}{2} \left\{ (T_{N,T}^*(i,t,p) - T'(i,t,p))^2 + (\beta^* - \beta') z_i \right\} + 2 |T_{N,T}^*(i,t,p) - T'(i,t,p)| |(\beta^* - \beta') z_i|
$$

$$
\geq \frac{2\delta^k}{NT} \|T_{N,T}^* - T'\|^2 + 2^k \ell^2 \mu \|\beta^* - \beta'\|^2 + \frac{\ell \mu}{NT} \|T_{N,T}^* - T'\|_2 \|\beta^* - \beta'\|_2 \tag{D.6}
$$

The last line uses three bounds and they follow from the following three observations.

1. The first bound is similar as in the proof of lemma D.16.

2. The second bound follows because by Markov’s inequality,

$$
E \left[ \langle z_i, \beta^* - \beta' \rangle \right] \geq c \Pr \left[ \langle z_i, \beta^* - \beta' \rangle \geq \sqrt{c} \right] \geq c \Pr \left[ \|z_i\|_2 \geq \sqrt{c} / \|\beta^* - \beta'\| \right]
$$

Now, substituting $c = \ell^2 \|\beta^* - \beta'\|^2_2$ we get the second bound.

3. By a similar argument, we can first show that $E \left[ \|z_i, \beta^* - \beta' \| \right] \geq \ell \mu \|\beta^* - \beta'\|_2$. Then we
bound the terms involving the tensors as

\[
\frac{2}{NT} \sum_{i,t} \sum_{a_{i,t-k+1:t}} \Pr [a_{i,t-k+1:t}] |T_{N,T}^*(i, t, p) - T'(i, t, p)|
\]

\[
\geq \frac{2^{k+1}}{NT} \sum_{i,t} \max_p |T_{N,T}^*(i, t, p) - T'(i, t, p)| \geq \frac{2}{NT} \sum_{i,t,p} |T_{N,T}^*(i, t, p) - T'(i, t, p)|
\]

\[
= 2\|T_{N,T}^* - T'\|_1 \geq 2\|T_{N,T}^* - T'\|_2
\]

Now, the result immediately follows since the third term in equation D.6 is non-negative. \(\square\)

**Theorem D.1.** Suppose \(\|\hat{T}_{N,T}\|_\infty \leq L\).

- If \(k \leq O\left(\log(1-\delta)/\delta N\right)\), then
  \[
  \frac{1}{NT} \sum_{p=1}^B \sum_{(i,t)\in O_p} w_{i,t} \left(Y_{i,t} - \hat{T}_{N,T}(i, t, p)\right)^2 \leq (1 + \varepsilon)\text{OPT} + O\left(\frac{L^4}{\varepsilon^2 N^2}\right)
  \]
  with probability at least 4/5.

- If assumption 5.9 holds, then
  \[
  \frac{1}{NT} \sum_{p=1}^B \sum_{(i,t)\in O_p} w_{i,t} \left(Y_{i,t} - \hat{T}_{N,T}(i, t, p)\right)^2 \leq (1 + \varepsilon)\text{OPT} + O\left(\frac{L^4}{\varepsilon^2 N^2}\right)
  \]
  with probability at least 4/5.

**Proof.** Lemma D.12 proves that
\[
\Pr \left[\sum_{p=1}^B \sum_{(i,t)\in O_p} w_{i,t} T(i, t, p)^2 \notin \left[\|T\|_W^2 - \varepsilon_1, \|T\|_W^2 + \varepsilon_1\right]\right] \leq O\left(\exp\left(-\frac{2N}{L^4(\frac{1}{\delta^2})^2}\right)\right).
\]
Therefore, with probability at least 1 - \(\exp\left(-\frac{2N}{L^4(\frac{1}{\delta^2})^2}\right)\) we have,

\[
\frac{1}{NT} \sum_{p=1}^B \sum_{(i,t)\in O_p} w_{i,t} \left(Y_{i,t} - \hat{T}_{N,T}(i, t, p)\right)^2
\]

\[
= \frac{1}{NT} \sum_{p=1}^B \sum_{(i,t)\in O_p} w_{i,t} Y_{i,t}^2 - \frac{2}{NT} \sum_{p=1}^B \sum_{(i,t)\in O_p} w_{i,t} Y_{i,t} \hat{T}_{N,T}(i, t, p) + \frac{1}{NT} \sum_{p=1}^B \sum_{(i,t)\in O_p} w_{i,t} \left(\hat{T}_{N,T}(i, t, p)\right)^2
\]

\[
\leq \frac{1}{NT} \left(\|Y_w\|_W^2 + \varepsilon_1\right) - 2 \sum_{i,t,p} W(i, t, p) Y_{w}(i, t, p) \hat{T}_{N,T}(i, t, p) + \frac{1}{NT} \left(\|T\|_W^2 + \varepsilon_1\right)
\]

\[
= \frac{1}{NT} \|Y_w - \hat{T}_{N,T}\|_W^2 + \frac{2\varepsilon_1}{NT}
\]

Now \(\hat{T}_{N,T}\) approximately solves objective function 5.13 and \(\hat{T}_{N,T}\) exactly solves the objective function 5.13. Therefore, with probability at least 9/10 we have \(\|Y_w - \hat{T}_{N,T}\|_W^2 \leq (1 + \varepsilon)\|Y_w - \hat{T}_{N,T}\|_W^2\).
$\hat{T}_{N,T}^2$. Therefore, with probability at least $9/10 - \exp\left(-\frac{2N\epsilon_1^2}{L^4(\log T)^{2/3}}\right)$ we have,

$$
\frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} \left(Y_{i,t} - \hat{T}_{N,T}(i,t,p)\right)^2 \\
\leq \frac{1}{NT} \|Y_w - \hat{T}_{N,T}\|_W^2 + \frac{2\epsilon_1}{NT} \\
\leq \frac{1+\epsilon}{NT} \|Y_w - \hat{T}_{N,T}\|_W^2 + \frac{2\epsilon_1}{NT} \\
\leq \frac{1+\epsilon}{NT} \|Y_w - T_{N,T}^*\|_W^2 + \frac{2\epsilon_1}{NT} \\
= \frac{1+\epsilon}{NT} \left[\|Y_w\|_W^2 - 2 \sum_{i,t,p} W(i,t,p)Y_w(i,t,p)T_{N,T}^*(i,t,p) + \|T_{N,T}^*\|_W^2\right] + \frac{2\epsilon_1}{NT} \\
\leq \frac{1+\epsilon}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w(i,t) \left(Y_{i,t} - T_{N,T}^*(i,t,p)\right)^2 + \frac{4\epsilon_1}{NT}
$$

The last inequality uses lemma D.12. Therefore with probability at least $9/10 - \exp\left(-\frac{2N\epsilon_1^2}{L^4(\log T)^{2/3}}\right)$ we get

$$
\frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} \left(Y_{i,t} - \hat{T}_{N,T}(i,t,p)\right)^2 \\
\leq (1+\epsilon)\text{OPT} + \frac{4\epsilon_1}{NT}
$$

Now substituting $k = 1/8 \log_{6(1-\delta)/\delta} N$ and $\epsilon_1 = O\left(\frac{L^2}{NT^2}\right)$ and for $N$ large enough we get the first result.

Now suppose $N$ is fixed. If assumption 5.9 holds, then using lemma D.13 we get with probability at least $1 - O\left(\frac{L^4(\log T)^{2/3}}{\epsilon_1^2 T/\log T}\right)$,

$$
\frac{1}{NT} \sum_{p=1}^{B} \sum_{(i,t) \in O_p} w_{i,t} \left(Y_{i,t} - \hat{T}_{N,T}(i,t,p)\right)^2 \\
\leq (1+\epsilon)\text{OPT} + \frac{4\epsilon}{NT}
$$

If we substitute $k = 1/8 \log_{6(1-\delta)/\delta} T$ and $\epsilon_1 = O\left(\frac{L^2}{T^{1/3}}\right)$ and for $T$ large enough, we get the second result. \qed