

# Truthful, Transparent and Fair Data Collection Mechanisms

Présentée le 30 septembre 2020

à la Faculté informatique et communications  
Laboratoire d'intelligence artificielle  
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

**Naman GOEL**

Acceptée sur proposition du jury

Prof. K. Aberer, président du jury  
Prof. B. Faltings, directeur de thèse  
Prof. M. Tambe, rapporteur  
Prof. Y. Chen, rapporteuse  
Dr R. West, rapporteur



Ask the right questions,  
and nature will open the doors to her secrets.

— Sir C. V. Raman

To my family...



# Acknowledgements

I thank my advisor Prof Boi Faltings for his invaluable research advice and excellent mentorship over the last 4.5 years. Thank you for believing in me, giving me this wonderful opportunity to work with you and investing so much time and effort in me during all these years!

I thank Prof Karl Aberer (EPFL), Prof Yiling Chen (Harvard), Prof Milind Tambe (Harvard), and Dr Robert West (EPFL) for their time to be part of my thesis committee and their valuable comments. I also thank my colleagues (Adam, Aleksei, Aris, Diego, Fei, Goran, Igor, Ljubomir, and Panayiotis) from the Artificial Intelligence Lab (LIA) for the great company during all these years and for all the constructive comments on my work and presentations. Special thanks to Diego and Aris for excellent collaborations. I thank Sylive Thomet for taking care of all the difficult administrative work during the past 4.5 years in such a smooth manner. I thank Prof Christoph Koch and Prof Giovanni de Micheli for being great EDIC mentors.

This PhD thesis would not have been possible without the encouragement provided by Prof Divyakant Agrawal, Prof Laure Berti-Equille, Prof Sanjay Chawla, Prof Anne-Marie Kermarrec, and Prof K K Shukla.

I thank Mohammad Yaghini (Spring 2017), Olivier Couque (Spring 2017), Hani Boudabous (Fall 2017), Jiayi Gu (Spring 2018), Jean-Thomas Furrer (Spring 2018), Maxime Rutagarama (Spring 2018), Cyril van Schreven (Fall 2018, Spring 2019, Summer 2019), Gauthier Boeshertz (Summer 2019, Fall 2019, Spring 2020), and Alfonso Amayuelas (Spring 2020) for working with me as part of their mandatory/optional projects at EPFL.

I thank my family for their love and support. Mummy, Papa and Shubham - Thank you!

Last but not the least, I thank the Almighty for always being there with me.

*Lausanne, 2020-08-28*

N. G.



# Abstract

An important prerequisite for developing trustworthy artificial intelligence is high quality data. Crowdsourcing has emerged as a popular method of data collection in the past few years. However, there is always a concern about the quality of the data thus collected. This thesis addresses two major challenges in collecting high quality data from a crowd: 1) how to incentivize crowd workers to report accurate data; 2) how to ensure that the data collection mechanism is transparent and fair.

We first propose two novel peer-consistency mechanisms for crowdsourcing: the Deep Bayesian Trust (DBT) mechanism and the Personalized Peer Truth Serum (PPTS). The DBT mechanism incentivizes workers to report accurate answers for objective questions with discrete ground truth answers. It is useful, for example, in collecting labels for supervised machine learning tasks. The mechanism ensures dominant uniform strategy incentive compatibility and fair rewards to the workers. The PPTS incentivizes workers to truthfully report their personal data (for example, body measurements). Since data is personal in nature, the tasks can not be shared between two workers. We show that when individuals report combinations of multiple personal data attributes, the correlation between them can be exploited to find peers and provide guarantees on the incentive compatibility of the mechanism.

We next address the transparency issue of data collection. Smart contracts often rely on a trusted third party (oracle) to get correct information about real-world events. We show how peer-consistency mechanisms can be used to build decentralized, trustless and transparent data oracles on blockchain. We derive conditions under which a peer-consistency incentive mechanism can be used to acquire truthful information from an untrusted and self-interested crowd, even when the crowd has outside incentives to provide wrong information. We also show how to implement the peer-consistency mechanisms in Ethereum. We discuss various non-trivial issues that arise in implementing peer-consistency mechanisms in Ethereum, suggest several optimizations to reduce gas cost and provide empirical analysis.

Finally, we address the problem of fair data collection from a crowd. Sharing economy platforms such as Airbnb and Uber face a major challenge in the form of peer-to-peer discrimination based on sensitive personal attributes such as race and gender. We show that how a peer-consistency incentive mechanism can be used to encourage users to go against common bias and provide a truthful rating about others, obtained through a more careful and deeper evaluation. In situations where an incentive mechanism can't be implemented, we show that a simple post-processing approach can also be used to correct bias in the reputation scores, while minimizing loss in the useful information provided by the scores. We also address

## **Abstract**

---

the problem of fair and diverse data collection from a crowd under budget constraints. We propose a novel algorithm which maximizes the expected accuracy of the collected data, while ensuring that the errors satisfy desired notions of fairness w.r.t sensitive attributes.

**Key Words:** Data Collection, Crowdsourcing, Incentive schemes, Game theory, Mechanism design, Peer prediction, Fairness, Blockchain, Decentralization, Transparency



# Résumé

Une condition préalable importante pour développer une intelligence artificielle fiable est la qualité des données. Le crowdsourcing est devenu une méthode populaire de collecte de données au cours des dernières années. Cependant, la qualité des données ainsi collectées est toujours préoccupante. Cette thèse aborde deux défis majeurs dans la collecte de données de haute qualité auprès d'une foule : 1) comment inciter les travailleurs de la foule à communiquer des données précises ; 2) comment garantir que le mécanisme de collecte de données est transparent et équitable.

Nous proposons d'abord deux nouveaux mécanismes de peer-consistency pour le crowdsourcing : le mécanisme Deep Bayesian Trust (DBT) et le Personalized Peer Truth Serum (PPTS). Le mécanisme DBT incite les travailleurs à signaler des réponses précises à des questions objectives avec des réponses discrètes. Il est utile, par exemple, pour collecter des étiquettes pour les tâches d'apprentissage automatique supervisé. Le mécanisme garantit une compatibilité incitative dominante de la stratégie uniforme et des récompenses équitables aux travailleurs. Le PPTS incite les travailleurs à communiquer fidèlement leurs données personnelles (par exemple, les mensurations). Les données étant de nature personnelle, les tâches ne peuvent pas être partagées entre deux travailleurs. Nous montrons que lorsque des individus signalent des combinaisons de plusieurs attributs de données personnelles, la corrélation entre eux peut être exploitée pour trouver des peers et fournir des garanties sur la compatibilité incitative du mécanisme.

Nous abordons ensuite la question de la transparence de la collecte de données. Les contrats intelligents s'appuient souvent sur un tiers de confiance (oracle) pour obtenir des informations correctes sur les événements du monde réel. Nous montrons comment les mécanismes de peer-consistency peuvent être utilisés pour construire des oracles de données décentralisés, sans confiance et transparents sur la blockchain. Nous dérivons des conditions dans lesquelles un mécanisme d'incitation à la peer-consistency peut être utilisé pour acquérir des informations véridiques auprès d'une foule non fiable et qui cherche son avantage personnel, même lorsque la foule a des incitations extérieures à fournir des informations erronées. Nous montrons aussi comment mettre en œuvre les mécanismes de cohérence par les pairs dans Ethereum. Nous discutons de divers problèmes non triviaux qui surviennent dans la mise en œuvre de mécanismes de cohérence entre pairs dans Ethereum, suggérons plusieurs optimisations pour réduire le coût du gaz et fournissons une analyse empirique.

Enfin, nous abordons le problème de la collecte équitable de données auprès d'une foule. Le partage des plateformes économiques telles qu'Airbnb et Uber est confronté à un défi ma-

## Résumé

---

jeu sous la forme de discrimination entre pairs basée sur des attributs personnels sensibles tels que la race et le sexe. Nous montrons comment un mécanisme d'incitation à la peer-consistency peut être utilisé pour encourager les utilisateurs à aller à l'encontre des préjugés courants et à fournir une évaluation véridique des autres, obtenue grâce à une évaluation plus minutieuse et plus approfondie. Dans les situations où un mécanisme d'incitation ne peut pas être mis en œuvre, nous montrons qu'une approche de post-traitement simple peut également être utilisée pour corriger le biais dans les scores de réputation, tout en minimisant la perte des informations utiles fournies par les scores. Nous abordons également le problème de la collecte de données équitable et diversifiée auprès d'une foule soumise à des contraintes budgétaires. Nous proposons un nouvel algorithme qui maximise la précision attendue des données collectées, tout en garantissant que les erreurs satisfont aux notions souhaitées d'équité par rapport à attributs sensibles.

Mots clés : collecte de données, crowdsourcing, systèmes d'incitation, théorie des jeux, conception de mécanismes, peer-prediction, équité, blockchain, décentralisation, transparence

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract (English/Français)</b>	<b>iii</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Challenges . . . . .	1
1.2 Summary of Main Contributions . . . . .	6
1.3 Thesis Organization . . . . .	7
<b>2 Deep Bayesian Trust: A Dominant and Fair Incentive Mechanism for Crowd</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.1.1 Related Work . . . . .	10
2.1.2 Our Contributions . . . . .	12
2.2 Model . . . . .	12
2.3 Finding Trustworthiness Transitively . . . . .	14
2.4 The Deep Bayesian Trust Mechanism . . . . .	15
2.5 Theoretical Analysis . . . . .	19
2.5.1 Fairness of Rewards . . . . .	20
2.6 Numerical Simulations . . . . .	21
2.7 Preliminary User Study . . . . .	23
2.8 Chapter Summary . . . . .	26
<b>3 Personalized Peer Truth Serum for Eliciting Multi-Attribute Personal Data</b>	<b>27</b>
3.1 Introduction . . . . .	27
3.1.1 Related Work . . . . .	28
3.1.2 Our Contributions . . . . .	30
3.2 Settings . . . . .	31
3.2.1 Belief Model . . . . .	31
3.3 The PPTS Mechanism . . . . .	32
3.4 Theoretical Analysis . . . . .	33
3.5 Clusters Approximation . . . . .	35
	vii

## Contents

---

3.6	Nearest Neighbors Scheme . . . . .	36
3.7	Experimental Evaluation . . . . .	38
3.7.1	Datasets . . . . .	38
3.7.2	Cluster Fitness Evaluation . . . . .	39
3.7.3	Attribute Score . . . . .	39
3.7.4	Cumulative Reward . . . . .	41
3.8	Application to Data Cleaning . . . . .	42
3.9	Chapter Summary . . . . .	46
<b>4</b>	<b>Infochain: A Decentralized, Trustless and Transparent Oracle on Blockchain</b>	<b>47</b>
4.1	Introduction . . . . .	47
4.1.1	Related Work . . . . .	49
4.1.2	Our Contributions . . . . .	50
4.2	Model and Objectives . . . . .	52
4.3	Truthful Equilibrium and Savings . . . . .	56
4.4	Honest Agents . . . . .	58
4.5	Simulations . . . . .	60
4.5.1	Dataset . . . . .	60
4.5.2	Simulation Parameters . . . . .	60
4.5.3	Experimental Results . . . . .	61
4.6	Infochain . . . . .	62
4.6.1	Commit-Reveal Protocol . . . . .	63
4.6.2	Cost Optimizations . . . . .	64
4.6.3	Random Peer Selection . . . . .	65
4.6.4	Negative Payments . . . . .	65
4.7	Experiments . . . . .	65
4.7.1	Dataset . . . . .	65
4.7.2	Results . . . . .	66
4.8	Chapter Summary . . . . .	67
<b>5</b>	<b>Tackling Peer-to-Peer Discrimination in the Sharing Economy</b>	<b>69</b>
5.1	Introduction and Related Work . . . . .	69
5.1.1	Our Contributions . . . . .	70
5.2	Airbnb Case Study . . . . .	71
5.2.1	Dataset . . . . .	71
5.2.2	Data Analysis . . . . .	72
5.3	Bias Free Rating Elicitation . . . . .	74
5.3.1	Preliminaries and Notation . . . . .	75
5.3.2	The Peer Truth Serum for Crowdsourcing [97] . . . . .	75
5.3.3	Belief Update Assumption . . . . .	76
5.3.4	Game-Theoretic Properties . . . . .	76
5.3.5	Sensitive Attribute Information . . . . .	77
5.4	Bias Correction . . . . .	78

5.4.1	Post-Aggregation Transformation . . . . .	79
5.4.2	Range Scaling . . . . .	80
5.4.3	Experiments . . . . .	80
5.5	Chapter Summary . . . . .	82
<b>6</b>	<b>Crowdsourcing with Fairness, Diversity and Budget Constraints</b>	<b>85</b>
6.1	Introduction . . . . .	85
6.1.1	Related Work . . . . .	86
6.1.2	Our Contributions . . . . .	87
6.2	Model . . . . .	88
6.3	Finding Optimal Crowdsourcing Policy . . . . .	90
6.3.1	Estimates of Worker Accuracy Matrices . . . . .	91
6.4	Theoretical Analysis . . . . .	92
6.5	Experimental Evaluation . . . . .	93
6.5.1	Datasets . . . . .	93
6.5.2	Baselines . . . . .	95
6.5.3	Observations . . . . .	95
6.6	Chapter Summary . . . . .	98
<b>7</b>	<b>Concluding Remarks</b>	<b>99</b>
<b>A</b>	<b>Appendix</b>	<b>103</b>
A.1	Missing Proofs for Chapter 2 . . . . .	103
A.1.1	Proof of Lemma 1 . . . . .	103
A.1.2	Proof of Theorem 1 . . . . .	104
A.1.3	Proof of Theorem 2 . . . . .	105
A.1.4	Proof of Theorem 3 . . . . .	105
A.2	Missing Proofs for Chapter 3 . . . . .	106
A.2.1	Proof of Theorem 4 . . . . .	106
A.2.2	Proof of Theorem 5 . . . . .	107
A.2.3	Proof of Theorem 6 . . . . .	108
A.2.4	Proof of Theorem 7 . . . . .	108
A.2.5	Proof of Theorem 8 . . . . .	109
A.3	Missing Proofs for Chapter 4 . . . . .	111
A.3.1	Proof of Proposition 2 . . . . .	111
A.3.2	Proof of Theorem 9 . . . . .	111
A.3.3	Proof of Theorem 10 . . . . .	113
A.3.4	Proof of Proposition 3 . . . . .	114
A.3.5	Proof of Lemma 2 . . . . .	115
A.3.6	Proof of Theorem 11 . . . . .	116
A.3.7	Proof of Theorem 12 . . . . .	118
A.4	Missing Proofs for Chapter 6 . . . . .	120

**Contents**

---

A.4.1 Proof of Theorem 13 . . . . . 120

A.4.2 Proof of Theorem 14 . . . . . 122

**Bibliography** . . . . . **133**

**Curriculum Vitae** . . . . . **135**

# List of Figures

2.1	Model . . . . .	14
2.2	Illustration of the Deep Bayesian Trust Mechanism . . . . .	17
2.3	$\beta$ distributed proficiencies . . . . .	22
2.4	Uniformly distributed proficiencies . . . . .	23
2.5	Mechanical Turk Task . . . . .	24
2.6	Time Spent on HITs . . . . .	25
3.1	Strategic agents in crowdsourcing . . . . .	28
3.2	Belief Model - Clustering . . . . .	32
3.3	PPTS-knn . . . . .	37
3.4	Cluster Distribution in Datasets . . . . .	38
3.5	Statistics of Attribute Scores . . . . .	40
3.6	Statistics of Cumulative Rewards (Average) . . . . .	40
3.7	Statistics of Cumulative Rewards (Median) . . . . .	40
3.8	Statistics of Cumulative Rewards (Median, knn) . . . . .	42
3.9	Performance comparison of data cleaning approaches (synthetic data) . . . . .	43
3.10	Performance comparison of data cleaning approaches (real data) . . . . .	44
3.11	Performance comparison of data cleaning approaches(real data, knn) . . . . .	45
4.1	A Motivating Example of Outcome Dependent Lying Incentives . . . . .	49
4.2	Relative saving made by PTSC. . . . .	61
4.3	Infochain Overview . . . . .	62
4.4	Experimental Results . . . . .	66
5.1	Controlling $\delta$ . . . . .	82
5.2	Running Time . . . . .	82
6.1	Varying $N_g$ (Number of gold tasks), Settings : Uniform Costs, $\beta = 0.01, \alpha = 0.01$ . . . . .	96
6.2	Varying $\alpha$ (Fairness Constraint), Settings : Uniform Costs, $\beta = 0.01, N_g = 20$ . . . . .	96
6.3	Varying $N_g$ (Number of gold tasks), Settings : Non-Uniform Costs, $\beta = 0.01, \alpha = 0.01, C = 1.5$ . . . . .	96
6.4	Varying $\alpha$ (Fairness Constraint), Settings : Non-Uniform Costs, $\beta = 0.01, N_g = 20, C = 1.5$ . . . . .	97

## List of Figures

---

6.5	Varying $N_g$ (Number of gold tasks), Settings : Non-Uniform Costs, $\beta = 0.01, \alpha = 0.01, C = 2.5$ . . . . .	97
6.6	Varying $\alpha$ (Fairness Constraint), Settings : Non-Uniform Costs, $\beta = 0.01, N_g = 20, C = 2.5$ . . . . .	97
6.7	Varying $N_g$ (Number of gold tasks), Settings : Non-Uniform Costs, $\beta = 0.005, \alpha = 0.01, C = 2.5$ . . . . .	98
6.8	Varying $\alpha$ (Fairness Constraint), Settings : Non-Uniform Costs, $\beta = 0.005, N_g = 20, C = 2.5$ . . . . .	98



## List of Tables

3.1	Average CMI estimates for different datasets . . . . .	38
5.1	Ratings and prices for different host ethnicity . . . . .	72
5.2	Ratings and prices for different neighborhoods . . . . .	73
5.3	Regression Analysis for Price . . . . .	74
5.4	Regression Analysis for Ratings . . . . .	74
5.5	MSE and Covariance after the transformation . . . . .	81
5.6	Regression Analysis for Transformed Ratings . . . . .	82
5.7	Synthetic Data Results (Power Law Distribution) . . . . .	82



# 1 Introduction

## 1.1 Motivation and Challenges

It is difficult to find any walk of life in 2020 that has not been influenced and changed by artificial intelligence (AI). Education, commerce, healthcare, entertainment, governance, food security, housing, employment, insurance, banking, travel, intelligence gathering, criminal justice, etc are just a few examples from our everyday lives that have witnessed a growing influence of AI in the past few years. While the impact of AI on our society is largely positive, it is not uncommon to find controversies and concerns about this technology. Thus, making AI “trustworthy” has now become one of the most important and high priority goals in the AI research community. The Oxford dictionary defines the term “trustworthy” as something **“that you can rely on to be good, honest, sincere, etc”**. Clearly, it is a crucial characteristic for wider social acceptance of the technology in the long term. Governments and organizations across the world have recently been announcing regulations and guidelines for trustworthy AI. For example, the European Commission recommends [87] that the development, deployment and use of AI systems meet the seven key requirements for trustworthy AI:

1. Human agency and oversight
2. Technical robustness and safety
3. Privacy and data governance
4. Transparency
5. Diversity
6. Non-discrimination and fairness
7. Environmental and societal well-being
8. Accountability

## Chapter 1. Introduction

---

Similarly, the White House lists the following [1] among its principles for the stewardship of AI applications:

1. Public trust in AI
2. Public participation
3. Information quality
4. Fairness and non-discrimination
5. Disclosure and transparency

The Australian government has recently released a similar list as part of Australia's AI ethics framework [24]. Private organizations like Microsoft and IBM have come up with their own sets of principles. For example, Microsoft lists the following as its responsible AI principles [81]:

1. Fairness
2. Reliability and safety
3. Privacy and security
4. Inclusiveness
5. Transparency
6. Accountability

Similarly, IBM identifies the following as five areas of ethical focus [51]:

1. Accountability
2. Value alignment
3. Explainability
4. Fairness
5. User data rights

While there are several principles (like fairness, explainability etc) that are common across these lists, there are also some other equally important principles in the above lists (like information quality, public participation, transparency, user data rights etc) that don't receive as much attention.

Given the fact that trustworthiness requires a number of non-trivial properties to be satisfied, it is obvious that making AI trustworthy is a highly ambitious goal. It is not even clear what many of these properties mean precisely. For example, both fairness and explainability may have different meanings in different contexts and applications. Further, many of these properties present novel trade-offs with the traditional notions of the “accuracy” of AI systems.

Modern AI systems are mostly data-driven and thus, the trustworthiness of these systems is inherently tied to the data collection mechanisms. Data is a primary requirement for machine learning algorithms and the algorithms are only as reliable/trustworthy as the data used for training them. This doesn't imply that having trustworthy data will automatically solve all other problems related to trustworthiness in AI and machine learning, but having trustworthy data is certainly one of the most important requirements for trustworthy AI. There are three important questions that arise immediately in this context:

- Q1.** How to get highest quality data for AI?
- Q2.** How to ensure that the data collection process is transparent and fair for the data providers?
- Q3.** How to ensure that the collected data is fair for the users affected by the applications that make use of this data?

Let's first discuss **Q1** and **Q2** in more detail. Obtaining high quality data while also ensuring that the data collection process is transparent and fair, is a very challenging task. Ideally, the owners of data should have complete control over what data they choose to provide (if any) and they should receive a fair compensation for their contribution. It is not difficult to see that this idea opens up a lot of problems related to the quality of the data thus obtained. For example, consider the data about ratings of products on e-commerce websites like Amazon. It is well known that online product reviews have a J-shaped distribution. However, as shown by [50], the reviews actually follow a roughly normal distribution under controlled settings. This discrepancy between the true data distribution and the observed data distribution is also caused by the misalignment of reviewers' incentives, among other reasons. For example, unlike reviewers with polarized reviews, most reviewers with moderate reviews don't have incentives to express their opinions. This is true not just for online reviews but for most of the other types of the data as well. For example, in healthcare, healthy people usually don't have incentives to report their data. Moreover, when there is a costly effort involved in obtaining correct data, the data providers may even have incentives to provide non-truthful or incorrect data. Clearly, we need incentive schemes to obtain correct and unbiased sample of the data from the data providers. A naive incentive scheme that comes to mind is a constant payment scheme. Unfortunately, this incentive scheme encourages free-riding but doesn't fulfill our objective of obtaining trustworthy data. For the incentive scheme to work, the payments have to be contingent on the correctness of the data. This is hard problem to solve if there is no way to verify the correctness of the data. Peer-consistency mechanisms [30] address this

problem by using the data reported by multiple “peers”. Consider a simple peer-consistency mechanism, the output-agreement mechanism, which rewards the data providers with, say, \$1 if there is an agreement between the data provided by two peers and \$0 otherwise. While this simple mechanism provides very weak guarantees for practical applications, it motivates the fundamental idea behind more advanced peer-consistency mechanisms like [20, 97, 104]. All these advanced mechanisms provide very strong game-theoretic guarantees that make them suitable for practical applications. However, the mechanisms also have the following limitations:

1. The existing peer-consistency mechanisms make truthful reporting the highest paid Nash equilibria. But they don’t make truthful reporting the dominant strategy, which is a more robust and desired notion of incentive compatibility in game theory.
2. The existing mechanisms can be perceived as being unfair in the sense that the reward of a data provider is also dependent on the accuracy and the strategy of her peer. This implies that two honest data providers who have similar accuracy are not guaranteed to get similar expected rewards because one of them may get matched with a good peer and the other may get matched with a bad peer. This violates the individual fairness [27] notion for algorithms. In fact, a data provider with higher accuracy can get lower expected reward than a data provider with lower accuracy due to the same reason.
3. The existing mechanisms work only for non-personal data, for example, opinions or measurements about places, objects or services that can be shared between different people. But they don’t work for personal data, for example data about one’s own body or other personal properties like smart homes, which obviously can’t be shared between two independent people.
4. Even though the mechanisms themselves are decentralized in the sense that the data is collected from multiple independent data providers, they are always thought to require a centralized system to be implemented in practice. A center is responsible for collecting data from multiple providers, performing reward calculation and paying out the corresponding rewards. This unnecessarily limits the transparency and decentralized nature of the data collection process.
5. In many cases, data providers have outside incentives to lie about their data. This is specially true when there is an outcome to be determined based on the collected data and the data providers are eligible for an outside monetary reward or compensation based on the determined outcome. The existing literature doesn’t answer whether peer-consistency mechanisms can still be used in such cases to get truthful data, and if yes, under what conditions.

Let’s now discuss **Q3** i.e. how to ensure that the collected data is fair for the users affected by the applications that make use of this data. There are three components of this question: data,

algorithm and application. All three of these components are equally important in ensuring end-to-end fairness. We mostly focus on the data component; discussion on fair algorithms and “ethical” applications is beyond the scope of this thesis <sup>1</sup>.

Sharing economy platforms like Airbnb, Uber, and freelancer platforms like Taskrabbit, Fiverr, etc are increasingly becoming popular and often dominate the traditional offline systems in the respective domains. While these platforms are considered more accessible and convenient, they also face some major ethical challenges. For example, studies have shown that the prices of properties offered by African-American hosts are significantly lower than those offered by their white counterparts [29] even while accounting for other observable features about the properties. As a more direct form of peer-to-peer discrimination, a recent study [28] has found that having an African-American name significantly reduces the likelihood of a guest’s accommodation request being accepted on Airbnb. A similar study on Uber [34] has shown that having an African-American name increases the likelihood of a passenger’s ride being canceled by the drivers. It is deeply concerning that the existing social biases are finding their way into web-based platforms too. A combination of biased human feedback and large scale algorithmic decision-making on the web can cause further social segregation of historically disadvantaged groups. Thus, the problem needs urgent attention and solution. Airbnb (with Stanford) recently conducted a user study [2] on Airbnb users and claimed that reputation systems offset the real-world social biases by building trust between different users. However, for this to work, the reputation systems must themselves be non-discriminatory. A reputation system that discriminates against people based on race or gender will only further reinforce the bias. Unfortunately, the reputation systems on these platforms are often discriminatory towards different races and genders [41]. It is a non-trivial problem to make a reputation system non-discriminatory because the goal of the reputation systems is really to discriminate between users. However, this discrimination must be based on relevant attributes and not on sensitive attributes like race or gender. Thus, the challenge is to make reputation systems racially non-discriminatory while retaining the other useful information they provide. It is interesting in the context of this thesis because reputation systems are based on aggregating the ratings provided by users (humans) to one another, and thus, it directly fits within the broader question of collecting trustworthy and fair data. We are interested in interventions that can ensure fairness on the platforms but without manipulating any information or making it easy for anyone to get opportunities. This subtle relation between fairness and truthfulness of information has not been explored in the literature. Another very recent study [26] conducted on Amazon Mechanical Turk showed that the crowdworkers are racially biased while predicting recidivism labels for defendants. The difference in false positive rates of crowd predictions for white and black defendants was significant and nearly equal to that of the predictions made by the controversial COMPAS algorithm [85]. The same was true for false negative rates also. Crowdsourcing is increasingly used to collect training data labels. More studies have shown other forms of bias in the data collected from crowd [88, 89]. Inevitably, crowdworkers have

---

<sup>1</sup>We have worked on fairness in algorithms during the PhD but the results are excluded from the thesis for the sake of coherence. Readers interested in our results about algorithmic fairness are encouraged to check [39].

different biases, which are then reflected in the labels collected from the workers. This raises further interesting challenges about collecting fair data from the crowd.

## 1.2 Summary of Main Contributions

### Novel Incentive Mechanisms

We present the Deep Bayesian Trust, a novel game-theoretic mechanism that uses a limited amount of ground truth and peer-consistency to incentivize rational agents to invest effort and honestly report the information. This mechanism is useful, for example, in collecting high quality labels for supervised machine learning. The mechanism ensures a strong notion of incentive compatibility (dominant uniform strategy) and solves several scalability and game-theoretic issues with the prior mechanisms that employed only ground truth or only peer-consistency. Besides, this is the first (and perhaps the only) peer-based mechanism to also guarantee individual fairness in addition to incentive compatibility. The mechanism ensures that any two agents with the same accuracy are not differentiated by the mechanism and receive the same reward.

To elicit personal attributes (for e.g. health and activity data) of people, we present the Personalized Peer Truth Serum (PPTS). Since attributes are personal in nature, different people can't observe the same data. We show for the first time how to extend peer-consistency incentive mechanisms to this setting for collecting continuous valued personal attributes. The main idea is to use other correlated attributes reported by the people to find peers. We also show on several real datasets that the scores calculated by the PPTS mechanism are also helpful in filtering out non-truthful data points and thus, the mechanism can potentially be used as a post-processing algorithm too for further improving the quality of collected data.

### Transparent and Decentralized Oracle

Blockchain based systems allow various kinds of financial transactions to be executed in a decentralized manner. However, these systems often rely on a trusted third party (oracle) to get correct information about the real-world events, which trigger the financial transactions. In this thesis, we identify two biggest challenges in building decentralized, *trustless*<sup>2</sup> and transparent oracles. The first challenge is acquiring correct information about the real-world events without relying on a trusted information provider. We show how a peer-consistency incentive mechanism can be used to acquire truthful information from an untrusted and self-interested crowd, even when the crowd has outside (outcome induced) incentives to provide wrong information. More precisely, we derive conditions under which a peer-consistency mechanism can be used to elicit truthful data from non-trusted rational agents when an aggregate statistic of the collected data affects the amount of their incentives to lie. Furthermore, we discuss the relative saving that can be achieved by the mechanism, compared to

---

<sup>2</sup>Trustless is a term increasingly used in the context of decentralized and blockchain systems, meaning not requiring trust.



the rational outcome, if no such mechanism was implemented. The second challenge is a system design and implementation challenge. For the first time, we show how to implement a trustless and transparent oracle in Ethereum. We discuss various non-trivial issues that arise in implementing peer-consistency mechanisms in Ethereum, suggest several optimizations to reduce gas cost and provide empirical analysis.

### **Fairness on Sharing Economy Platforms**

To address the problem of discrimination on sharing economy platforms, we show how a game-theoretical incentive mechanism can be used to encourage users to go against common bias and provide truthful ratings about others, obtained through a more careful and deeper evaluation. In situations where an incentive mechanism can't be implemented, we show that a simple post-processing approach can also be used to correct bias in the reputation scores, while minimizing the loss in the useful information provided by the scores. We evaluate the proposed solution on synthetic and real datasets from Airbnb.

### **Fairness and Diversity in Budget Constrained Crowdsourcing**

Even when people behave honestly, they may make errors in judgments. We propose a novel algorithm which assigns tasks to crowdworkers to maximize the expected accuracy of the crowdsourced data, while ensuring that the errors satisfy desired notions of fairness. We provide guarantees on the performance of our algorithm and show that the algorithm performs well in practice through experiments on a real dataset.

## **1.3 Thesis Organization**

The thesis is organized as follows:

- In Chapter 2, we discuss the game theoretic framework for information elicitation tasks with objective answers and present the Deep Bayesian Trust mechanism. We discuss the theoretical guarantees of the mechanism, followed by the results of some numerical simulations and a preliminary user study on Mechanical Turk.
- In Chapter 3, we discuss the game theoretic framework for information elicitation tasks involving personal data and present the Personalized Peer Truth Serum. We discuss the theoretical guarantees of the mechanism, followed by experimental evaluation on real datasets.
- In Chapter 4, we discuss the issue of outcome dependent lying incentives, the suitability of peer-consistency mechanisms for designing decentralized, trustless and transparent oracles on blockchain, followed by empirical analysis about the gas costs incurred due to the computations involved in different mechanisms.
- In Chapter 5, we address the issue of bias and discrimination on sharing economy

platforms like Airbnb and Uber by proposing two different approaches (a game theoretic incentive scheme and a post-aggregation bias correction technique), and discuss the results on a real dataset from Airbnb.

- In Chapter 6, we discuss the novel task allocation algorithm for crowdsourcing under fairness, diversity and budget constraints, followed by the results based on a recidivism dataset from Broward County.

## 2 Deep Bayesian Trust: A Dominant and Fair Incentive Mechanism for Crowd

This chapter is an extended version of the following publication:

N Goel, B Faltings. Deep Bayesian Trust: A Dominant and Fair Incentive Mechanism for Crowd. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2019.

### 2.1 Introduction

Crowdsourcing is a popular method for collecting data but the collected data is often noisy and of low quality. The quality gets significantly degraded if solving the tasks require some costly effort. The problem can be addressed by rewarding the workers with a performance based bonus. One way to reward the workers is to use the peer based mechanisms [20, 97]. In these mechanisms, the reward of a worker depends on her own answers and of other workers. The mechanisms admit honesty as an equilibrium strategy (i.e. if other workers do high quality work, the best response for any worker is also to do the same). But the mechanisms also admit other dishonest equilibria. The other way to reward the workers is to use gold standard tasks [86]. A common technique is to randomly mix some gold tasks in the batch of tasks solved by *every* worker. Workers are then rewarded based on their performance on the gold tasks. This incentivizes the workers for performing high quality work as a dominant strategy (i.e. regardless of other workers' answers). However, since every worker solves gold tasks and the requester already knows the correct answers of the gold tasks, it leads to a waste of the useful task budget of the requester. Moreover, to reduce the variance in the rewards, one may require a sufficient number of gold tasks to be solved by every worker. Also, this technique only works if the gold tasks are not identified but as shown by [15], the gold tasks can be easily leaked because of their repeated use for all workers. Another technique is to assign gold tasks to a worker only with a small (constant) probability and give a fixed reward (independent of the quality of work) otherwise. While it solves many of the problems with the first technique, the scalability of the constant probability technique remains limited because the number of workers who need to be assigned gold tasks, grows as the total number of workers grows.

Otherwise, this technique assumes that the rewards of the workers solving gold tasks can be made arbitrarily large to compensate for the smaller probability<sup>1</sup>. In many practical settings, this is undesired or not possible.

In this chapter, we introduce a scalable incentive mechanism (the ***Deep Bayesian Trust Mechanism***) which guarantees strong incentive compatibility while assigning gold tasks only to a small (constant) number of workers and rewarding the rest using the answers provided by peer workers. Since we ensure that gold tasks are assigned to a constant *number* of workers and not just with a constant *probability*, our mechanism is also suitable for very large scale settings. Moreover, this mechanism doesn't suffer from the problem of arbitrarily large payments discussed earlier. When there is a non-zero cost of effort for solving the tasks, the mechanism still ensures the desired theoretical properties if the payments are scaled appropriately only to cover the cost of effort. This scaling constant doesn't depend on the probability of assigning gold tasks. The mechanism is based on the observation that in large scale crowdsourcing settings, every worker reports answers to many similar tasks and hence the joint distribution of the answers of any two workers can be used to infer the accuracy of one worker given the accuracy of the other worker. It starts by rewarding a small set of workers based on gold tasks and then uses the answers provided by the workers on non-gold tasks as *contributed* gold tasks to reward more workers. It continues this *deep* chain of trust to an arbitrary depth, until all the tasks have been solved by the required number of workers.

As fairness of algorithms affecting humans is becoming a critical issue, it is important to justify the fairness of algorithms that determine payments of the workers. We, for the first time, address the issue of fairness of crowdsourcing incentive mechanisms in a principled manner and show that our mechanism ensures fair rewards to the workers.

### 2.1.1 Related Work

The research on crowdsourcing incentive mechanisms is mainly divided into two categories. The first category of work assumes that some spot checking option (for example, gold standard tasks) is available. The constant probability mechanism, discussed in Section 3.1, is analyzed formally by [33]. This mechanism randomly selects a few workers and spot checks only those workers with an oracle. The rest of the workers are given a constant amount of reward (independent of the quality of their work). The scalability of this mechanism is limited because the number of workers who need to be spot checked, grows as the total number of workers grows. Otherwise, in order to compensate for the smaller probability of spot checking, the mechanism allows the payments of the spot checked workers to be arbitrarily large.

The second category of work assumes no gold tasks to be available and uses only peer answers. Such mechanisms are called the peer-consistency (or peer-prediction) mechanisms. The

---

<sup>1</sup>This can be understood using the example of penalty mechanism in public transport systems. If tickets are checked very rarely, then the penalty for being found without ticket has to be made very high to discourage rational people from traveling without tickets.

early mechanisms in this category were either not detail-free (required knowledge about the beliefs of the workers) [82] or not minimal (required workers to also submit some additional information other than their answers on the tasks) [93, 95, 114]. On the other hand, a simple output-agreement mechanism [110] works only under strong assumptions on the correlation structure of workers' observations. A seminal work in the category of minimal, detail-free mechanisms for crowdsourcing is [20], which ensures that truth-telling is a focal equilibrium in binary answer spaces. The Correlated Agreement mechanism [104] generalizes the mechanism of [20] to non-binary answer spaces with moderate assumptions on the correlation structure of workers' observations. Both these mechanisms require that workers solve multiple tasks. The Logarithmic Peer Truth Serum [96], which is based on an information theoretic principle, requires no such assumptions and ensures strong-truthfulness in non-binary answer spaces. The guarantees of the mechanism are ensured in the limit (when every task is solved by an infinite number of workers). The Peer Truth Serum (PTSC) of [97] doesn't require even this assumption for the theoretical guarantees and works with a bounded number of tasks overall. In theory, these peer-consistency mechanisms offer comparatively weaker incentive compatibility than the gold tasks based mechanisms. They make truth-telling an equilibrium strategy for the workers but also admit some non-truthful equilibria. While the *no-effort* or the *heuristic equilibria* exist in these mechanisms, the equilibria are not attractive since they pay zero reward to the workers. The mechanisms also admit the *permutation equilibria*, which give the same payoff as the truthful equilibrium. [77] avoid this issue in binary answer space by using ground truth of the answer statistics. As shown by [33], it is possible to eliminate the undesired equilibria in the peer based mechanisms if the center can employ a limited amount of spot checking. When spot checking is not possible, it is enough that there exist a small fraction of honest workers. Either of these options work if the rewards are scaled appropriately to compensate for a low probability of spot checking and a low fraction of honest agents respectively.

Finally, the mechanism of [22] combines ideas from the two categories of work. It arranges the workers in a hierarchy. A constant number of workers in the top level of the hierarchy are evaluated by an oracle. The workers below that level are evaluated by the workers (peers) in one level above them. The mechanism solves the scalability issue of the gold tasks based mechanisms. Though it offers comparatively weaker incentive compatibility (unique Nash equilibrium) as compared to the gold tasks based mechanisms but eliminates all the undesired equilibria that exist in peer based mechanisms. However, it requires that workers are informed of their level in the hierarchy. The mechanism is also ex-ante unfair towards the workers in the sense that workers in the top level of hierarchy are evaluated more correctly than the workers in the lower levels. Similar to [22], our work is also at the intersection of the two categories of works. However, our mechanism doesn't suffer from the issues (level information requirement and unfairness) that their mechanism has and also guarantees stronger incentive compatibility.

### 2.1.2 Our Contributions

The summary of our main contributions is as follows:

- We propose a dominant uniform strategy incentive compatible (DUSIC) mechanism, called the **Deep Bayesian Trust Mechanism**, which rewards a constant number of workers with gold tasks and the rest using peer answers.
- On one hand, our mechanism addresses the issues with existing gold tasks based mechanisms and on the other hand, it also shows how the limitations of purely peer based incentive mechanisms can be overcome in some cases by assigning gold tasks to a few workers. Thus, it is also of interest for the peer-prediction community.
- We define a notion of fairness of rewards in crowdsourcing and show that our mechanism ensures fairness.
- Through numerical experiments, we show the robustness of our mechanism under various reporting strategies of the workers. In a preliminary study conducted on Amazon Mechanical Turk, we observe that the mechanism helps in eliciting effort and improving the quality of responses.

## 2.2 Model

We consider large scale crowdsourcing settings in which workers provide answers of many micro-tasks requiring human intelligence. The tasks have a discrete answer space  $\{0, 1, \dots, K-1\}$  of size  $K$ . We will use  $[K]$  to denote this space. For any task, our model has 3 random variables. The first is the unknown **ground truth**  $G$  answer for the task. The second is the worker  $i$ 's **observed answer**  $X_i$  that she obtains on solving the task.  $X_i$  is worker's private information. The third is the worker's **reported answer**  $Y_i$  that she actually reveals as her answer for the task. We use  $g, x_i, y_i \in [K]$  to denote realizations of these random variables and will drop the subscript  $i$ , when the context is clear.

**Definition 1** (Effort Strategy). *The effort strategy of a worker  $i$  is a binary variable  $e_i$ . If the worker invests effort in solving a task,  $e_i$  is 1 and is 0 otherwise.*

The effort strategy captures the standard binary effort model of the incentive mechanisms literature [20]. Whenever  $e_i = 1$ , the worker incurs a strictly positive finite cost.

**Definition 2** (Reporting Strategy). *When  $e_i = 1$ , the reporting strategy  $S_i$  of a worker  $i$  is a  $K \times K$  right stochastic matrix, where  $S_i[x, y]$  ( $\forall x, y \in [K]$ ) is the probability of her reported answer on a task being  $y$  given that her observed answer is  $x$ . When  $e_i = 0$ , the reporting strategy  $\vec{S}_i$  of a worker is a  $K$  dimensional probabilistic vector, where  $\vec{S}_i[y]$  ( $\forall y \in [K]$ ) is the probability of her reported answer on a task being  $y$ .*

The effort and the reporting strategy together model possible strategies that a worker may play in obtaining and reporting her answer and is a standard model in the literature [104]. Two common strategies are truthful and heuristic.

**Definition 3** (Truthful Strategy). *A worker  $i$ 's strategy is called truthful if  $e_i = 1$  and  $S_i$  is an identity matrix.*

In a truthful strategy, a worker solves a task and reports her answer as obtained.

**Definition 4** (Heuristic Strategy). *A worker  $i$ 's strategy is called heuristic either if  $e_i = 0$  or if  $e_i = 1$  and all rows of  $S_i$  are identical.*

In a heuristic strategy, a worker either doesn't solve the tasks ( $e_i = 0$ ) or solves the tasks ( $e_i = 1$ ) but reports independently of the obtained answer. Note that a common colluding heuristic strategy, in which workers collude using a "default" answer, is included in the model. For example, in binary case, when  $e_i = 0$ , a probabilistic vector  $\vec{S}_i = [1, 0]$  means that the worker always answers 0. Similarly, when  $e_i = 1$ , a matrix  $S_i$  with both rows equal to  $[1, 0]$  means the same. It is also easy to see that the model also includes mixed strategies since mixed strategies can be written as convex combination of the pure strategies.

**Definition 5** (Proficiency Matrix). *The proficiency matrix  $A_i$  for a worker  $i$  is a  $K \times K$  right stochastic matrix, where  $A_i[g, x] (\forall g, x \in [K])$  is the probability of her obtained answer on a task being  $x$  given that the ground truth is  $g$ .*

This definition is due to [21], which is a widely accepted model in crowdsourcing literature. The proficiency matrix models the ability of a worker to obtain correct answers, when she invests effort. Every worker can have a different proficiency matrix.

**Definition 6** (Trustworthiness Matrix). *The trustworthiness matrix  $T_i$  of worker  $i$  is a  $K \times K$  right stochastic matrix, where  $T_i[g, y] (\forall g, y \in [K])$  is probability of her reported answer on a task being  $y$  given that ground truth is  $g$ .*

Note the difference between the proficiency and trustworthiness matrices. Proficiency models worker's ability while trustworthiness is a function of her ability and honesty.

**Proposition 1.** *If  $e_i = 1$ , the trustworthiness matrix  $T_i$  of a worker  $i$  is given by  $T_i = A_i S_i$ . If  $e_i = 0$ ,  $T_i$  is a matrix with all rows equal to reporting strategy vector  $\vec{S}_i$ .*

Our model is summarized in Figure 2.1.

Until now, we defined the strategy space for the settings in which workers solve one task each. In our work, we consider settings, in which workers solve multiple tasks and next define the strategy space for multi-task settings.

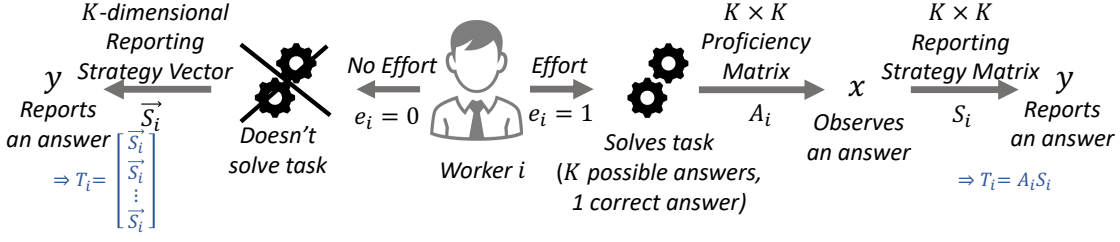


Figure 2.1: Model

**Definition 7** (Uniform Strategies for Multi-task Settings). *A worker's strategy (effort and reporting) in multi-task settings is called uniform if the strategy is the same on all tasks in a given batch solved by the worker.*

The uniform strategies are sometimes also called consistent strategies in the literature. It is important to note that the space of uniform strategies DOES INCLUDE mixed strategies. The motivation for considering only uniform strategies in the multi-task literature is that the tasks of similar nature can be grouped in batches so that workers don't strategically distinguish between tasks assigned to them.

**Definition 8** (Dominant Uniform Strategy Incentive Compatibility). *Given that workers can play any strategy from the space of uniform strategies, an incentive mechanism is called dominant uniform strategy incentive compatible (DUSIC) if the expected reward of any worker is strictly maximized by playing a truthful strategy, no matter what uniform strategies other workers use.*

In this notion of incentive compatibility, the reward of a worker is *strictly* maximized in a truthful strategy even if others are not truthful. Thus, the truthful strategy dominates any heuristic strategy of not solving the tasks and non-truthful strategies of solving the tasks but reporting non-truthfully. It also dominates any mixed uniform strategy.

**Definition 9** (Oracle). *An agent  $o$  is called an oracle if her trustworthiness matrix  $T_o$  is known and  $T_o$  doesn't have identical rows.*

For example, if the oracle is the source of gold standard answers, then by definition of gold tasks, oracle's trustworthiness matrix is an identity matrix.

We use  $P(g)$  to denote the prior probability of the ground truth answer of any randomly selected task being  $g$ . It is assumed to be known and fully mixed ( $P(g) > 0 \forall g \in [K]$ ). It can also be estimated from the gold standard answers.

## 2.3 Finding Trustworthiness Transitively

In this section, we first explain the main building block of our mechanism: the process of finding the trustworthiness of a worker, given the trustworthiness of another worker by using



the joint distribution of their answers on shared tasks.

**Definition 10 (Peer).** *For a worker  $i$ , the mechanism assigns another worker  $j$  as her peer. Workers  $i$  and  $j$  are assigned sets of tasks  $Q^i$  and  $Q^j$  respectively such that  $|Q^i \cap Q^j| \gg 0$ .*

The definition requires that some tasks are solved by both the worker and her peer. Both workers also solve some other tasks that are not shared. It may be noted that eliciting answers of multiple workers on same tasks is the central idea in crowdsourcing [105] and is not a new requirement introduced in our work.

Let  $T_j$  be the known trustworthiness matrix of worker  $j$  and let  $j$  be the peer of another worker  $i$ , whose trustworthiness matrix  $T_i$  is not known. We want to find the unknown  $T_i$  using the answers given by the two workers and the known  $T_j$ . Since the worker  $i$  and her peer  $j$  solve some shared tasks by definition, their reported answers on these shared tasks provide the mechanism with an empirical joint distribution of their answers. We use  $\omega(Y_i = y_i | Y_j = y_j)$  to denote this conditional empirical distribution and  $\omega(Y_j = y_j)$  to denote the empirical distribution of answers of peer  $j$  only.

**Lemma 1.** *As  $|Q^i \cap Q^j| \rightarrow \infty$ , the following holds w.h.p.*

$$\omega(Y_i = y_i | Y_j = y_j) = \sum_{g \in [K]} T_i[g, y_i] \cdot \left( \frac{T_j[g, y_j] \cdot P(g)}{\omega(Y_j = y_j)} \right) \quad (2.1)$$

$\forall y_i, y_j \in [K]$  and  $\omega(Y_j = y_j) \neq 0$ .

The proof of Lemma 1 is provided in the appendix. The LHS in Equation 2.1 is the conditional probability  $P(Y_i = y_i | Y_j = y_j)$  in the limit. When we apply Bayes' rule to write this conditional probability in terms of other model probabilities, we get the RHS of Equation 2.1. This assumes that the answers of workers  $i$  and  $j$  are conditionally independent given the ground truth.

In the linear system of Equations 2.1,  $T_i[g, y_i] \forall g, y_i \in [K]$  are unknowns. Since the matrix  $T_i$  is also right stochastic, we have as many equations as the number of unknowns. This system can be solved for  $T_i$ , provided the system is *well-defined*. This requires that  $\omega(Y_j = y_j) \neq 0$  and for a unique solution, the coefficient matrix of this linear system must have linearly independent rows. This system of linear equations can be solved analytically. In practice, many libraries are also available for computing the solution efficiently. We used the `numpy.linalg` library in Python for this purpose. We now use this transitive method of finding unknown  $T_i$  to develop our mechanism in the next section.

## 2.4 The Deep Bayesian Trust Mechanism

The Deep Bayesian Trust mechanism is summarized in Mechanism 1 on the next page. It maintains a pool of workers' answers which are "informative" for evaluating other workers. The meaning of the term *informative* will be explained later. The pool is initialized with some

### Mechanism 1 : The Deep Bayesian Trust Mechanism

---

1. Assign a set of tasks to the oracle  $o$  and obtain its answers on the tasks.
  2. Initialize an *Informative Answer Pool* (IAP) with the answers given by oracle.
 

$$IAP = \left[ \left[ o : T_o : q_1 - \mathbf{a}_1, q_2 - \mathbf{a}_2, q_3 - \mathbf{a}_3, \dots \right] \right]$$

$o$  stands for oracle,  $T_o$  is the trustworthiness of the oracle and  $q_l - \mathbf{a}_l$  are the task – answer pairs provided by the oracle.
  3. Select some tasks submitted by a worker from the IAP. If there is no worker yet in the IAP, select the oracle's tasks.
  4. Prepare a set of batches of tasks such that each batch contains tasks selected in the previous step. Mix some fresh tasks in each of the batch.
  5. Publish the batches on the platform and let workers self-select themselves to solve one batch each.
  6. For any worker  $i$  who submits her batch, solve the system of Equations in 2.1 to find the unknown trustworthiness  $T_i$ . Reward worker  $i$  for her answers with an amount equal to  $\beta \cdot R_i$  where,  $R_i = \left( \sum_{g \in [K]} T_i[g, g] \right) - 1$ .
  7. If the answers of worker  $i$  satisfy the informativeness criterion, add the answers to the IAP and assign them trustworthiness  $T_i$  as obtained in Step 6.
 

For example, at a given instant, the pool may look as follows :

$$AP = \left[ \left[ o : T_o : q_1 - \mathbf{a}_1, q_2 - \mathbf{a}_2, \dots \right], \left[ W_1 : T_{W_1} : q_2 - \mathbf{a}_2, q_4 - \mathbf{a}_4, \dots \right], \left[ W_2 : T_{W_2} : q_2 - \mathbf{a}_2, q_5 - \mathbf{a}_5, \dots \right], \dots \right]$$

Here,  $W_i$  are the identities of workers followed by their trustworthiness  $T_{W_i}$  and their submitted task-answer pairs.
  8. *Asynchronously* repeat steps 3, 4, 5, 6 and 7 whenever any worker submits her batch, until desired number of answers are collected for all tasks.
- 

tasks and their answers given by the oracle. In crowdsourcing terminology, these are the gold task-answer pairs. The trustworthiness matrix of the oracle is initialized to be  $T_o$ . Since by definition, gold tasks are the tasks whose *correct* answers are known,  $T_o$  is an identity matrix. The mechanism then publishes several batches of tasks on the platform such that each batch has some tasks in common with the tasks solved by the oracle and some unique new tasks in each batch. Workers self-select themselves to solve one batch each and report their answers for respective batches. Thus, the oracle becomes the peer of each of these workers.

## 2.4. The Deep Bayesian Trust Mechanism

Let's assume that the oracle solves  $s_o$  number of tasks. The mechanism publishes  $k$  batches of tasks such that there are  $s_o$  tasks in common with the oracle and  $s_n$  unique new tasks in each batch. Thus, it publishes  $k \cdot s_n$  tasks that are new (not solved by the oracle already) and also  $k$  instances of the same  $s_o$  tasks that are already solved by the oracle.  $k$  becomes a hyper-parameter of the mechanism and  $s_o + s_n$  becomes the size of the batches solved by every worker.

As the workers start submitting their respective batches, the mechanism also starts rewarding the workers for their answers, asynchronously (without waiting for other workers). To calculate the reward, the mechanism uses Lemma 2.1 for finding the trustworthiness matrix of the answers given by workers. Note that the lemma is applicable because the trustworthiness of the peer (oracle) is known. The reward for worker  $i$  is given by  $\beta \cdot R_i$ , where  $R_i = \left( \sum_{g \in [K]} T_i[g, g] \right) - 1$  and  $\beta$  is a scaling constant.

$R_i$  takes the summation of the diagonal entries of the trustworthiness matrix  $T_i$ . These are the accuracy parameters of the worker (the probabilities of the workers' answers being same as the ground truth).  $R_i$  further subtracts 1 from this summation for technical reasons that will be exploited later to ensure a desired incentive property.

The worker gets her reward and is out of the mechanism. The mechanism now decides whether to reuse the answers given by the worker for evaluating more workers. If the worker's answers satisfy a certain "informativeness" criterion, they are added to the pool. If the worker's answers are added to the pool, the mechanism can immediately publish more batches such that there are some tasks in common with the new (non-gold) tasks solved by the previous worker and some more new tasks in each of the batches. This step is the same as described earlier. The only difference is that now the batches being published have tasks in common with the tasks solved by a worker, not the oracle (i.e. the peers are now other workers, not the oracle). These steps are repeated in parallel and asynchronously until the mechanism has obtained the desired number of answers for all its unsolved tasks. Note that non-gold tasks are assigned to multiple workers in crowdsourcing for obtaining more accurate aggregate answers and this is not a waste of effort/budget.

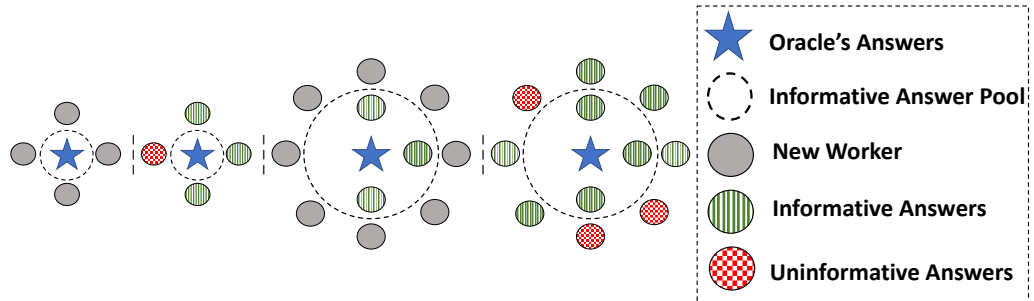


Figure 2.2: Illustration of the Deep Bayesian Trust Mechanism

To summarize (Figure 2.2), the mechanism starts with an answer pool seeded with the oracle's

answers, uses the answers in the pool to assess trust in other workers' answers, expands this pool based on the informativeness of the workers' answers and repeats the process. We emphasize that the mechanism doesn't assign any permanent reputation to workers. A worker's answers being added to the pool is NOT equivalent to a worker being "pre-screened". We only evaluate the answers provided by a worker in any given batch and add them to the answer pool together with an estimate of the trustworthiness of that batch of answers.

### Informativeness Criterion

We can now discuss the informativeness criterion for workers' answers, which was omitted earlier. The purpose of the informativeness criterion is to check whether the answers provided by a worker  $i$  can be used to estimate the trustworthiness of another worker or not. As discussed in Section 2.3, this depends on whether the coefficient matrix of the linear system of equations has linearly independent rows or not. For example, assume that the answers of worker  $j$  are added to the pool and let  $i$  be another worker who gets  $j$  as her peer in the future. In that case, the mechanism will solve the following equations to estimate the trustworthiness of  $i$ :

$$\omega(Y_i = y_i | Y_j = y_j) = \sum_{g \in [K]} T_i[g, y_i] \cdot \left( \frac{T_j[g, y_j] \cdot P(g)}{\omega(Y_j = y_j)} \right)$$

$$\forall y_i, y_j \in [K]$$

The coefficients  $\frac{T_j[g, y_j] \cdot P(g)}{\omega(Y_j = y_j)}$  of this linear system don't depend on the answers given by worker  $i$  and the mechanism can determine in advance whether the system will be solvable by just looking at these coefficients. If  $\omega(Y_j = y_j) \neq 0$  and the coefficient matrix is full rank, the informativeness criterion is said to be satisfied and the answers of worker  $j$  are added to the pool. It may be noted that the informativeness criterion doesn't require answers of only truthful or high proficiency workers to be added to the pool. Answers from non-truthful or low proficiency workers can also be added to the pool.

To understand this technical criterion in more depth, note that the coefficients  $\frac{T_j[g, y_j] \cdot P(g)}{\omega(Y_j = y_j)}$  of the linear system are equal to the posterior distribution  $P(G = g | Y_j = y_j)$  by Bayes' rule. Thus, the answers of a worker  $j$  satisfy the informative criterion if the posterior distributions  $P(G = g | Y_j = y_j)$  over  $g \in [K]$  for any two different  $y_j \in [K]$  are not identical. One interesting example, where the informative criterion is not satisfied, is when the peer  $j$  plays a heuristic strategy. In such a case, the reported answers are not correlated with the ground truth and the posterior distributions  $P(g | Y_j = y_j)$  are same as the prior distribution  $P(g)$  or in other words, the reported answers are not "informative" of the ground truth.

It may be noted that the informativeness criterion is fairly weak for the binary answer spaces. It only requires that the reports of the peer  $j$  are not independent of the ground truth ( $T_j[1, 1] + T_j[2, 2] \neq 1 \implies T_j[1, 1] \neq T_j[2, 1] \implies T_j[2, 2] \neq T_j[1, 2]$ ).

## 2.5 Theoretical Analysis

We now prove strong game-theoretical properties for our mechanism. In this discussion, we will assume that a worker and her peer solve many shared tasks ( $|Q^i \cap Q^j| \rightarrow \infty$ ). This is **not** the same as requiring every task to be solved by large number of workers, which would have been inefficient. In later sections, we will also discuss the empirical performance of our mechanism without this assumption. We use  $C^E$  to denote the cost of effort required to solve a batch of tasks. Proofs are provided in the appendix.

**Theorem 1.** *If  $\beta > \frac{C^E}{\left(\sum_{g \in [K]} A_i[g, g]\right)^{-1}}$  and  $A_i[g, g] > A_i[g', g], \forall g' \neq g$ , then the Deep Bayesian Trust mechanism*

- (i) *is dominant uniform strategy incentive compatible (DUSIC) for every worker  $i$ ;*
- (ii) *ensures strictly positive expected reward in the truthful strategy.*

Theorem 1 requires a condition on the scaling constant  $\beta$  to cover the cost of effort, and reduces to  $\beta > 0$  when cost of effort is 0. The condition required on proficiency matrix  $A_i$  (i.e.  $A_i[g, g] > A_i[g', g], \forall g' \neq g$ )<sup>2</sup> can be more easily understood in the case of binary answers. In binary settings, the condition is satisfied if  $A_i[0, 0] > 0.5$  and  $A_i[1, 1] > 0.5$ . This is **not** a condition on the honesty of the workers but only on their ability. The condition merely ensures that the worker can obtain answers that are positively correlated with the ground truth. Such conditions are common in the literature [20]. Unlike the literature, the condition here only affects the best strategy of a given worker, not of all the workers. For example, if the condition is not satisfied for a worker, she may find it better to deviate to a non-truthful strategy but it doesn't affect the dominant strategy of other workers. We note that such informed deviation by a low proficiency worker to increase the accuracy of her reported answers is not bad for the requester.

**Corollary 1.** *The scaling constant  $\beta$  of the Deep Bayesian Trust mechanism is independent of the probability of a worker getting oracle or another truthful worker as peer.*

Corollary 1 implies that to ensure incentive compatibility, our mechanism doesn't need to scale up the rewards of workers if the probability of a worker getting oracle or another truthful worker as peer decreases.

**Theorem 2.** *In the Deep Bayesian Trust mechanism, a heuristic strategy gives zero expected reward.*

It may be noted that the DUSIC result in Theorem 1 already implies that the heuristic strategies are not in equilibrium but Theorem 2 answers the question that what if someone still plays those strategies.

<sup>2</sup>For binary answer space, the theorem can also be shown to hold under a weaker condition  $A_i[0, 0] + A_i[1, 1] > 1$ .

### Limitation

If, despite all these guarantees, every single worker chooses to *irrationally* play a heuristic strategy, then our mechanism will not be able to expand its pool and will be forced to behave like other mechanisms which assign gold tasks to every worker. But (i) such workers don't gain anything from the mechanism; (ii) the dominant incentive compatibility of the mechanism remains unaffected for any *rational* workers even in such a degenerate case; and (iii) the heuristic strategy doesn't become an equilibrium strategy.

### 2.5.1 Fairness of Rewards

Recently, concerns have been raised about fairness and other ethical considerations in algorithms that affect humans [3, 53]. The discussion on fair rewards in crowdsourcing has included issues such as minimum wages and adequate compensation for time and effort [102] but there has not been any principled approach to address the issue of fairness in rewards from a non-discrimination perspective. For example, if a worker with higher ability gets a lower reward than a worker with lower ability because of the difference in the way they were evaluated, then this is a potential case of unfairness. The unfairness is an unintentional and undesired property of the existing mechanisms. Peer based mechanisms in the literature randomly select peers and reward the workers based on their answers and the answers of their respective peers. The reward of the workers is generally a function of their own ability as well as their peers' ability, making the rewards unfair. This unfairness issue in the peer based mechanisms was first pointed out by [59]. The issue becomes more serious when workers know ex-ante that they are being evaluated using peers with different proficiencies. This is the case, for example, with the mechanism of [22]. Our mechanism doesn't need to inform the workers about their peers at all but as we show, the mechanism can satisfy an even stronger definition of fairness.

**Definition 11** (Fair Incentive Mechanism). *An incentive mechanism is called fair if the expected reward of any worker is directly proportional to the accuracy of the answers reported by her and independent of the strategy and proficiency of her random peer.*

This is a reasonable definition of fairness and is in agreement with the broader theory for **individual fairness** of algorithms. For example, the pioneering work of [27] defines that fair algorithms take similar decisions for individuals with similar relevant attributes. The relevant attribute in our case is the worker's accuracy. The definition is also non-trivial to satisfy. In existing peer based mechanisms, the rewards also depend on the unknown ability of the peer (even if the peer can be believed to be truthful). For example, in the mechanism of [20], the reward of a worker in the truthful equilibrium is an increasing function of her proficiency as well as her peer's proficiency. On the contrary, our mechanism satisfies this definition of fairness. The mechanism carefully uses the peer answers only to find trustworthiness of a worker, which is completely her own accuracy parameter and doesn't depend on her peer's proficiency or strategy.

**Theorem 3.** *The Deep Bayesian Trust Mechanism is fair.*

This is perhaps a surprising result because in the existing framework of the peer based mechanisms, one would perhaps reason that it is impossible for the rewards to not depend on the accuracy of the peer.

## 2.6 Numerical Simulations

In this section, we evaluate the performance of our mechanism empirically. We simulate the settings in which workers with different proficiencies  $A_i$  report answers to different tasks. The proficiency matrices of different workers were generated independently such that the diagonal entries  $A_i[g, g] \forall g \in [K]$  were  $\beta(5, 1)$  distributed. The diagonal entries  $A_i[g, g] \forall g \in [K]$  for a given worker  $i$  are not necessarily the same as they are also independently generated. Rest of the entries are generated randomly such that every row of proficiency matrix sums to 1.

We consider following strategies that workers may play:

1. **Truthful** - Workers obtain answer for any given task based on their respective proficiency matrices and report the answers truthfully.
2. **Heuristic** - Workers' reported answers are generated independently of their proficiency using a common distribution over the answer space.
3. **Permutation** - Workers obtain answer for any given task based on their respective proficiency matrices but they apply a common permutation on the answers before reporting it to the mechanism. In a non-truthful permutation deterministic strategy [104], workers solve the tasks, but they apply a permutation mapping on the answers before reporting it to the mechanism. For example, in a ternary answer space ( $K = 3$ ), a permutation  $f$  can be as follows :  $f(0) = 1, f(1) = 2, f(2) = 0$ , i.e., whenever the obtained answer is 0, workers report 1, for 1, they report 2 and for 2, they report 0. In a binary answer space, this corresponds to reporting the opposite of the obtained answer.

In general, the simulations performed in the literature for peer based mechanisms compare the average reward in different equilibria. For example, the average reward of workers when all of them play a truthful strategy may be compared with the average reward when all play a heuristic strategy. This is because such mechanisms only guarantee that different strategies are in equilibria and that one equilibrium is more profitable than the other. But our stronger theoretical result (dominant incentive compatibility) demands stronger simulations. We go beyond comparing just equilibrium rewards and instead compare the rewards of workers playing different strategies against one another at the same time. More precisely, in our simulations, we don't require every worker to play a common strategy. Any worker can play a heuristic, permutation or truthful strategy with equal probability. Such settings can't be

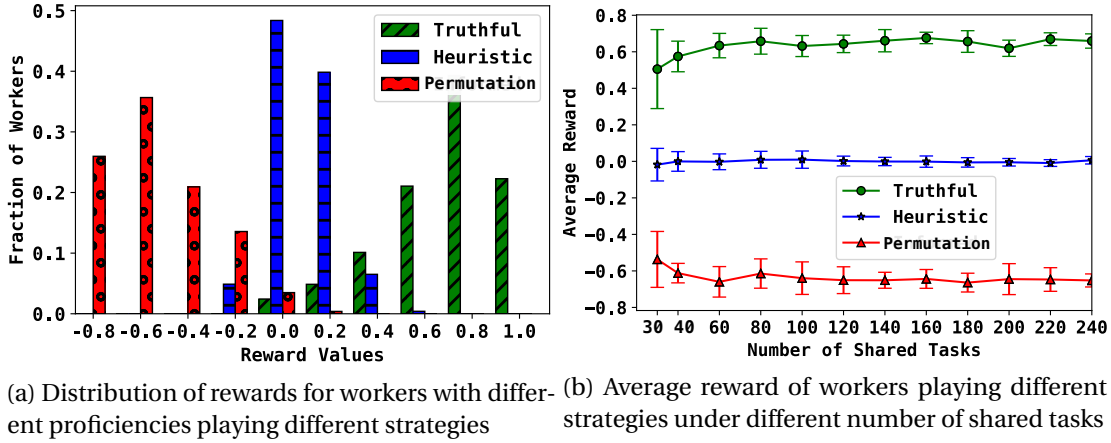


Figure 2.3:  $\beta$  distributed proficiencies

handled by mechanisms that guarantee only equilibrium results. We will show that in our mechanism, there is a clear distinction between the rewards of workers playing different strategies with truthful workers being nicely rewarded and others being penalized. Workers were simulated to be hired in 4 rounds, with 5, 25, 125 and 625 workers in successive rounds.  $K$  was set to 2 in all the experiments discussed in the chapter.

Figure 2.3a compares the distribution of rewards of workers playing the three strategies. The rewards of the workers playing the heuristic strategy are centered around 0, as expected from Theorem 2. The reward of workers playing truthful strategy are centered around a strictly positive value as predicted by Theorem 1. On the contrary, the rewards of workers playing the permutation strategy are symmetrically centered around a strictly negative value. It may be noted that in existing peer based mechanisms, permutation strategies (in equilibria) are equally profitable as the truthful strategy, which is clearly not the case with our mechanism. Firstly neither heuristic nor permutation strategies are in equilibrium in our case and even if workers use any of these strategies, they get lower reward than the truthful strategy.

We now show the **robustness of our mechanism with respect to the number of shared tasks between workers**. We discussed only the asymptotic properties of the mechanism earlier in the theoretical analysis. Hence, this simulation study is important to show the performance of the mechanism with a finite number of shared tasks. Figure 2.3b compares the average of rewards of the workers (with  $\beta(5, 1)$  distributed proficiencies) playing different strategies under different settings of the number of shared tasks. Error bars show the standard deviation in 100 repeated runs. The trend discussed in previous experiment can be observed to be very robust to the number of shared tasks. Thus, the Deep Bayesian Trust mechanism is attractive even when the number of shared tasks is not large. This simulation also implies that with only 30 gold tasks (and given only to 5 workers), the mechanism can reward  $5 + 25 + 125 + 625 + \dots$  workers.

We also simulated the settings in which the diagonal entries  $A_i[g, g] \forall g \in [K]$  were uniformly



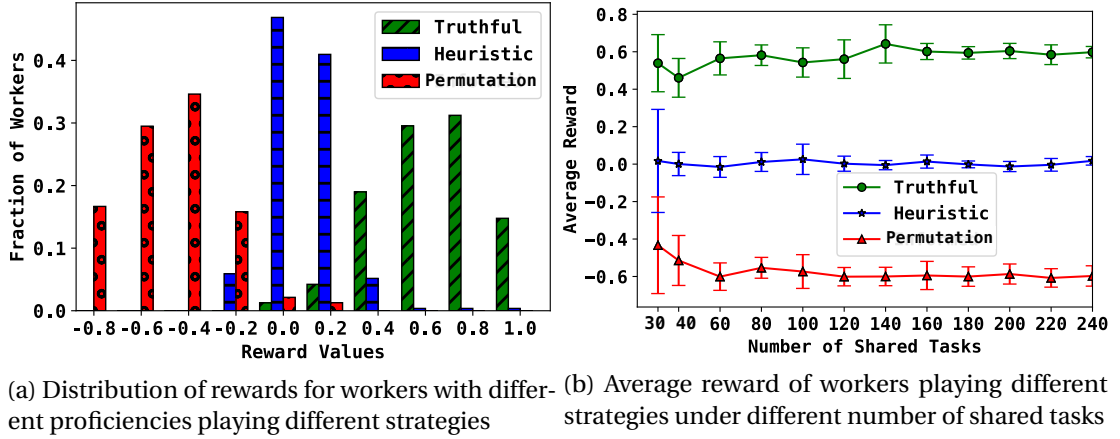


Figure 2.4: Uniformly distributed proficiencies

distributed in  $(\frac{1}{K}, 1]$  and repeated the above experiments.

## 2.7 Preliminary User Study

Any conclusive study on the effects of reward mechanisms requires a large budget and needs to be conducted over a long period. Such a study is beyond the scope of this thesis. We performed a limited scale study on Amazon Mechanical Turk to make some preliminary observations about the ease of implementing the mechanism and workers' response to the mechanism. We created some synthetic tasks which resemble tasks requiring human intelligence (natural language understanding) and elicited the answers of the crowd on MTurk with and without our reward mechanism in place. The advantage of using synthetically generated tasks was that we had access to ground truth and thus, we could judge the performance of workers with and without our mechanism objectively. The structure of our task (named '*story disentanglement task*') was the following : we mixed a few sentences from different real news stories into one paragraph and asked the workers to count the number of news stories in the paragraph. Solving this task requires identifying the context of different sentences and whether they are related. The number of sentences in paragraphs was kept independent of the number of stories, making it harder to guess by just looking at the paragraph length. We asked workers to give a binary answer ('Yes' if the number of stories is less than 3 and 'No' otherwise). We also asked them to identify the sentences belonging to different stories. We will discuss only the binary answers of the workers in this chapter. Each HIT<sup>3</sup> corresponds to giving an answer for one such paragraph. We conducted the experiments under two settings.

- In the first setting, workers were told that they would be paid 0.03\$ per HIT and there would be additional performance based payments without discussing a specific reward rule. We will refer to this setting as the **unspecified** reward setting.

<sup>3</sup>A single task is referred to as HIT(Human Intelligence Task) on Amazon Mechanical Turk.

## Chapter 2. Deep Bayesian Trust: A Dominant and Fair Incentive Mechanism for Crowd

HIT Preview

Story Disentanglement Instructions (Click to collapse)

Please read the instructions below carefully and attempt our task only if you commit to solving the complete task. If you don't read the instructions or spam our system, the task will be rejected and the worker ID will be permanently blocked for our future tasks.

**Payment Scheme:**

In addition to the fixed payment (80 x \$0.03 = \$2.4), there will be a performance based bonus. The bonus will be calculated in the following way :

We have prepared a set of batches each containing 80 micro-tasks. Among these batches, there are a few gold batches. The gold batches contain some micro-tasks for which we already know the correct answers. If you are assigned one such batch, your reward will depend on your accuracy on these tasks. The rest of the batches are designed in such a way that they have some tasks in common with one another. If you are assigned this kind of batch, we will determine your accuracy by looking at your answers and the answers obtained on a chain of batches starting from one of the gold batches. For example, if your batch is not a gold batch and has some tasks in common with a gold batch, then we will first determine the accuracy of the worker solving gold batch, then using this accuracy we will determine your accuracy through the answers given on common questions. Thus, irrespective of the type of batch you are assigned, your reward will depend only on your accuracy and nothing else. Your bonus will be proportional to difference of your accuracy and random guessing accuracy. For example, if your accuracy is 50% (which is one way of random guessing), your bonus will be 0. **The bonus can also be negative, which means that if your accuracy is low, the negative bonus will reduce your fixed payment of \$2.4 as well.** It is important to note that accuracy is an aggregate over all the tasks you solve, hence a consistent effort throughout the batch will be required to maintain high accuracy.

The tasks are hosted on an external website. The link is given below in the task. Once you have submitted answers for the entire batch, you will get a unique token. You can submit this token as a solution on Amazon Mechanical Turk. This will let us know that you have solved our tasks and we will process your payment.

### (a) Description of Payment Scheme

**Sample Task:**

In this job, you will be presented with 80 short text paragraphs. Each of these paragraphs have a mix of random parts of several news stories but we don't know exactly how many stories there are in each of the paragraphs. Help us in finding out.

Steps :

1. Read the given text carefully until very last sentence.
2. Determine how many stories are being told in the text.
3. Select 'YES' if you find that there are 1 or 2 stories in the text and select 'NO' if there are 3 stories.
4. Further, mention the last word of each story in the format described in example below. It is important to follow the format, otherwise we will not be able to use your answers.

Rules and Tips :

Each of the paragraphs contain many stories. Stories can be from the same field or different fields such as politics, business, sports, technology and entertainment. If they are from different fields, you may find it comparatively easier to count the stories but in general there are no guarantees that the stories will be from different fields. Any story can end abruptly in the paragraph and a new story can begin. Any story in the paragraph can begin from any random sentence of the original story. Hence, reading every sentence with full attention is required. There is no correlation between the number of stories and the length of the paragraph. A paragraph with 2 stories can be of the same length as a paragraph with 3 stories.

### (b) Task Instructions

Example :

Consider the following text paragraph :

The panel suggested different kinds of murders could be "graded" to recognise the seriousness of the offence. But he argued the minister should have volunteered a formal statement instead of having to be "dragged" to face MPs. Sound distribution of the cash could cut poverty levels to 36% from 53%, the government believes. Colombia has a population of about 44 million and half lives below poverty line. The seventh seed, who has never gone beyond the quarter-finals in the year's first major and is lined up to meet Roddick in the last eight, is looking forward to the match.

This paragraph contains parts of 3 different news stories as shown below. The first story is about some court case. The second story is about demographics in Colombia. The third story is about about an sports match. Hence, you should select 'No'.

The panel suggested different kinds of murders could be "graded" to recognise the seriousness of the offence. But he argued the minister should have volunteered a formal statement instead of having to be "dragged" to face MPs.

Sound distribution of the cash could cut poverty levels to 36% from 53%, the government believes. Colombia has a population of about 44 million and half lives below poverty line.

The seventh seed, who has never gone beyond the quarter-finals in the year's first major and is lined up to meet Roddick in the last eight, is looking forward to the match.

The last word of each of the stories are : MPs, line, match. Hence, you should write :

MPs, line, match

in the text field. You can ignore any special characters such as " or . while writing the last word.

### (c) Sample Task

Figure 2.5: Mechanical Turk Task

- In the second setting, each HIT was worth 0.03\$ and we explained our Deep Bayesian Trust mechanism to the workers in plain English with almost no use of mathematical language or notations. Figures 2.5a, 2.5b and 2.5c show screen shots of the instructions given to the crowdworkers. We will refer to this setting as the **DB Trust** setting.

In the DB Trust setting, batches of 80 HITs were designed such that each batch had 40 HITs in common with another batch to satisfy peer relationship. In both settings, we had 3 workers giving answers for each paragraph, giving us a total  $3 \times 480$  HITs from 480 paragraphs. We thus collected a dataset of 1440 worker responses on these HITs, 720 in each setting. In total, 129 workers participated in the experiment. We judge the mechanism on two most important criteria. First, the ability to discourage workers from heuristic reporting and second, the ability to get more accurate answers from crowd.

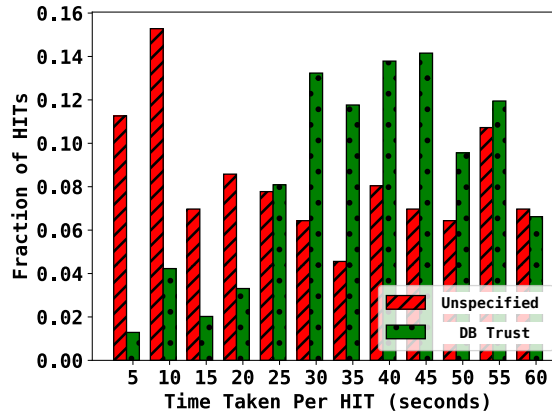


Figure 2.6: Time Spent on HITs

## Observations

1. Figure 2.6 compares the time workers spent on solving the tasks in the two settings. The reported time doesn't include time spent on reading instructions and other details about the payment scheme. The fraction of HITs that were given very little time has significantly decreased with our mechanism and the fraction of HITs that were given more time has significantly increased (the green distribution with dots is more skewed towards the right side as compared to the red distribution with slashes, which is more skewed towards the left). This can be interpreted as a success in eliciting effort from the crowd and discouraging low quality/heuristic reporting. We used a browser based JavaScript solution to measure the actual time spent on solving tasks to get tight estimates of time spent in the DB Trust setting, without workers being aware of it. Amazon uses the difference between time of accepting and submitting a HIT as estimates of time spent, which (even after filtering very large values) tend to be highly inflated. As one can see, even with such tight estimates in the DB Trust setting, the time spent by workers is better.

2. The average accuracy of workers was found to increase from 70.86% in the unspecified setting to 79.17% in the DB Trust setting.
3. The average accuracy of all responses was also found to increase from 75.69% to 79.17% with our scheme.

### 2.8 Chapter Summary

We proposed the Deep Bayesian Trust mechanism to incentivize crowdworkers in large scale settings. The mechanism rewards the workers for the correctness of their reports without checking every worker with gold tasks. Instead, it uses the correlation in the answers of the workers and their peers to estimate their accuracy. The mechanism is guaranteed to be game theoretically robust to any strategic manipulation. Thus, it is also suitable even for the settings in which workers of very heterogeneous proficiencies and motivations solve the tasks at the same time. The mechanism also ensures fair rewards to workers, thus contributing towards the bigger movement of making algorithmic decisions fair. Among other issues, our mechanism notably addresses the scalability issues in purely gold tasks based mechanisms, the incentive compatibility issues in purely peer based mechanisms and the information requirement and fairness issues in the mixed mechanisms.

# 3 Personalized Peer Truth Serum for Eliciting Multi-Attribute Personal Data

This chapter is an extended version of the following publication:

N Goel, B Faltings. **Personalized Peer Truth Serum for Eliciting Multi-Attribute Personal Data.** In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.

## 3.1 Introduction

Crowdsourcing is a promising method to collect data in an inexpensive way. The data can be, for example, subjective opinions such as restaurant reviews or objective measurements such as pollution levels in a city and image labels. Measurements tasks are particularly important in collecting features which are useful in supervised and unsupervised machine learning. However, there is always a concern about the reliability of the data thus obtained. While some crowdworkers (henceforth called agents) will do their best to provide accurate data, many are not motivated to make the effort to obtain and report the data properly. This degrades the quality of the data. For example, if the data is to be observed with a sensor device, many may not be willing to buy and maintain the device to obtain correct measurements. One approach to address this problem is providing the agents with incentives that cover the cost of their effort and encourage them to provide high quality data. The incentives have to be contingent on the quality of the data, for example, based on spot-checking the data for agreement with a trusted ground truth. In many of the most interesting applications, however, the ground truth is not accessible. Peer consistency is an elegant idea for designing incentive mechanisms in this situation. The output agreement [110], the Bayesian truth serum [93, 113], the peer prediction [82], the peer truth serum [97] and the correlated agreement [20, 104] are all examples of the peer consistency mechanisms.

There is a lot of interest in extending this approach to collecting information such as records of personal sports activity, physiological measurements or diet. Other examples of personal information include sensor measurements observed at private properties such as smart homes



Figure 3.1: Strategic agents in crowdsourcing.

or hotels. However, a fundamental limitation of the existing peer consistency mechanisms is that they require that a group of agents, called peers, observes the same data or a noisy version of it. For example, a group of agents should label the same image, measure pollution at the same location or give opinion about the same service. This does not work with personal data, since every agent is reporting data about a different object. However, we can extend the idea of peer consistency to a setting where agents report a *combination* of attributes that are known to be correlated with one another, even if the correlation structure is not known. When rewarding the report about one of the attributes, we can identify peers based on similarity in the *other* attributes reported at the same time.

In this chapter, we show how one such mechanism, the *Logarithmic Peer Truth Serum (LPTS)* [96], can be extended to elicit multi-attribute personal data from a crowd. We call this novel mechanism the *Personalized Peer Truth Serum (PPTS)*. We introduce task settings for eliciting continuous valued personal features from agents, exploit them to develop the PPTS and discuss the theoretical properties and practical applicability of the new mechanism. Our mechanism works in large scale settings when there are multiple agents reporting data and there exist (unknown) groups of agents sharing some personal characteristics. We show the validity of our assumptions on real datasets. *We also show that even when these groups are estimated from the data reported by agents, the incentive compatibility of the mechanism is not affected.* We then show that once the data has been collected, the PPTS reward scores can also be used to filter the non-truthful data points with very high accuracy. The filtering scheme not only shows better empirical performance than many outlier detection algorithms but also it comes with a sound explanation of filtering the data based on truthfulness.

#### 3.1.1 Related Work

Quality control in crowdsourcing [52] is a widely recognized research issue. The issue arises mainly because of the impossibility to verify the reports due to absence of ground truth. Various complementary approaches exist in the literature to improve the quality of crowdsourced data. The approaches can be broadly divided into two categories. The first category includes post-collection quality control measures, for e.g., unsupervised truth-discovery [61],[73],[120],

which jointly estimate workers' credibility and latent objective truth in the data, with some recent work [112] in direction of estimating subjective truth. There are also techniques designed specifically for aggregating crowdsourced data coming from workers with different competence [61, 75, 98, 121]. However, truth discovery has its limitation [111] of not working if data comes from strategic workers [25], for example, if all workers collude to report the same false answer. Also, it works well only if there is enough data available about every worker. Due to these limitations, there is a parallel effort to develop techniques that help in getting as high quality data as possible to start with. This is the second category of approaches, which includes designing [64] and assigning [80, 122] tasks for optimal quality and providing performance based payments or incentives [46].

The goal of information elicitation mechanisms is to encourage workers to invest effort and honestly report their observations. Different information elicitation mechanisms exist in the literature for two major settings. Techniques such as proper scoring rules [36] and prediction markets [115] can be used to elicit truthful beliefs about events that are to be realized in future, if the realized outcomes of the events are observable by the mechanism. When such verification is not possible, peer-prediction mechanisms are a well-known solution for truthful information elicitation. In this chapter, we are interested in the truthful mechanisms for information elicitation without verification.

The original peer-prediction method [82] is a mechanism for information elicitation without verification. The mechanism uses proper scoring rules to reward agents for reports that are predictive of other agents' reports and admits truth-telling as a Nash equilibrium. Several other methods [55] don't use proper scoring rules and instead use an "automated mechanism design" approach to determine adaptive payment rules that are incentive compatible. However, these mechanisms are not detail-free in the sense that they require agents' beliefs to be known.

The Bayesian Truth Serum (BTS) [93] is another classic mechanism for information elicitation without verification. BTS doesn't use the knowledge of common beliefs to compute rewards, but collects two reports from each agent - an 'information' report (agent's own observation) and a 'prediction' report (agent's prediction about the distribution of information reports from other agents). The reward mechanism of the BTS ensures that truthful reporting is the highest-reward Nash equilibrium as number of agents solving a task tend to infinity. The Robust BTS of [95, 114] generalize the BTS to small populations in binary and non-binary answer settings respectively. These mechanisms are not minimal in the sense that they ask the agents to submit additional information than desired.

Several minimal and detail-free game theoretic incentive mechanisms have been developed recently for crowdsourcing. The seminal work in this category is due to [20]. The main idea in this work is to exploit multi-task settings, in which every agent solves multiple tasks. The mechanism rewards the agents for agreeing on a shared task and penalizes them for agreeing on a non-shared task. This mechanism ensures that truth-telling is a focal equilibrium in binary answer settings. The Correlated Agreement mechanism [104] generalizes the mechanism

of [20] to non-binary answer spaces with additional assumptions on the correlation structure of workers' observations. Both these mechanisms require that workers solve multiple tasks. The Logarithmic Peer Truth Serum [96], which is based on an information theoretic principle, requires no such assumptions and ensures strong-truthfulness in non-binary answer spaces. [70] provide further complementary analysis for this information-theoretic framework. The guarantees of the mechanism are ensured in the limit (when every task is solved by an infinite number of workers). The Peer Truth Serum (PTSC) of [97] doesn't require even this assumption for the theoretical guarantees and works with a bounded number of tasks overall. The Deep Bayesian Trust mechanism [37] ensures dominant strategy incentive compatibility and also computes fair rewards in large scale crowdsourcing by using both peer answers and some gold standard answers. The fundamental assumption in all of the above mechanisms is that the task solved by an agent can be shared with another agent, who submits independent noisy observation. [5] extend the Correlated Agreement mechanism to the settings where agents belong to one of the  $k$  possible categories of rating behaviors (for e.g. strict and lenient rating behavior). They cluster the agents with similar rating behavior to apply the CA mechanism. However, in this mechanism too, the assumption of shared tasks remains.

All these mechanisms are inherently inapplicable to elicit personal data. This is because when workers are asked to report measurements about the personal objects she owns (for example, her body or house), no other worker can share that task (because no worker can access the personal object owned by another worker). We extend the Logarithmic Peer Truth Serum to this setting while using a concept similar to that of "peers". In such a setting, these peers can not be distinguished using the 'shared task' definition. Our mechanism approximates them from the data reported by the workers while guaranteeing truthful equilibrium.

#### 3.1.2 Our Contributions

From a technical standpoint, we address three main challenges in this work:

1. Define which agents can act as peers for one another in settings when agents can't share tasks.
2. Show that even if such peers are estimated from the reports submitted by the agents, the incentive compatibility is not affected.
3. Extend the mechanism to handle continuous data values instead of only discrete answers.

The summary of our contributions in the chapter is as follows:

- We propose a novel incentive mechanism to elicit **continuous valued, multi-attribute and personal** data from crowd.



- We analyze and present several interesting theoretical properties of the mechanism. Our mechanism ensures that truthful reporting is an equilibrium and other undesired equilibria are less attractive. We also provide a practically useful and theoretically sound test to judge the applicability of our mechanism on a new type of data to be elicited.
- We show the performance of the mechanism on three real datasets, which are publicly available and are relevant to the settings of the chapter.
- We show experimentally the utility of PPTS as a post-processing cleaning method to filter out non-truthful data points with high accuracy.

## 3.2 Settings

We consider the settings in which a requester (center) is interested in collecting data from a large number of agents  $W$  ( $|W| = n \rightarrow \infty$ ) with some personal characteristics. The data being elicited consists of a set of attributes  $A$  ( $|A| = d \geq 2$ ). The attributes  $A$  are personal characteristics such as body measurements of the agents. Agents independently obtain measurements for their attributes and report them to the center. The center in turn rewards them based on the quality of their reports. We assume the agents to be rational, seeking to maximize their expected rewards. The agents choose a reporting strategy to maximize their expected rewards. In a heuristic reporting strategy, they save the effort of even measuring the attribute and just report a random measurement drawn from an arbitrary probability distribution. In an informed reporting strategy, they obtain the measurement but report a mapping of the obtained measurement. Our aim is to formulate our incentive mechanism as a Bayesian game between the agents (who have probabilistic beliefs about the measurements of one another) and make truthful reporting (i.e. informed reporting with identity mapping) a profitable equilibrium strategy of the game for all agents. The strategic setting is described in Figure 3.1.

### 3.2.1 Belief Model

We model the beliefs of an agent  $i$  using three continuous random variables for each attribute  $j$ . The first random variable  $X_{ij}$  is the attribute measurement itself.  $P(X_{ij})$ <sup>1</sup> is agent  $i$ 's prior belief about measurements for the attribute  $j$ . The second random variable  $G_j$  models the global factors that affect the value of the  $j^{th}$  attribute of any random agent.  $P(G_j)$  is the agent's prior belief about the global factors before obtaining her measurement for attribute  $j$  and  $P(G_j|X_{ij})$  is her posterior belief after obtaining the measurement. The third random variable models the local factors that are personal to the agent and affect her attribute value. For every agent  $i$ , we model a set of other agents  $N_i \subset W$  ( $1 < |N_i| < |W|$ ), called cluster of agent  $i$  which share only these personal factors. Note that this is a much weaker modeling condition as compared to that of sharing personal measurements. Further, the clusters are unknown

<sup>1</sup>In the chapter, we use  $P(\cdot)$  for density functions to keep notations simple.

to the mechanism. The random variable for personal factors is denoted by  $L_{kj}$ ,  $k$  being the cluster to which agent  $i$  belongs. In the rest of the chapter, we will simply use notation  $L_{ij}$  for  $L_{kj}$  such that  $L_{ij}$  are equal for all  $i$  in the same cluster  $k$ . The  $P(L_{ij})$  is the agent's prior belief about the personal factors before taking measurement for attribute  $j$  and  $P(L_{ij}|X_{ij})$  is the posterior belief after taking measurement.  $L_{ij}$  and  $G_j$  are related through the conditional distribution  $P(L_{ij}|G_j)$ . It is easy to show that, in this model, the global distribution  $P(X_{ij}|G_j)$  can be modeled by a mixture distribution as follows:

$$P(X_{ij}|G_j) = \sum_{k=1}^K \alpha_k \cdot P(X_{ij}|L_{kj})$$

where  $K$  ( $\ll N$ ) is the number of distinct clusters in the population and  $\alpha_k$  is the mixing probability of  $k^{th}$  cluster. The model is summarized in Figure 3.2.

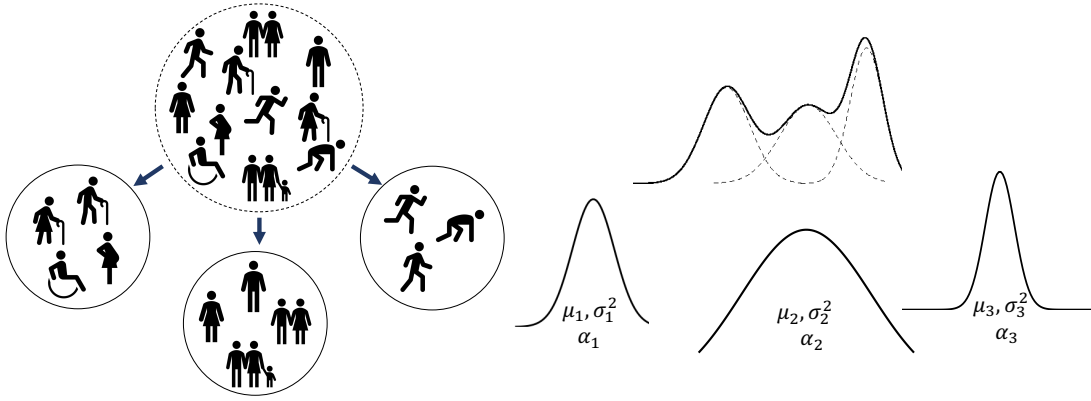


Figure 3.2: **(Belief Model)** *Left* - Agent population contains clusters of agents with similar characteristics. *Right* - Agents' measurements can be modeled using Gaussian mixture.

In this chapter, we will use [79] normal distribution<sup>2</sup> to model  $X_{ij}$ 's dependence on  $L_{ij}$ , i.e.,

$$P(X_{ij}|L_{ij}) = \mathcal{N}(\mu_{L_{ij}}, \sigma_{L_{ij}}^2)$$

### 3.3 The PPTS Mechanism

The center collects reports from all agents for all their attributes. It then assigns each agent to its corresponding cluster described in agent  $i$ 's belief. The cluster assignment step is discussed in Section 3.5. For now, let's assume this as an oracle that provides the mechanism with every

<sup>2</sup>This assumption simplifies the analysis of the mechanism and is not crucial for the main results presented in the chapter.

agent's **true** cluster label. We define the  $j^{th}$  attribute score of agent  $i$  for reporting  $X_{ij} = y$  as :

$$r_{ij} = \log \frac{f(y|\hat{\mu}_{L_{ij}}, \hat{\sigma}_{L_{ij}}^2)}{\sum_{k=1}^K \hat{\alpha}_k \cdot f(y|\hat{\mu}_{L_{kj}}, \hat{\sigma}_{L_{kj}}^2)} \quad (3.1)$$

where  $f$  is the Gaussian function given by

$$f(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$\hat{\mu}_{L_{ij}}$  and  $\hat{\sigma}_{L_{ij}}^2$  are the mean and variance of values reported for attribute  $j$  by agents in the cluster  $N_i$ .  $\hat{\alpha}_k$  is the empirical relative mixing frequency of cluster  $k$ .

Agent  $i$  finally gets a cumulative reward (CR) equal to the average of attribute scores  $r_{ij}$  for all attributes  $j \in \{1, 2, \dots, d\}$ . More formally,

$$CR(i) = \frac{\sum_{j=1}^d r_{ij}}{d}$$

*Example:* As an example for calculation of attribute scores, consider agent  $i$  who reports its wrist measurement as 4.5 units. The reported wrist measurements of agents in the cluster of agent  $i$  have mean 4 and s.d. 3. If there are 2 distinct clusters in the population, another with mean 5 and s.d. 5 (with equal mixing frequencies), then the wrist attribute score of agent  $i$  is given by:

$$\begin{aligned} r_{ij} &= \log \frac{f(4.5|4, 3^2)}{0.5 \cdot (f(4.5|4, 3^2) + f(4.5|5, 5^2))} \\ &\approx \log \frac{0.1311}{0.5 \cdot (0.1311 + 0.0793)} \approx 0.22 \end{aligned}$$

On the other hand, if the means and s.d. of reports in the cluster of  $i$  are 0 and 1 respectively, then wrist attribute score of  $i$  is

$$\begin{aligned} r_{ij} &= \log \frac{f(4.5|0, 1^2)}{0.5 \cdot (f(4.5|0, 1^2) + f(4.5|5, 5^2))} \\ &\approx \log \frac{0.00002}{0.5 \cdot (0.00002 + 0.0793)} \approx -8.2 \end{aligned}$$

### 3.4 Theoretical Analysis

Intuitively, the numerator of the fraction inside the logarithm in Equation 3.1 measures how common (likely) a report is in its cluster while the denominator measures how likely a report is globally. Thus, similar to the Bayesian Truth Serum, PPTS rewards ‘surprisingly common’

reports. In the following theorems, we formally discuss the incentive compatibility and other properties of the mechanism. For better understanding, we first discuss the theoretical properties treating the cluster assignment step as a black box oracle and show in Section 3.5 how the mechanism obtains the clusters while preserving incentive compatibility. Proofs are provided in the appendix.

We call a mechanism Bayes-Nash incentive compatible if truthful reporting is an equilibrium of the mechanism i.e. if other agents report their observations truthfully, no agent has an incentive to deviate from the truthful strategy for any observation of the agent. This is sometimes also called as the ex-post subjective equilibrium [113] since the beliefs of the agents are different (subjective).

**Theorem 4.** *The PPTS mechanism is Bayes-Nash incentive compatible, with strictly positive expected payoffs in the truthful reporting equilibrium.*

The theorem states that given other agents are truthful, it is the best strategy for any agent to be truthful. The sketch of our information-theoretic proof is that from many independent and identically distributed truthful observations of other agents, the mechanism obtains maximum likelihood estimates of the true global and personal factors. A simple application of Bayes rule then shows that the mechanism rewards a report for its informativeness in predicting the personal factors, and the reward is maximized and is strictly positive for a truthful report.

While a truthful equilibrium is a desired outcome, there are other (non-truthful) equilibria that the mechanism admits - which is a common feature in the peer-consistency methods. It is important to ensure that such equilibria are not more profitable than the truthful equilibrium. They include heuristic reporting strategy equilibria. As discussed in Section 3.2, in heuristic reporting strategy, agents save the effort of even making an observation and report a random sample drawn from a probability distribution.

**Theorem 5.** *Heuristic reporting equilibria result in zero expected payoff in the mechanism.*

This is because when agents draw independently from a random distribution, both local and global MLEs converge to common values and it results in a reward of  $\log 1 = 0$ .

There are also informed non truthful equilibria, where agents do take the measurements but use a mapping to transform their actual measurements  $x$  into their reports  $y$ . Consider linear transformation mappings, where agents use a function  $y = g(x) = ax + b$  to get their reports from their measurements  $x$ . In the real world, this strategy corresponds to agents systematically over reporting or under reporting their measurements.

**Theorem 6.** *In the PPTS mechanism, an equilibrium strategy profile defined by a function  $g(x) = ax + b$  is not in expectation more profitable than the truthful strategy.*

The proof uses the observation that if agents use linear transformation to report, the MLE estimates also change accordingly and reward remains unchanged. Such equilibria don't give higher expected reward but choosing same  $g$  requires a lot of coordination among the agents and hence are unlikely to be played. Agents unilaterally choosing a different linear  $g'$  get lower scores than if they stay with  $g$  as well and thus such profile is not in equilibrium.

Next, we look at the ex-ante expected score of a truthful agent i.e. expected score before taking the measurement.

**Theorem 7.** *The ex-ante expected score of a truthful agent is equal to the conditional mutual information (CMI) of the attribute measurements and the personal factors given the global factors.*

The CMI [19] is the expected value of the mutual information of two random variables given the value of a third, where the mutual information of two random variables measures the mutual dependence between two random variables. Since, CMI is always non-negative, the ex-ante expected score of a truthful agent is always non-negative. When the CMI is 0 i.e. when the attribute is independent of the personal factors, the mechanism can't be used to elicit truthful information because the expected payment is 0 regardless of the report. We discuss an interesting use of this theorem in further sections.

### 3.5 Clusters Approximation

A crucial step in the mechanism described in Section 3.3 was to assign every agent to its correct cluster. We now describe how the mechanism achieves this without affecting the incentive compatibility. In the absence of the oracle, naturally the only option available to the center is to use the reports of the agents themselves to approximate the clusters. However, the question is whether doing this is game theoretically sound and preserves incentive compatibility?

**Definition 12. ( $\epsilon$ -Correct Clustering Algorithm)** *A clustering algorithm is called  $\epsilon$ -correct, if given true reports, it assigns a true report to a wrong cluster with probability at most  $\epsilon$  and  $\epsilon$  is such that as  $|N_k| \rightarrow \infty$ , the MLE estimates  $\{\hat{\mu}_{kj}, \hat{\sigma}_{kj}^2\}$  converge to  $\{\mu_{kj}, \sigma_{kj}^2\}$  and  $\hat{\alpha}_k$  converge to  $\alpha_k$ ,  $\forall k$ .*

Note that the definition doesn't require every point to be assigned to correct clusters but only the approximated cluster parameters to converge to correct parameters. The conditions required for correct estimation of Gaussian mixture parameters from a finite sample are discussed in [58],[83]. The conditions include a lower bound on the mixing probabilities and the statistical distance between the cluster distributions. This implies that the more separated the clusters are, the better are the approximations of cluster parameters with fewer samples.

**Theorem 8.** *Given an  $\epsilon$ -correct clustering algorithm, the PPTS is Bayes-Nash incentive compatible even if the clusters are approximated from the reports.*

The main insight of this theorem is the following : the fact, that the mechanism doesn't know the cluster labels but instead uses an  $\varepsilon$ -correct clustering algorithm to cluster the reports of the agents, doesn't provide any agent with a more profitable non-truthful strategy to deviate from the truthful equilibrium. This result addresses the concern that agents may strategically manipulate their report to get assigned to a different cluster and get a better reward. Hence, an  $\varepsilon$ -correct clustering algorithm can be applied to assign the clusters while preserving incentive compatibility.

#### Implementation and Practical Considerations

In this chapter, we implement the *PPTS* mechanism by using the following technique to approximate the clusters. Consider approximating the cluster for calculation of the  $j^{th}$  attribute score of the agents. Let  $A_{-j}$  be the set of all attributes excluding attribute  $j$  i.e.  $A_{-j} = A \setminus \{j\}$ . We then apply  $k$ -means clustering algorithm on attribute sets  $A_{-j}$  to obtain the clusters used in calculating of the  $j^{th}$  attribute score.

It remains to discuss how one can judge if the clusters found using the above technique are indeed fit for being used with the *PPTS* mechanism in practice. For this, we make use of Theorem 7. The theorem says that if the conditional mutual information  $I(X_{ij}; L_{ij} | G_j)$  is close to 0, then the mechanism can't be used for truthful elicitation. If some trusted prior data (i.e. some true observations  $X_{ij}$ ) is available to the center for analysis, CMI estimators [108],[109] can be used to estimate  $I(X_{ij}; L_{ij} | G_j)$  by using  $\hat{\mu}_{L_{ij}}$  from the approximated clusters in place of  $L_{ij}$ . A low value of this CMI estimate suggests the unsuitability of the clusters for the mechanism. In the next section, we demonstrate this method on real datasets. To understand this in a more intuitive manner, recall that we use attribute set  $A_{-j}$  for approximating the clusters. If all attribute pairs are independent, observations for attribute  $j$  will be independent of the cluster approximated using  $A_{-j}$ , which means that the estimated clusters can't be used with the mechanism. Therefore, to find suitable clusters, we need to elicit interdependent attributes.

### 3.6 Nearest Neighbors Scheme

Before we move to the empirical evaluation of the *PPTS* scheme, we discuss an alternate reward scheme motivated by the *PPTS*. We call this scheme the *PPTS-knn* scheme. In this alternate reward scheme, we don't assume that the true data has separate clusters but for every worker, we assume only the existence of many other workers who are "similar" to her in their personal factors. Because of this slightly different assumption on the neighborhood structure, this scheme approximates the neighborhood and the global distribution of the measurement values in a slightly different manner. The main idea of this scheme is described in Figure 3.3.

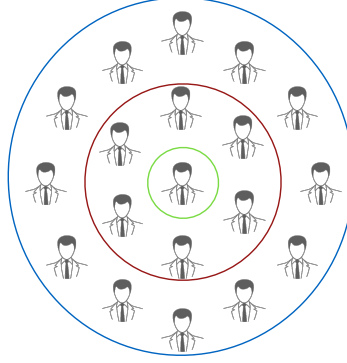


Figure 3.3: **(PPTS-knn and Belief model)** For a worker  $i$  (at the center), there exists a neighborhood of workers who are similar to it in their personal characteristics (middle ring) and thus distinguished from rest of the workers (outer ring). The worker is rewarded for its report being more likely in the neighborhood than overall.

In this scheme, we define the attribute score of worker  $i$  as :

$$r_{ij} = \log \frac{f(y|\hat{\mu}_{L_{ij}}, \hat{\sigma}_{L_{ij}}^2)}{f(y|\hat{\mu}_{G_j}, \hat{\sigma}_{G_j}^2)} \quad (3.2)$$

where  $f$  is the Gaussian function given by

$$f(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$\hat{\mu}_{l_{ij}}$  and  $\hat{\sigma}_{l_{ij}}^2$  are the mean and variance of values reported for attribute  $j$  by workers in the neighborhood  $N_i$ .  $\hat{\mu}_{g_j}$  and  $\hat{\sigma}_{g_j}$  are the mean and variance of values reported for attribute  $j$  by all workers in  $W$ .

The cumulative reward is defined in terms of these attribute scores in the same way as described in original *PPTS* i.e. the average of the attribute scores.

The neighborhoods in this scheme are approximated using  $k$ -nearest neighbor technique instead of clustering. More precisely, while calculating the attribute score for attribute  $j$ ,  $k$ -workers who have minimum distance from the worker  $i$  are chosen as her neighborhood. The distance between workers is calculated using euclidean distance between the attribute vectors  $A_{-j}$  of the workers.

The basic idea of this scheme is also based on the same “surprisingly common” principle. One can view the statistical model in this approach as a Gaussian process model. The observation of every worker is modeled by a gaussian distribution with a different mean and variance. The mean and variance for each, are approximated from the reports of the  $k$ -nearest neighbors of the worker. However, we conjecture that a formal proof of incentive compatibility of this

Dataset	CMI Estimate
Body Measurements	0.41559387
Air Quality	0.98769209
Seed	0.98322659
Census Income	0.0194241

Table 3.1: Average CMI estimates for different datasets

scheme would require stronger assumptions on the beliefs of the workers. Nevertheless, we discuss the empirical performance of this scheme also in the chapter.

### 3.7 Experimental Evaluation

A real world validation of our mechanism by using it to collect new personal data is perhaps not feasible in the absence of ground truth for performance evaluation. However, the manipulation resistant properties of the mechanism can be best verified through simulations on real datasets. We simulate, on three real datasets, the strategies that agents may adopt and discuss the rewards that our mechanism decides for them.

#### 3.7.1 Datasets

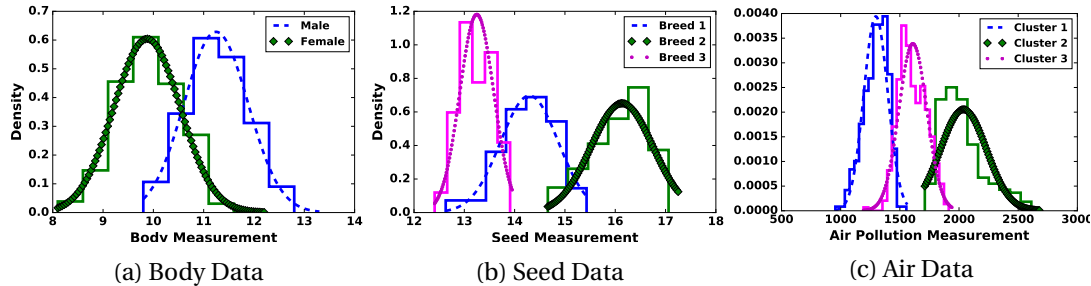


Figure 3.4: Cluster Distribution in Datasets

We selected three datasets from different domains for evaluating the mechanism through simulations. The *Body Measurements* [43] dataset contains 21 body dimension measurements as well as age, weight, height, and gender of 507 individuals. The 247 men and 260 women were mainly young adults, with a few older men and women. The *Seed* [14] dataset consists of 7 measurements of 210 seeds of wheat. It has 70 samples each of three varieties of seeds (with labels). The *Air Quality* [23] dataset consists of 9358 instances (852 complete instances) of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an air quality multi-sensor device. The *Air Quality* dataset was not collected at different places



but at a single place at different times. Another dataset that we considered for evaluation was extracted from U.S. 1994 census data. This *Census Income* [67] has 15 personal information attributes (continuous and categorical) about the population such as salary class, education level, working hours, native country, age, sex, race, occupation etc. For simulations, we can assume each instance (row) in a given dataset to be reported by a different agent and each instance having multiple attributes. For example, in the *Air Quality* dataset, an instance has 5 attributes corresponding to the 5 metal oxide sensors. The datasets act as true private observations of agents. In *Seed* and *Body* datasets, clusters capture similarity between different individuals and seeds. In the *Air Quality* dataset, clusters capture the temporal similarity between pollution measurements. As datasets with more personal attributes are hardly available publicly, these public datasets do a good job at simulating the task settings we target i.e. elicitation of continuous valued unique personal attributes with normal like distribution. Figure 3.4 shows one attribute each in the *Body Measurements* dataset, the *Seed* dataset and the *Air* dataset along with their normal approximations in the clusters. The *body* and *seed* datasets are labeled but labels are used only for visualization and not for other experiments reported in the chapter. *Air* dataset is unlabeled, hence we used our approximated clusters for visualization also.

#### 3.7.2 Cluster Fitness Evaluation

To evaluate the fitness of the clusters approximated by the  $k$ -means algorithm on these datasets, we make use of Theorem 7. The average CMI estimates (of all attributes) from the four datasets are shown in Table 3.1. Note that for *Census Income*, the average CMI estimate is very small (close to 0). Hence, we can not use the clusters for eliciting the attributes of this dataset for reasons explained in Section 3.5.

### Results

For better understanding, we present the results in two parts - attribute scores in Section 3.7.3 and cumulative rewards in Section 3.7.4. We will be discussing the following statistics of scores/rewards - mean (average of scores/rewards), Q1 (1<sup>st</sup> quartile of scores/rewards), Q2 (2<sup>nd</sup> quartile), Q3 (3<sup>rd</sup> quartile) and  $F$  (fraction of agents receiving strictly positive score/reward), under different simulated strategies.

#### 3.7.3 Attribute Score

We simulate the following reporting strategies that can be used by agents :

1. TR - All agents report all attributes truthfully.
2. RA - All agents report  $j^{th}$  attribute randomly within its true range and all other attributes truthfully.

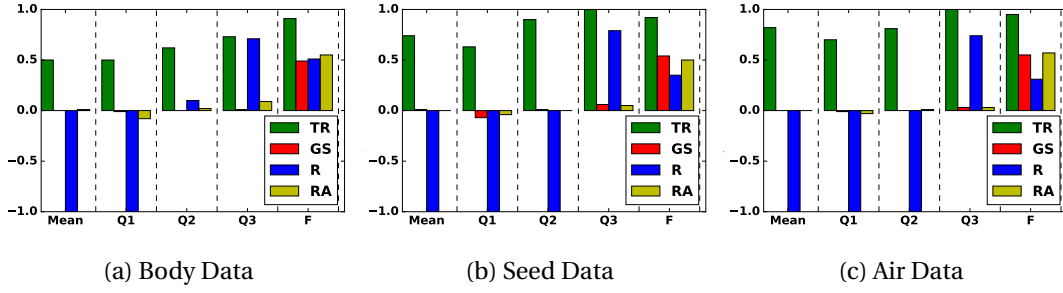


Figure 3.5: Statistics of Attribute Scores

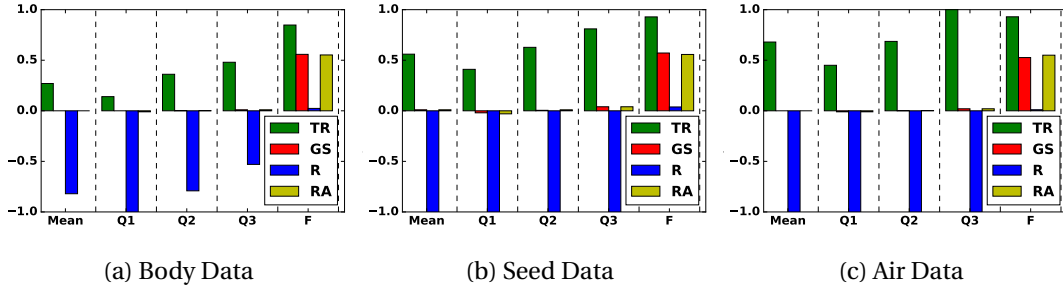


Figure 3.6: Statistics of Cumulative Rewards (Average of attribute scores of all attributes)

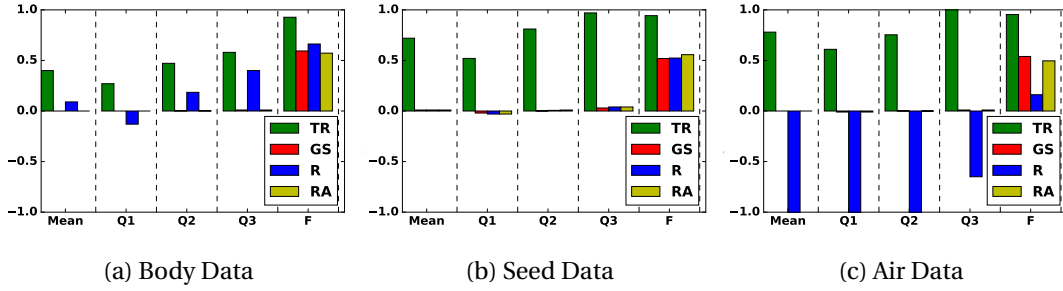


Figure 3.7: Statistics of Cumulative Rewards (Median of attribute scores of all attributes)

3. R - All agents report all attributes truthfully except agent  $i$ , who reports  $j^{th}$  attribute randomly within its true range but other attributes truthfully.
4. GS - All agents collude to report  $j^{th}$  attribute using a Gaussian distribution with true mean and variance of the attribute and report all other attributes truthfully.

Figures 3.5a, 3.5b and 3.5c show statistics of attribute scores for  $j^{th}$  attribute in each dataset under different reporting strategies of agents. This  $j^{th}$  attribute is 'height' in the *Body Measurements* data, 'kernel length' in the *Seed* data and 'PT08.S2' in the *Air Quality* data. Figure 3.5b shows results for the *Seed Measurements* data. The first important point to note is that the fraction of agents getting strictly positive score is more than 0.92 when agents report truthfully but hardly goes above 0.5 in other non-truthful strategies, which means that non-truthful strategic agents do no better in expectation than a random guesser. The other thing to note is that the mean score when agents are non-truthful is not positive, whereas for truthful agents, it is strictly positive with sufficient value to distinguish it from a 0 score. A similar trend can be observed for other statistics such as Q1, Q2 and Q3, where the score

for truthful reporting is always greater than that for non-truthful strategies. In particular, we can observe that  $Q2$  (i.e. the median) is also strictly positive for truthful agents and not more than 0 for non-truthful agents. Similar results can be seen in Figures 3.5a and 3.5c for *Body* and *Air* datasets respectively. It is worth mentioning here that the scores can be appropriately scaled to cover the cost of participation and satisfying budget constraints without affecting the incentive-compatibility of the mechanism.

Also to confirm our earlier conclusion of the clusters not being useful for the *Census Income* dataset, we computed the rewards of agents for reporting this data truthfully and found that only about 32% of the agents get positive score with mean score approaching 0.

#### 3.7.4 Cumulative Reward

Here, we report simulation results for the following reporting strategies :

1. TR - All agents report all attributes truthfully.
2. RA - All agents report all attributes randomly within true ranges of respective attributes.
3. R - All agents report all attributes truthfully except agent  $i$ , who reports all attributes randomly within true ranges of respective attributes.
4. GS - All agents collude to report all attributes using Gaussian distributions with true means and variances of respective attributes.

In section 3.3, we defined the cumulative reward of a agent as the average of all attribute scores of this agent. Figures 3.6a, 3.6b and 3.6c show statistics of final or cumulative rewards. Figure 3.6b shows the results for *Seed* data. Similar to attribute scores discussed in Section 3.7.3, the fraction of agents with strictly positive cumulative reward is 0.93 when they report truthfully and is hardly more than 0.5 when they report non-truthfully. The mean cumulative reward for truthful reporting strategy is strictly positive and is not more than 0 for non-truthful strategies, attesting Theorem 4 and Theorem 5. In Figures 3.7a, 3.7b and 3.7c, we show statistics of cumulative rewards calculated as the *median* of the attribute scores instead of average of attribute scores, i.e.,

$$CR(i) = \text{median}_{j \in \{1 \dots d\}} \{r_{ij}\}$$

The median is another way to calculate CR from attribute scores and makes it robust to outliers in attribute scores. We also find the median to perform better in simulations as it makes the minimum reward of truthful agents non-negative.

#### PPTS-knn Scheme

Using the *PPTS-knn* Scheme also gives similar or better results for rewards. The statistics of the cumulative rewards for the three datasets are shown in Figures 3.8a, 3.8a and 3.8c

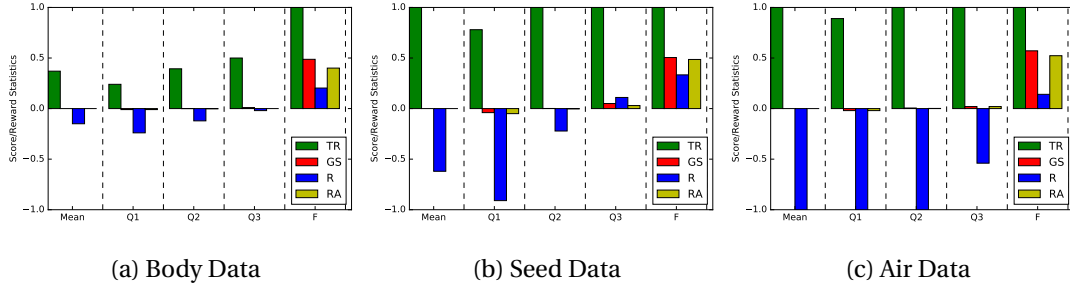


Figure 3.8: Statistics of Cumulative Rewards (Median of attribute scores of all attributes)  $k$ -nn approach

respectively. We can see that on all three datasets, this scheme is able to reward almost 100% of the truthful workers with positive reward, while still nicely penalizing the non-truthful ones. This improved performance can be explained by the fact that that nearest neighbor based neighborhood approximation is able to better capture the personal factors of the workers than a clustering algorithm. Because of improved correlation between reports and the neighborhoods, the empirical performance of the scheme improves. This observation is in agreement with Theorem 7

### 3.8 Application to Data Cleaning

When it is not possible to provide incentives or if the data has already been collected without providing incentives, the *PPTS* scheme can still be used for data cleaning. In general, there is a fundamental difference between reward schemes and data cleaning schemes. While reward schemes have to reward truthful workers only in expectation to get high quality data, the data cleaning approaches can only be applied once data has been collected and have to be very precise to avoid throwing away truthful data or keeping non-truthful data. The data cleaning approaches are also in general susceptible to strategic data sources and can not really add much value to mostly incorrect data sample. Similarly, the reward schemes are also susceptible to unexpected non-truthful behavior of some fraction of workers. In an ideal system, both the schemes should be used in parallel to get the best quality data. Though data cleaning is not the focus of this chapter, the high accuracy of our reward scheme on real data naturally motivates the use of cumulative reward for also cleaning the collected data by truthfulness criteria. In this section, we experimentally evaluate the use of calculated cumulative reward to filter out the synthetically inserted “non-truthful” data points. Our experimental results show a performance better than state-of-the-art implementations of the many popular outlier detection algorithms for cleaning the data of this nature. We perform experiments on both synthetic datasets and real datasets. Synthetic datasets were generated by sampling from a mixture of gaussians with randomly generated means in a given range, covariance matrices and mixing probabilities. For the purpose of experiments, we replace a fraction of data points in the dataset with incorrect data points. The incorrect data points were generated randomly by sampling each attribute from a uniform distribution in  $(\mu - \sigma, \mu + \sigma)$ , where  $\mu$  and  $\sigma$  are the mean and s.d. of the true observations of the overall worker population.

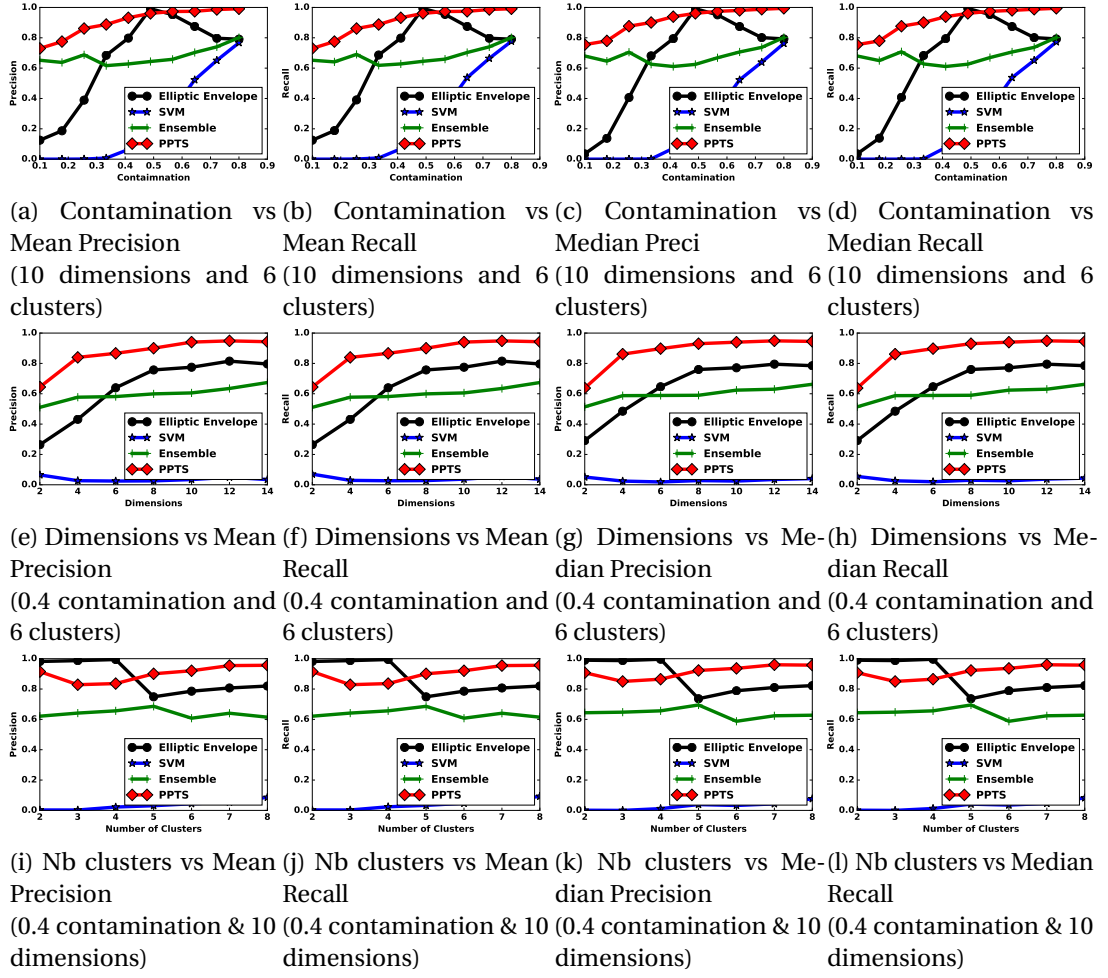


Figure 3.9: Performance comparison of data cleaning approaches on synthetic datasets

For cleaning the data, we propose the following steps :

1. Collect the multi-attribute reports from workers.
2. Cluster the collected reports using  $k$  means or EM for gaussian mixture.
3. Calculate the cumulative reward of each of the collected data points using the *PPTS* scheme described earlier.
4. Given information about the level of contamination (fraction of incorrect data points), remove that fraction of data points with least cumulative rewards.

We compare this data cleaning approach with commonly used state-of-the-art implementations of the one-class SVM based approach [103], an ensemble approach based on the isolation forests [74] and a robust covariance estimate based method called the elliptic en-

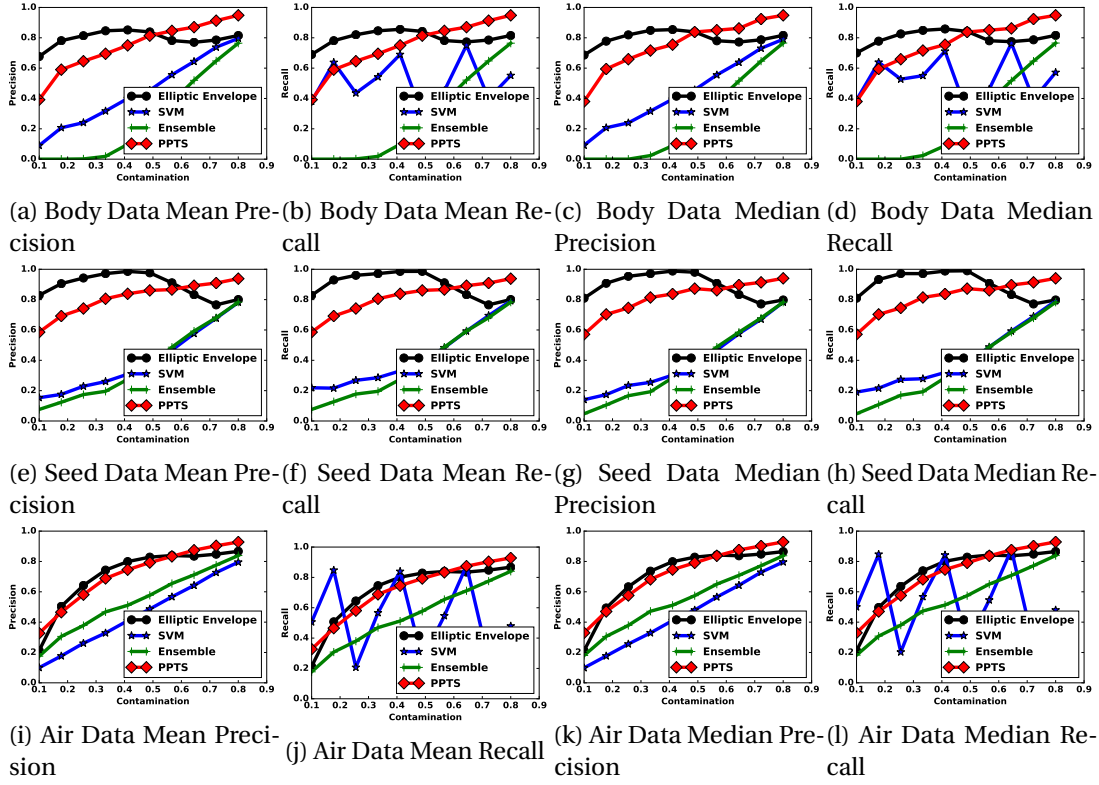
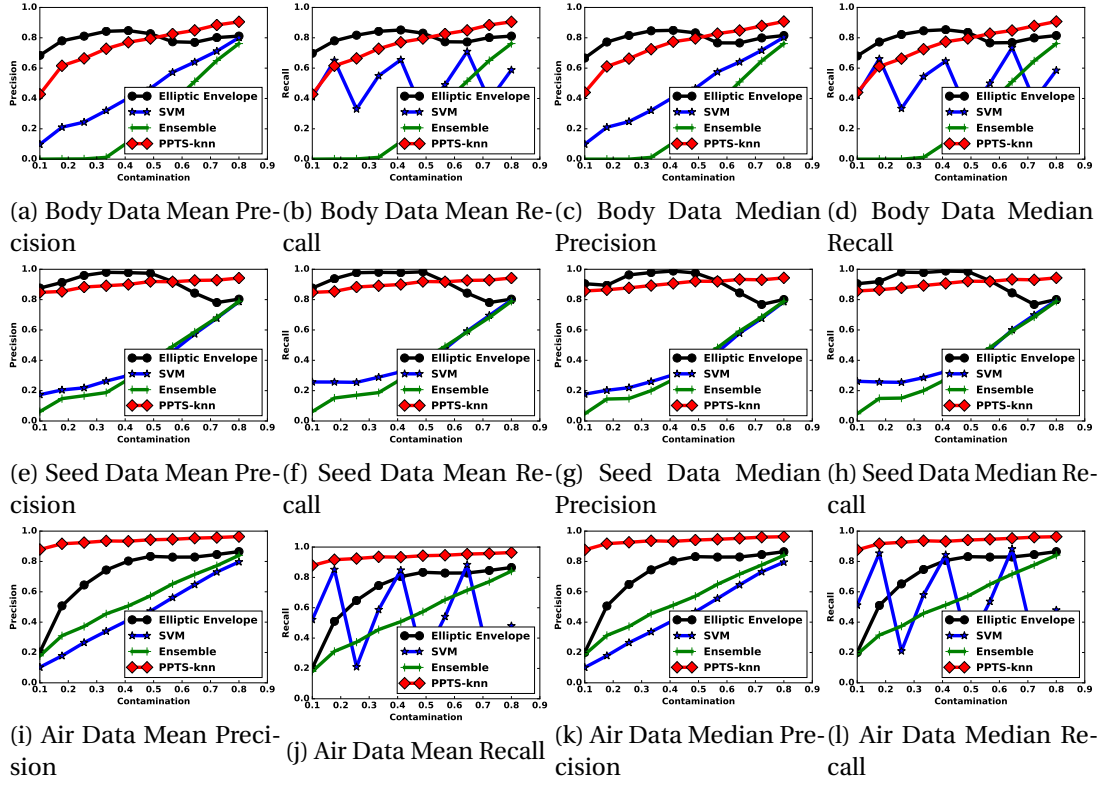


Figure 3.10: Performance comparison of data cleaning approaches on real datasets.

velop technique [99]. We used scikit-learn’s [91] implementations<sup>3</sup> of the above methods in our study. The results of the experiments with synthetic datasets are shown in Figure 3.9. Due to the randomized nature of the experiments, we performed each of the experiments 10 times and report the aggregate (mean and median) precision and recall of different approaches. The figures show the conditions in which our method provides superior performance as compared to all other baseline methods, at the same time guaranteeing that the filtering is, in expectation, on truthfulness criteria and not on outlier criteria. In Figures 3.9a, 3.9b, 3.9c, 3.9d, we can see that keeping the number of dimensions and clusters in the data constant, our method consistently outperforms other methods as level of contamination in the data continues to increase. In Figures 3.9e, 3.9f, 3.9g, 3.9h, we can see the effect of number of dimensions, keeping contamination level and number of clusters constant. Our method continues to beat all methods but the elliptic envelope method is competitive for large number of dimensions. In Figures 3.9i, 3.9j, 3.9k, 3.9l, we see the effect of number of clusters, keeping the number of dimensions and contamination level constant. While the elliptic envelope method performs better with low number of clusters, our method is the preferred choice for highly clustered data.

The results of the experiments with the three real datasets (*Body*, *Seed* and *Air*) are shown in Figure 3.10, further attesting the empirical performance our method. Since the datasets have

<sup>3</sup>[http://scikit-learn.org/stable/modules/outlier\\_detection.html](http://scikit-learn.org/stable/modules/outlier_detection.html)

Figure 3.11: Performance comparison of data cleaning approaches on real datasets ( $k$ -nn approach).

low number of clusters (2 – 3), the results are in agreement with what we observed in synthetic experiments. Elliptic envelope method gives a competitive performance while other methods being inferior.

**Remark:** The SVM based outlier detection method seems to show a zig-zag recall curve for different values of contamination. While the reason for this is not entirely clear, it is most certainly not a bug in our experiments, given that the experiment code is re-used for generating results for all other methods as well. Moreover, the same experiments generated the corresponding precision and recall data points for SVM, and the precision curves don't show any zig-zag behavior.

### PPTS-knn Scheme

Similar to *PPTS* scheme, one can also use the *PPTS-knn* reward scheme for data cleaning. The fraction of data points, equal to the contamination level in the data, which have the lowest cumulative reward according the *PPTS-knn* scheme are filtered out. As it was the case with cumulative reward statistics discussed in Section 3.7.4, we observe a similar performance boost for data cleaning task also by using the  $k$ -nearest neighbor based reward scheme. The performance comparison of this reward scheme with other baseline methods is shown in Figures 3.11 on all three real datasets. Looking at Figures 3.11 and Figures 3.10, we can see the

difference in filtering accuracy that the  $knn$  based scheme makes. The difference is particularly more significant in the *Air* dataset, where we see upto 200% performance boost. The reason of this performance improvement, as explained earlier, can be attributed to the learning of better local structure in the data by  $k$ -nearest neighbors.

## 3.9 Chapter Summary

In this chapter, we investigated the problem of incentivizing agents to honestly report their personal attributes such as physiological measurements. We distinguish this problem from the problem of incentivizing agents where multiple agents can solve a common task such as labeling a common image. We thus extend the applicability of the peer based incentive mechanisms from discrete labels for shared objects to real valued multi-dimensional personal features. We propose the *Personalized Peer Truth Serum (PPTS)* to address the problem. The PPTS shows desired properties by making the honest reporting equilibrium more profitable than heuristic reporting equilibria. We further investigate the problem of finding peer agents against whom the report of an agent is to be evaluated and propose to exploit other reports of the agent to estimate its peers. We guarantee that the incentive compatibility of the mechanism continues to hold while doing so. We provide a theoretically sound practical test to determine the applicability of PPTS for a given set of attributes by estimating the ex-ante expected payment. We empirically analyze the performance of PPTS using estimated peers on real datasets. The PPTS is able to incentivize/penalize simulated honest and heuristic reporting strategies with a good accuracy. We further discussed the application of reward score for data cleaning and empirically compared its performance on 3 real datasets against several popular baseline methods. We thus provide an end-to-end solution to acquire high quality data through incentives and enriching the quality of acquired data through data cleaning. We envision the deployment of such systems, merging the complementary approaches of incentives and cleaning, to become crucial in future to crowdsource high quality data for machine learning tasks.



## 4 Infochain: A Decentralized, Trustless and Transparent Oracle on Blockchain

This chapter is based on the following publications:

- N Goel, A Filos-Ratsikas, B Faltings. **Peer-Prediction in the Presence of Outcome Dependent Lying Incentives.** In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.
- N Goel\*, C van Schreven\*, A Filos-Ratsikas, B Faltings. **Infochain: A Decentralized, Trustless and Transparent Oracle on Blockchain.** In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2020. AI in FinTech Special Track.

### 4.1 Introduction

With the increasing popularity of the blockchain technology, the implementation of commercial and governmental systems has witnessed a large shift towards distributed and decentralized approaches. In particular, the emergence of the Ethereum platform has given rise to the development of several applications, often referred to as *decentralized apps* or *DAPs*, which aim to apply this latter principle to many areas such as finance, education, intellectual property or government. At the heart of these approaches lies the concept of the *smart contract*, i.e., lines of code that contain the terms of the agreement between the involved parties, which are automatically executed once triggered by events happening in the real world. For example, consider the case of a web service, which is typically dictated by a service level agreement (SLA) between the service provider and the clients. The SLA can be coded into a smart contract between the involved parties which will trigger an automatic payment upon detection of a violation. For instance, if the service guarantees a response time of at most 1 second with high probability, frequent slower responses would trigger automatic compensation. An important issue here is, *how to determine whether the real-world event has actually happened*. In the above example, this means how to determine that the SLA has been violated? We use the case of the web service only as a simple running example but this is in fact a *fundamental challenge*

*in developing information infrastructure for FinTech.*

The need for trusted information about a real-world event that triggers some conditional financial transactions arises in applications ranging from insurance, banking, trade, governance and law etc. The entities responsible for acquiring such data about the real-world events are called *oracles*. Existing solutions include Town Crier and Chainlink among others. Traditionally, oracles are implemented using trusted third party data sources responsible for acquiring the information. However, besides the fact that such an approach is in conflict with the decentralized nature of the blockchain technology, it is also prone to problems such as trustworthiness and cost.

An alternative solution would be to appeal to the “wisdom of the crowds” and ask the users themselves about the information (for e.g., the quality of service received). The idea has also been proposed for outcome resolution in decentralized prediction markets like Augur and Gnosis. While this approach is more decentralized in nature, it poses a significant challenge: the agents can not be relied on to provide correct information. The task of eliciting information from self-interested agents is one of the fundamental problems in game theory, and has been extensively studied. In the center of these investigations lies the literature on peer-consistency mechanisms [30]; these are game-theoretic mechanisms that incentivize agents to report the information truthfully, even if the information is unverifiable.

The literature on peer-consistency [30]) is quite rich. This literature includes solutions that are guaranteed to incentivize truth-telling even when there is a *cost of effort* for forming an informed report [97], but it does not address settings in which agents have other incentives dependent on the aggregate feedback. Such cases arise frequently, for example, in decentralized QoS monitoring, environmental data collection and surveys that inform policy-making.

As a concrete example, consider the case of a web service which is typically dictated by a service level agreement (SLA) between the service provider and the clients, where the agreement dictates that the client will be compensated if there is a violation. Traditionally, a trusted third party monitors the quality of service (QoS) and sends the reports to a regulatory authority (or the service provider itself for self-regulation). Depending on the collected reports from the trusted party and the conditions of the SLA, users are issued refund. Not only this traditional approach is costly due to the high cost of hiring a commercial third party but it is also not a transparent and decentralized approach from the users’ perspective. On the other hand, if we were to decide whether such a violation actually took place based on the feedback from the clients themselves, it is clear that the clients would have incentives to report a violation regardless, in order to be compensated. To get truthful reports from the users, their incentives must be aligned with truthful behavior by using a game-theoretic mechanism. The main question here is, can we still use a peer-consistency mechanisms to counter-act this type of *outside* incentives?

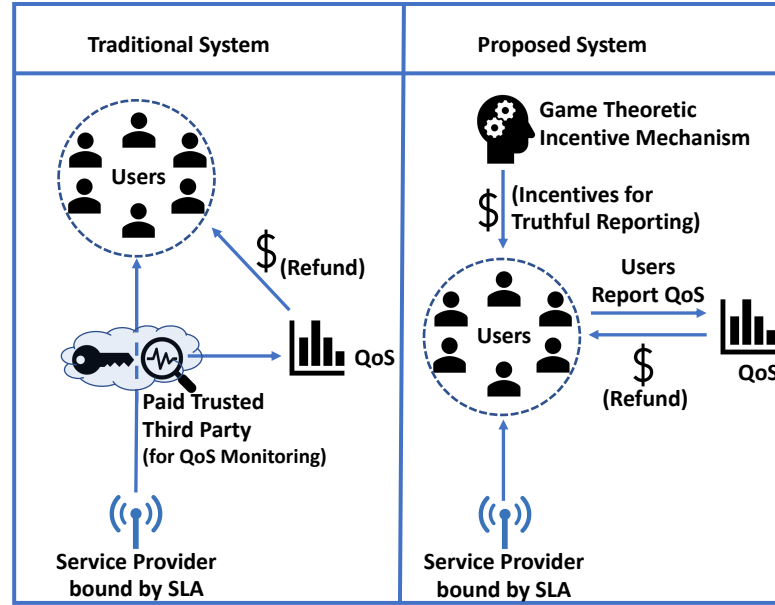


Figure 4.1: QoS Monitoring: A Motivating Example of Outcome Dependent Lying Incentives. Replacing trusted third party with the users themselves creates lying incentives for the users to misreport the true QoS received from the service provider. A game-theoretic mechanism incentivizes the users to report truthfully. The game-theoretic mechanism itself can be implemented by a regulatory authority in a completely decentralized, trustless and transparent manner by using the blockchain technology. A smart contract can automatically process refunds based on the crowdsourced outcome and the conditions of the SLA.

#### 4.1.1 Related Work

Many decentralized systems have been proposed for crowdsourcing and information trading [6, 72, 78, 116] but none addresses the issue of providing quality based incentives for information. In a recent and independent work, [68] also use peer-consistency for *trust-free* data trading systems but the analysis in this theoretical paper focuses on a secure multi-party computation protocol for rewarding information that loses value if revealed. [4] propose a system for a decentralized oracle but it requires a “random assignment” of questions to agents, which has the drawback that agents may be asked to answer questions that they may have no information about.

The topic of outside incentives in decentralized platforms has been recently explored in the context of prediction markets, drawing motivation from applications like Augur and Gnosis. [12] perform equilibrium analysis when the market participants may significantly influence the actual realization of the outcome, in a game which is played in two stages; first the agents trade in the market and then they vote on the outcome. Their model captures the empirical observations in prior work [13]. These works however only analyze the effects of rational behavior, rather than aim to counteract it, by implementing appropriate mechanisms. [16] consider similar two-stage models of prediction markets, where the agents strategize only in the first stage to manipulate the market prices used for the predictions. The authors analyze

information aggregation properties of the market and don't consider outcome manipulation, and their setting is thus quite different from ours.

[32] study a related setting, where they assume that agents trade honestly in the first stage and only behave strategically in the second. They use a peer-consistency mechanism to elicit truthful votes in the equilibrium of the second stage, and show that under certain conditions, the fees charged by the market are enough to cover the side payments. Interestingly, they also use a similar measure of signal correlation, which they refer to as the “update strength”, and they express some of their results using this quantity.

Our work differs from [32] in two key aspects. First, our informational assumptions are weaker. In particular, we only require access to a measure of signal correlation (the self-predictor value) and actually, only an estimate of that measure is sufficient. In contrast, [32] use the prior distribution of the agents' beliefs, which they obtain from the closing price of the market, enabled by the assumption that the agents are honest in the trading stage. While this may be meaningful in a prediction market domain, such assumptions are far less realistic in the more general settings that we consider. Secondly, [32] do not address the issue of non-truthful equilibria in their work. In settings other than prediction markets, [57] consider QoS feedback elicitation but assume full knowledge of agents' beliefs and only constant lying incentives.

Finally, we remark that, as we mentioned earlier, [97] have already shown that the PTSC mechanism can be tuned to overcome the cost of effort that the agents might have for coming up with their observations, which is a constant quantity. This is therefore markedly different from the case of outside incentives, where the “cost” that needs to be overcome crucially depends on the reports of the other agents.

Our work draws on the recent ideas in the peer-consistency literature [5, 20, 33, 60, 69, 76, 95, 96, 97, 104, 110]; here we focus on the results related to settings with outside incentives. A survey of the techniques in this area can be found in [30].

### 4.1.2 Our Contributions

We propose the employment of peer-consistency mechanisms for the design of trustless, decentralized oracles. However, this quest imposes two major challenges:

- In many financial settings, agents also have incentives to lie about their true observations and provide false information. In the web service example, the clients would have an incentive to always report “bad” response times, in order for the conditions of the smart contract to be violated in their favor. *How large do the incentives have to be, to counteract the lying incentives, and is the approach economically feasible?*
- For a peer-consistency mechanism to actually work, the agents must be convinced of its incentive properties, contrary to the traditional case, where the implementation of the mechanism is done by a trusted third-party. *How can one implement the incentive*

*scheme in a transparent and trustless manner, what is the cost and how can we optimize this cost?*

In this chapter, we address the above questions. We summarize our contributions below.

1. We formally analyze the settings when agents have outside incentives to lie.
  - We employ the PTSC mechanism of [97] as a side-payment scheme. We prove that, with an appropriate choice of the scaling constant, the mechanism can be used to ensure that truth-telling is a strict equilibrium of the induced game. This is the first result of this nature, that shows that a peer-consistency mechanism can be applied to the case of outcome-dependent incentives in a general information elicitation scenario.
  - Furthermore, we show that if there is *any positive* fraction  $f$  of *honest* agents (i.e., agents that always report truthfully), the strategy profile in which the agents exercise their outside incentives, or *denial strategies* (e.g., reporting “bad” service) is no longer an equilibrium. Assuming the *existence* of honest agents is very different from using trusted authorities since our method does not depend on knowing who these honest agents are. This is a rather common scenario, as in a large platform, one would normally expect at least a few agents to behave honestly but we would not expect to know their identities. These properties of the PTSC mechanism were already known in the *absence* of the outside lying incentives [97]; our work extends the analysis of PTSC in the *presence* of the outside incentives.
  - Additionally, for the first time, we compute a bound on the scaling required for ensuring a truth-telling equilibrium of the side-payment scheme, as a function of the outside incentives. We also provide conditions under which the side-payment scheme gives positive saving compared to the rational outcome (i.e., the denial strategy outcome) and we prove a lower bound on this saving. We show that as the number of agents grows large, the saving approaches the best possible saving, attainable when all agents are honest, without any side-payments. We also provide bounds (on the same quantities) when one has to not only ensure a truthful equilibrium, but also eliminate the denial strategy equilibrium. Interestingly, in the process of doing this, we find an upper bound on the fraction of honest agents that should be present, in order for the side-payment scheme to still be profitable.
  - Finally, the scaling constant, as well as the savings of PTSC depend on a quantity  $\delta^*$ , which we refer to as the *self-predictor value* and is essentially a measure of correlation strength between prior and posterior signals. The assumption that  $\delta^* > 0$  is a standard assumption in the literature of peer-consistency (e.g. see [54], [113]) and translates to positive correlation between the observations of the agents. We quantify the required scaling constant as well as the saving in terms of this quantity. Moreover, we do not need to know this quantity; an *estimate* is sufficient

for the results to either hold exactly or *approximately*, where the approximation error goes to 0 as the number of agents grows large.

2. We design and implement INFOCHAIN, a completely decentralized peer-consistency based truthful information collection system in Ethereum. We address the following technical challenges in its implementation.
  - Writing data and performing computation on Ethereum's Virtual Machine (EVM) is expensive. Information providers must be compensated for this cost, increasing the overall cost of information acquisition. For the first time, we discuss several non-trivial ways of implementing three different peer-consistency mechanisms in Solidity (Ethereum's programming language) and empirically compare their costs.
  - While transparency is a desired inherent feature of blockchain, the peer-consistency mechanisms are compromised if an agent can see the information submitted by their "peers" before submitting their own information. We propose to use a *commit-reveal protocol* to address this challenge.
  - In order to reduce computation complexity, peer-consistency mechanisms use only one (or a few) randomly selected peer(s) for every agent. However, if the random peer(s) can be predicted, the agents get an opportunity to collude and the mechanisms can be compromised, and this risk is increased by the transparency of blockchain. We show that under reasonable assumptions, random peer selection can still be implemented safely.

### 4.2 Model and Objectives

We consider settings in which questions are to be resolved on a decentralized platform through acquiring feedback from agents. No agent, however, is required to answer more than one question. The questions can be, for example, of the following form : "Is the responseTime of web service  $W$  less than 10 seconds?". An agent  $i$  makes a private binary observation  $X_i \in \{0, 1\}$  about a question and submits her feedback report  $Y_i \in \{0, 1\}$  to the platform. For any question,  $n$  agents are asked to submit their feedback, and based on this feedback, the questions are said to be resolved by announcing their outcomes. The *outcome*  $o_w$  for a question  $w$  is defined as the fraction of agents who reported 0 as their feedback. In the web service example, this corresponds to the fraction of agents who report that the responseTime of the service was not less than 10 seconds.

Note that we define the outcome  $o_w$  to be a continuous variable, whereas the feedback is elicited as a discrete variable. This is because of the noisy (and in some cases subjective) nature of the feedback. In the web-service case, responseTime is a noisy measurement and no service can promise a certain response time 100% of the time. Thus, it is important to define the outcome as a continuous variable measuring the fraction of time that the service did provide a good response time. We remark here that more generally, the outcome can be

defined as any non-negative, non-decreasing function of the fraction of agents who report 0 (e.g., a *threshold* function that becomes 1 if, say, 70% of the agents report 0). We choose the fraction of dissatisfied agents as our outcome function, for the reasons mentioned above, and also following the related literature [32].

The main novelty of our setting is that the agents receive an outside incentive that is dependent on this aggregate outcome. More precisely, the payment given to an agent is  $\mathcal{R} \cdot o_w$ , where  $\mathcal{R}$  is a positive constant. In the web service example, such payments might arise through the service level agreements between the web service provider and the agents<sup>1</sup>. The focus of this chapter is how to adapt the incentives given for the reports to overcome such lying incentives.

After making her private observation, agent  $i$  uses a strategy  $\sigma_i$  to submit a report  $Y_i$  based on observation  $X_i$ , in order to maximize her expected payment. The agents are assumed to be *rational* and therefore they may not report their true observations, if not properly incentivized to do so. We follow the common assumptions that agents are *risk-neutral* and that the utilities are *non-transferable*.

**Definition 13** (Agent Strategy  $\sigma_i$ ). *An agent  $i$ 's strategy, denoted by  $\sigma_i(Y_i = y|X_i = x), \forall x, y \in \{0, 1\}$ , is the probability of the agent's report for the question being  $y$  given that her observation is  $x$ .*

The strategy models a variety of possibilities that are available to the agent for mapping her observation to report. Some examples are as follows:

**Definition 14** (Truth-telling Strategy). *An agent's strategy is called truth-telling if and only if  $\sigma_i(Y_i = y|X_i = x) = 1, \forall x = y$  and  $\sigma_i(Y_i = y|X_i = x) = 0, \forall x \neq y$ .*

In *heuristic strategies*, the report of the agents are independent of their observations. One heuristic strategy of particular importance is always reporting 0, formally defined below.

**Definition 15** (Denial Strategy). *An agent's strategy is called the denial strategy if and only if  $\sigma_i(Y_i = 0|X_i = x) = 1$  and  $\sigma_i(Y_i = 1|X_i = x) = 0$ .*

The denial strategy is an interesting strategy in our setting because the payment that agents receive depends on how many of them report 0 as their feedback. The following observation is fairly easy to see.

**Observation 1.** *In the settings described above, the denial strategy is the (strictly) dominant strategy for all agents and gives the maximum payment  $\mathcal{R}$ .*

A strategy  $\sigma_i$  is called (strictly) *dominant* if it gives agent  $i$  her highest possible payment, given any strategies of the remaining agents. Observation 1 implies that in the presence of rational

<sup>1</sup>Please see [38] for a model in which only agents who submit 0 as their answers are eligible for such payments. In this chapter, we assume that every agent is eligible for the payment  $\mathcal{R} \cdot o_w$  regardless of her answer.

agents, the outcome determined by the decentralized platform is bound to be 1.00, since every such agent will report 0 irrespective of their true observation. Such an outcome determination is not useful for any practical purposes; on one hand, it is not informative and hence provides no utility in terms of the information acquired, and on the other hand, if such an outcome is used to issue the payments to the agents, it can incur a huge loss on the platform.

**Peer-consistency.** To counteract this phenomenon, the agents need to be properly incentivized by the platform to provide their feedback truthfully. We propose to do this, by issuing them a side-payment in addition to the payment that they receive based on the outcome resolution. Clearly, any constant amount of such side-payment does not achieve this objective; the side-payments have to be contingent on the truthfulness of the agents' reports. However, since there is no way to directly establish the truthfulness of the feedback, we will appeal to the power of *peer-consistency mechanisms* [30] to align the incentives of the agents with their feedback. The most important constituents of the peer-consistency framework are the agents' beliefs about the observations of their peers. We will let  $P_i(X_p = x')$ , for  $x' \in \{0, 1\}$ , denote agent  $i$ 's (prior) belief about a randomly selected peer  $p$ 's observation  $X_p$  on a question being  $x'$ . We will assume that all questions are a priori similar so the prior belief of the agent is same for all questions.<sup>2</sup> After the agent makes a private observation  $X_i$  for a question, she updates her belief (posterior) about her peer's observation on that question only, to  $P_i(X_p = x'|X_i = x)$ .

The first objective is to ensure that the decentralized platform can be used as an oracle, in the sense that the outcome determined by the platform is correct. The next question is, how large do the side-payments need to be? Is it possible to implement the side-payment scheme suggested by the peer-consistency mechanism without incurring loss to the platform? Our benchmark here is the amount of money that the platform would have to pay if there were no side-payments in place, and therefore the outcome would be determined by the denial strategies of the agents. In other words, we define the *relative saving* of a side-payment scheme to be

$$\text{relative saving: } \frac{n\mathcal{R} - \mathcal{P}}{n\mathcal{R}},$$

where  $\mathcal{P}$  is the total payment (side-payment + outcome dependent payment) under the scheme to the agents. The reason for considering relative saving in this work and not the actual saving in monetary units is that the absolute saving is domain and scale dependent and not very informative in a general sense. Before we proceed, let us see what the best relative saving that we could hope for is.

**Proposition 2.** *If agents were honest (i.e. they reported truthfully ignoring the outcome dependent payments), the platform could make an expected relative saving of up to  $P(1)$  in the payments, where  $P(1)$  is the actual probability of a randomly selected report on the platform being 1.*

---

<sup>2</sup>If not all questions are a priori similar but there are known batches of a priori similar questions, our results can be extended for each batch separately. For example, in the web-services case, this can be done by grouping web-services with similar SLAs.



Note that the best possible saving is not 100%, because it depends on the actual quality of the service. In the web service example, Proposition 2 states that when the response times of the services are generally good i.e.,  $P(1)$  is high, the platform could make significant savings (up to 100% as  $P(1) \rightarrow 1$ ) if the agents were honest. Also, note that we are comparing against the ideal outcome, when agents would not need to be incentivized to act truthfully; a mechanism that fares well against this outcome, will fare well against any other side-payment scheme, including one in which the outcome determination is done by a costly third party.

**The PTSC Mechanism.** Since we are interested in relaxing the informational assumptions as much as possible, we will use a detail-free mechanism (that doesn't know agents' beliefs) for determining the side-payments on the decentralized platform. Note that it is not necessary that a given user may be able to answer multiple questions (about different web services). This rules out several multi-task mechanisms like [20, 104]. Thus, we will use the PTSC mechanism [97], which we describe here for completeness. To decide the reward for an agent, the mechanism selects another agent  $p$  who also submitted feedback for the same question. Suppose that the agent submits  $Y_i = y$  and the peer submits  $Y_p = y'$ . The side-payment of  $\tau(y, y')$  agent  $i$  under the PTSC mechanism is:

$$\tau(y, y') = \begin{cases} \alpha \cdot \left( \frac{\mathbb{1}_{y=y'}}{R_i(y)} - 1 \right) & \text{if } R_i(y) \neq 0 \\ 0 & \text{if } R_i(y) = 0 \end{cases}$$

where  $\alpha$  is a strictly positive scaling constant. The mechanism uses  $R_i(y) = \text{num}_i(y) / \sum_{\bar{y} \in \{0,1\}} \text{num}_i(\bar{y})$ , where  $\text{num}_i(y)$  is a function that counts occurrences of  $y$  in the feedback of all agents (except  $i$ ) across all questions. The PTSC mechanism is a special case of the PTS mechanism [56] and is based on the idea of using  $R_i(y)$  from other apriori similar questions to estimate the prior belief of the users. It is possible to use other ways to estimate the prior in the PTS mechanism and relax the requirement of having other questions.

**Subjective Equilibrium.** When referring to the “correct outcome” for rational agents, one needs to define an appropriate *solution concept* in which the outcome will be obtained. The standard objective in the peer-consistency literature is to ensure that the correct outcome is achieved in the equilibrium, or, in other words, that truth-telling is an equilibrium. A strategy profile  $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$ , which represents a collection of strategies of agents  $\{1, 2, \dots, n\}$ , is a *strict equilibrium* if for any agent  $i \in \{1, 2, \dots, n\}$ , the agent's expected payment is strictly maximized when she adopts strategy  $\sigma_i$ , i.e.  $\sigma_i$  is her *best response* to the strategies of the other agents. A strategy profile  $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$ , is an  $\varepsilon$ -*approximate equilibrium* if for any agent  $i \in \{1, 2, \dots, n\}$ , the agent's expected payment when she adopts strategy  $\sigma_i$ , is smaller than the expected payment any other strategy  $\sigma'_i$  by at most  $\varepsilon$ . Since beliefs need not be common among agents, i.e. they are subjective, the equilibrium concept that we adopt is the *ex-post subjective equilibrium* [113]. In this equilibrium concept, an agent's best response is independent of the beliefs of others. In the chapter, we will simply use the terms “equilibrium” and “ $\varepsilon$ -approximate equilibrium” for brevity.

### 4.3 Truthful Equilibrium and Savings

We first derive the conditions under which the PTSC mechanism can be used to ensure that the truth-telling strategy profile is an equilibrium, in the presence of outcome-dependent lying incentives for the agents. This is certainly a critical requirement for a side-payment scheme which elicits reliable information. In the next section, we will provide an even stronger guarantee, ensuring that truth-telling is also a “good” equilibrium, under some reasonable assumptions. In our analysis, we will use the following quantity.

**Definition 16** (Self-Predictor Value).

$$\delta^* = \min_i \left( \frac{P_i(X_p = 1 | X_i = 1)}{P_i(X_p = 1)} - \frac{P_i(X_p = 0 | X_i = 1)}{P_i(X_p = 0)} \right)$$

We note that  $\delta^* > 0$ , whenever the observations of agents are *positively correlated*; this means that conditional on observing 1, the posterior belief of the agent about her peer also observing 1 strictly increases compared to her prior belief about the same. This positive correlation of signals is a standard assumption in the literature of peer-consistency for binary answer spaces, e.g. see [54, 113] and it is under this condition that PTSC guarantees that truth-telling is an equilibrium.<sup>3</sup> We will make the same assumption throughout this chapter, and we will quantify the required scaling constant of PTSC as well as the relative savings of the mechanism in terms of  $\delta^*$ . Intuitively,  $\delta^*$  is a measure of correlation strength, and captures the relative increase in the posterior compared to the prior belief, as described above. A very similar quantity was defined in [97] capturing similar concept, differing on the fact that it was a multiplicative parameter rather than an additive one. The parameter is also closely related to the *update strength* in [32].

We emphasize here that the mechanism does not need to know the exact value of  $\delta^*$ , but we assume that an estimate of this value ( $\delta = \delta^* + \beta$ , for some  $\beta \in \mathbb{R}$ ) is known.

**Theorem 9.** *Given  $\delta$  and a scaling constant  $\alpha > \frac{\mathcal{R}}{n \cdot \delta}$ , the truth-telling strategy profile is a strict equilibrium if  $\beta \leq 0$ , and is a  $(\frac{\beta \cdot \mathcal{R}}{n \cdot \delta})$ -approximate equilibrium if  $\beta > 0$ .*

Note that the theorem is stated in terms of  $\varepsilon$ -approximate equilibria. This is because if the value of  $\delta^*$  is *overestimated* (i.e.,  $\beta > 0$ ), then the agents might have incentive to actually deviate from their truth-telling strategy, but that incentive is bounded by a typically small quantity. In fact, when the overestimation imprecision tends to be negligible (i.e.,  $\beta \rightarrow 0$ ) or when the number of agents grows large (i.e.,  $n \rightarrow \infty$ ), then  $\varepsilon$  goes to 0 and we obtain exact equilibrium. On the other hand, if we only *underestimate*  $\delta^*$  (i.e.,  $\beta < 0$ ), then we obtain exact equilibrium, regardless of the imprecision parameter or the number of agents.

Any overestimation of  $\delta^*$  does not hurt the saving compared to the case of a precise estimation; in fact, it actually improves it. In contrast, underestimating  $\delta^*$  can diminish the saving, but the

---

<sup>3</sup>In the original settings for which it was proposed [97], outcome-dependent lying incentives were not present.

loss again vanishes as the number of agents grows large. The relative savings of the mechanism are captured in the following theorem.

**Theorem 10.** *The expected relative saving in payments made in the truth-telling equilibrium is at least  $P(1) - \frac{1}{n\delta}$ , where  $P(1)$  is the actual probability of a randomly selected report being 1 in the truth-telling equilibrium.*

Note that as long as the condition  $n > \frac{1}{P(1)\delta}$  is satisfied, the lower bound on saving is actually a positive number. Finally, notice that as  $n \rightarrow \infty$ , the relative saving reaches the maximum achievable value  $P(1)$  discussed in Proposition 2. In a more favorable setting, when the beliefs of the workers are not arbitrary but are aligned with the real observation probabilities and the mechanism has access to  $\delta^*$ , it can be shown that for any  $n \geq 2$ , the platform makes strictly positive relative savings given by  $P(1)\left(1 - \frac{1}{n}\right)$ .

We conclude the section with the following observation. While the employment of the PTSC mechanism with an appropriate scaling constant can guarantee that truth-telling is an equilibrium strategy, it is not hard to see that if no further assumption are made about agents' beliefs, the denial strategy is still an equilibrium strategy in addition to the truth-telling strategy. [97] have shown that when outcome dependent lying incentives are not present, while this uninformed equilibria does exist in PTSC, it is not profitable (pays zero expected reward). Unfortunately, in the presence of outcome dependent lying incentives, this undesired equilibrium becomes more profitable than the truth-telling equilibrium because every agent can now get the maximum value of the refund  $\mathcal{R}$  by playing the denial strategy. Any attempts of making the truth-telling equilibrium more profitable in this setting are impaired by the following result.

**Proposition 3.** *If the denial strategy equilibrium exists in any mechanism in the presence of outcome dependent lying incentives, it is not possible to make the truth-telling equilibrium more profitable without causing loss to the platform.*

Here loss means that the total payment will be higher than the maximum refund  $\mathcal{R}$ . While this negative result is reminiscent of the known negative result about uninformative equilibria in the peer prediction mechanism of [82] reported in [54, 55], in our result focal uninformative equilibria arise because of outside incentives and not due to a weakness of the incentive mechanism.

## 4.4 Honest Agents

In many real-life platforms with many participants, it is natural to assume that at least a few of them will behave honestly, regardless of the monetary incentives that the platform provides. This can be attributed to several reasons; for example, to rational choices that are not explicitly captured by the payments, e.g., an interest in the well-being of society or some intrinsic

utility from “doing the right thing”, or even to some form of bounded-rationality [100] or risk-aversion. We show that the undesirable equilibrium highlighted in the previous section can be eliminated in our setting if it is known that there exists an arbitrary small non-zero fraction  $f$  of honest agents on the platform. In fact, it is only necessary that the agents *believe* that there is such a fraction of honest agents, which is a reasonable assumption in most real-world platforms. As it will be evident later, neither the rational agents nor the platform know the identity of the honest agents. Only assuming the existence of honest agents (without known identities) is fundamentally different from using identified trusted authorities for obtaining observations (as proposed in [54]), since the latter violates the decentralization of the platform, while the former does not.

For the analysis, we will use an alternative definition of the self-predictor value that we defined in Section 4.3. This definition adapts the self-predictor value to the situation when agents believe that only a  $f$ -fraction of other agents are honest and the remaining  $(1 - f)$ -fraction always report 0 irrespective of their observations, i.e. they follow the denial strategy.

**Definition 17** (Self-Predictor Value with Colluding Agents). *Let  $Q_i(X_p = 0|X_i = 1) = (1 - f) + f \cdot P_i(X_p = 0|X_i = 1)$  and  $Q_i(X_p = 0) = (1 - f) + f \cdot P_i(X_p = 0)$ . The self-predictor value with colluding agents is defined as*

$$\delta_c^* = \min_i \left( \frac{P_i(X_p = 1|X_i = 1)}{P_i(X_p = 1)} - \frac{Q_i(X_p = 0|X_i = 1)}{Q_i(X_p = 0)} \right)$$

Note that when  $f = 1$ , we obtain exactly the same quantity as in Definition 16.

**Lemma 2.** *If  $\delta^* > 0$ , then  $\delta_c^* > 0$ , for any  $0 < f < 1$ .*

We will exploit this property of  $\delta_c^*$  to show that it is possible to eliminate the denial strategy equilibrium for any non-zero value of  $f$ . Similar to the previous section, we assume that the mechanism knows only an estimate  $\delta_c = \delta_c^* + \beta_c$ .

**Theorem 11.** *Given that for  $f > 0$ , (a) an  $f$ -fraction of agents are honest, (b) the remaining  $(1 - f)$ -fraction adopt the denial strategy and (c) it holds that  $\alpha > \frac{\mathcal{R}}{n \cdot \delta_c}$ , the truth-telling strategy is a strict best response if  $\beta_c \leq 0$  and is an  $(\frac{\beta_c \cdot \mathcal{R}}{n \cdot \delta_c})$ -approximate best response if  $\beta_c > 0$ .*

The theorem implies that the collusion of the  $(1 - f)$ -fraction who adopt the denial strategy becomes unstable and the rational choice for them will be to break the collusion and deviate to the truth-telling strategy. In other words, the denial equilibrium is eliminated and the truthful equilibrium prevails. Thus, we get the following proposition.

**Proposition 4.** *Under the conditions derived in Theorem 11, the denial strategy is no longer an equilibrium strategy.*

Given that  $\delta_c^* \leq \delta^*$  by definition (and strictly smaller when  $f > 0$ ), the scaling constant  $\alpha$  of PTSC in this case is actually larger than before. The reason is that we are now not only

requiring that truth-telling is an equilibrium, but also that the denial strategy equilibrium is eliminated. Note that  $\delta_c^*$  is strictly decreasing in  $f$  and achieves its maximum, which is  $\delta^*$ , at  $f = 1$ .

For the saving, we first remark that the benchmark against which we compare now naturally becomes the rational outcome in which the honest agents report the truth and the remaining agents play according to their denial strategies. Concretely, the saving of a side-payment scheme, under which a total payment of  $\mathcal{P}$  are made to the agents, now becomes:

$$\text{relative saving: } \frac{n\mathcal{R}' - \mathcal{P}}{n\mathcal{R}'},$$

where  $\mathcal{R}' = \mathcal{R} \cdot \left[ (1-f) + f \cdot (1 - P(1)) \right]$ . Note that  $\left[ (1-f) + f \cdot (1 - P(1)) \right]$  is the expected value of the outcome when  $(1-f)$ -fraction of the agents play the denial strategy (always report 0) and the honest  $f$ -fraction report 0 only when they actually observe 0.

**Theorem 12.** *If  $0 < f < 1$ , the expected relative saving made by the platform in the truth-telling equilibrium is at least*

$$\left[ (1-f)P(1) - \frac{1}{n\delta_c} \right] \cdot \frac{1}{(1-fP(1))}$$

We remark that the baseline for computing relative saving now naturally becomes the rational outcome in which the honest agents report the truth and the remaining agents play according to their denial strategies and the above theorem has been derived accounting for this fact. In theorem 12, the lower bound on  $n$  needed for the saving to be positive is given by  $n > \frac{1}{P(1) \cdot \delta_c \cdot (1-f)}$ . Note that this lower bound depends inversely on  $(1-f)$ . If  $n$  is fixed, then one gets an upper bound on  $f$  given by

$$f < 1 - \frac{1}{P(1) \cdot \delta_c \cdot n}$$

An upper bound on  $f$ , or the direct dependence of  $n$  on  $f$  may seem counter-intuitive at first; why would one want to put a cap on the number of agents that always behave honestly? This is explained by the fact that these are merely the conditions required for a relative saving to be strictly positive. When there is a big enough fraction of honest agents, the effect of the colluding agents on the outcome decreases and so does the relative saving that can be made by incentivizing these colluding agents to deviate to the truth-telling strategy. This means that if there are more honest agents than what the bound suggests (which tends to 1 for large  $n$ ), then the platform will not actually save any money by implementing a side-payment mechanism. It should be noted however that Theorem 11 holds no matter how large  $f$  is, meaning that if the platform desires, at the expense of a negative saving, it can still implement the side-payment scheme in order to enforce that all agents are actually truth-telling in the equilibrium. The reason for wanting to do that could be to obtain correct information from the rational agents too, who would otherwise play denial strategy and introduce noise. It is further shown in the proof that the relative saving in this case too approaches the optimal relative saving as  $n \rightarrow \infty$ .

### 4.5 Simulations

In this section, we evaluate the savings of PTSC experimentally on two real-world datasets, described below.

#### 4.5.1 Dataset

We conducted experiments on the dataset<sup>4</sup> of [123], which contains real-world Quality of Service evaluation results from 339 trusted agents on 5,825 web services. The agents observe the response time (in seconds) and throughput (in kbps) of the web-services and therefore, the observations can be used as two different datasets for our purposes. The dataset exhibits some missing observations but still has an overall density of 94.8% for response time and 92.74% for throughput. The observations are real values which we placed into two categories, corresponding to “good” and “bad” performance, in order to fit them to our binary observation setting. We treated a response time of at most 1 second as a “good” response time and the rest as “bad”. This resulted in 83.71% good response time observations, on average across all services. Similarly, we treated a throughput above 5 kbps as a good throughput and anything below that as a bad. This resulted in 78.18% good throughput observations, on average across all services. Thus, in the context of our model,  $P(1) \approx 0.8371$  for response time and  $P(1) \approx 0.7818$  for throughput.

#### 4.5.2 Simulation Parameters

We are interested in simulating settings in which the observations in the dataset would have been made by self-interested agents (rather than trusted ones) who have an incentive to play the denial strategy. Therefore, the dataset acts as the *true* private observations of the agents, which they may or may not reveal truthfully to the platform depending on their incentives. We fix a constant refund amount  $\mathcal{R}$  in our simulations; since we will only discuss the relative saving, the actual choice of  $\mathcal{R}$  is not important here. We vary the number of agents that are asked to report their observations for a service, by randomly selecting a subset of the agents from the dataset for every web-service.

We approximate the self-predictor value  $\delta^*$  using the following process. We randomly sample, for each web service, two true observations. We use this sample to get an empirical estimate of the joint distribution of the observations of the agents and the prior distribution, and these two empirical estimates are used in the expression for  $\delta^*$ . The result of this process can be thought of as a way to produce  $\delta = \delta^* + \beta$ , i.e., the value  $\delta$  that appears in the statements of our theorems. As we mentioned in Section 4.3, since the value of  $\delta^*$  is calculated as a minimum over all the agents, overestimating this value might cause some agents to have incentives to deviate, and in particular switch to their denial strategies. To examine the robustness of our scheme against this phenomenon, we quantify the savings of the mechanism when a fraction

---

<sup>4</sup>Dataset is available at <http://wsdream.github.io>.

of agents, even with PTSC implemented, play the denial strategy.

### 4.5.3 Experimental Results

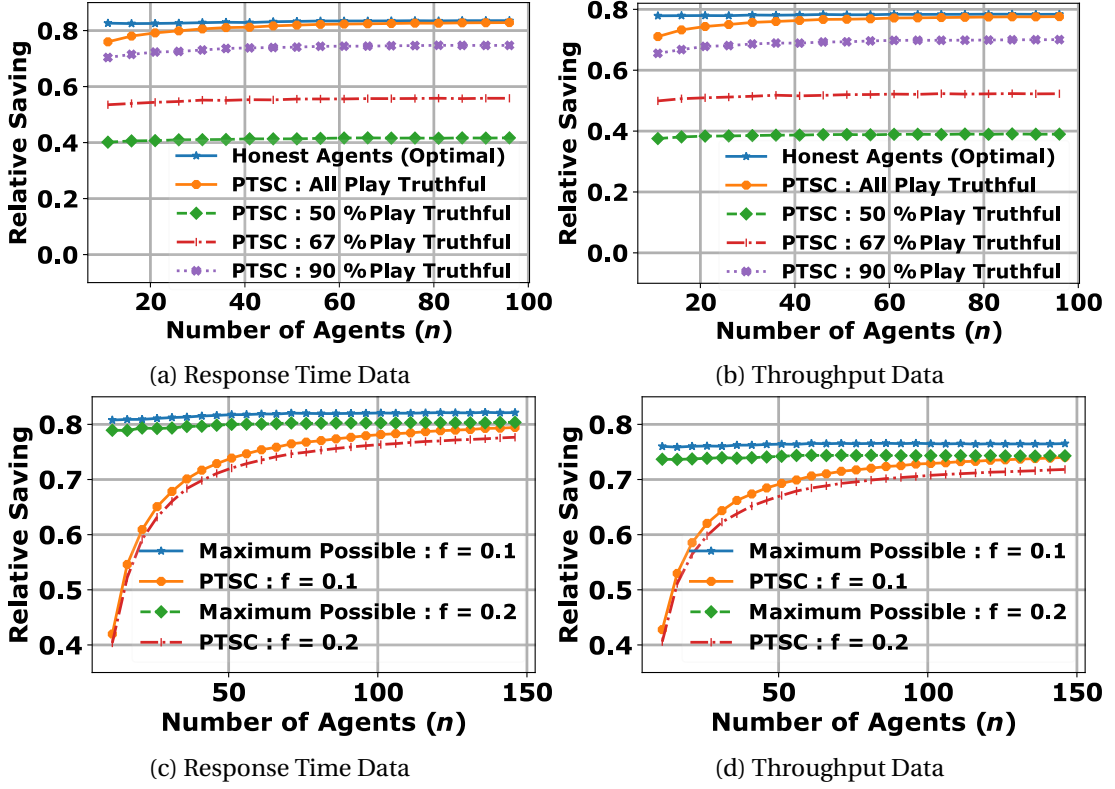


Figure 4.2: Relative saving made by PTSC.

In Figures 4.2a and 4.2b, we compare the saving achieved by PTSC against the optimal saving, which is obtained when all the agents are honest. Specifically, the optimal saving is given by  $(\mathcal{P}_d - \mathcal{P}_\alpha)/\mathcal{P}_d$ , whereas the saving of PTSC is given by  $(\mathcal{P}_d - \mathcal{P}_{eq})/\mathcal{P}_d$ , where  $\mathcal{P}_d$  is the refund payment of the denial strategy equilibrium,  $\mathcal{P}_\alpha$  is the refund payment when all agents are honest and  $\mathcal{P}_{eq}$  is the total payment of PTSC, including the refund and side-payments. In line with our theoretical observation in Theorem 10, the saving achieved by PTSC converges the optimal saving, which is approximately  $P(1)$ , as the number of agents increases. In fact, the saving approaches the optimal levels quite quickly, for reasonable numbers of agents (i.e., approximately 40 agents). To quantify the robustness of PTSC with respect to the estimation of  $\delta^*$ , the figure also depicts the relative saving made when only a 90%, 67% and 50% fraction of the agents receive the PTSC side-payments and report truthfully, and the rest receive the PTSC side-payment but still use the denial strategy. While the saving naturally declines, we observe that even with 90% of the agents being truthful, we achieve a significant saving.

We also consider the relative saving of PTSC when the side-payments are large enough to not only make truth-telling an equilibrium, but to also eliminate the denial strategy equilibrium, as

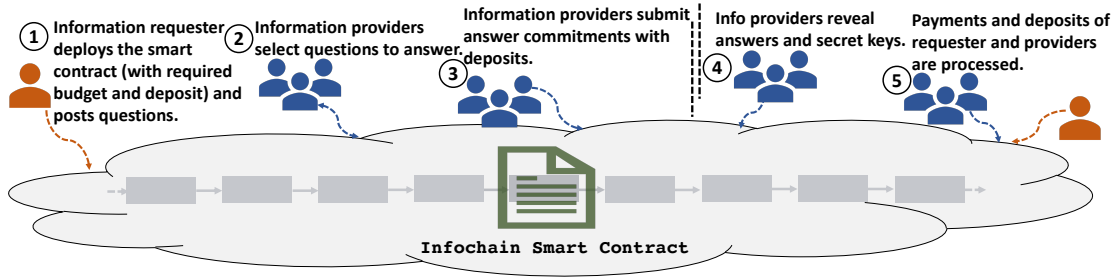


Figure 4.3: Infochain Overview

discussed in Section 4.4, assuming that there exists an  $f$ -fraction of honest agents who always report truthfully. We set the value of  $f$  to either 0.1 or 0.2, and observe how quickly the relative savings made by PTSC can reach the maximum achievable relative saving as the number of agents increase; this is shown in Figures 4.2c and 4.2d. Note that unlike Figures 4.2a and 4.2b, here the relative saving starts at a lower value; this is because the scaling constant and hence the payment made by PTSC are required to be larger as discussed after Theorem 11. Also, note that we have a different maximum possible saving bound for each  $f$ . This is in agreement with our discussion following Theorem 12 i.e., larger values of  $f$  lower the maximum achievable relative saving.

## 4.6 Infochain

To collect truthful information from self-interested agents, we propose a completely decentralized, transparent and trustless system called Infochain. Infochain enables information requesters to post questions, which can be selected by information providers (agents). The questions can be, for example, of the following form: “Is the responseTime of web service  $W$  less than 1 second?”. Once the agents submit information for the questions they select to answer, their payments in Ether are processed by a smart contract. All the collected information and payments are stored on a public blockchain to ensure transparency and immutability.

In addition to the Peer Truth Serum for Crowdsourcing (PTSC) [97], we also include the following two mechanisms in Infochain.

- 1. The Output Agreement (OA) Mechanism [110]:** This is perhaps the simplest of all peer-consistency mechanisms. In the OA mechanism, an agent gets a reward of 1 unit only if her answer for a question matches the answer of her peer for the same question. The reward of the agent for a question is the average over the rewards earned by matching with all peers. The final reward of the agent is the average of her rewards from all the questions answered by her.
- 2. The Dasgupta and Ghosh (DG) Mechanism [20]:** In the DG mechanism, an agent gets a reward of 1 unit if her answer for a question matches the answer of her peer for the



same question but also gets a penalty of 1 unit if her answers match the answers of the peer on non-common questions. The DG mechanism requires that two agents, who are peers of one another, must also have some non-common questions that are answered by one of them but not by both. The final reward is calculated by averaging as described in the OA mechanism. The **Correlated Agreement** mechanism [104] is a generalization of the DG mechanism and exhibits similar computations.

The final reward in the PTSC mechanism is also calculated by averaging as for the other two mechanisms. Traditionally, these mechanisms are implemented by a centralized trusted third party. Implementing them in Infochain, which doesn't assume any centralization or trust, is challenging. In the following subsections, we address the main implementation and theoretical challenges. An overview of Infochain is provided in Figure 4.3.

#### 4.6.1 Commit-Reveal Protocol

Transparency is an inherent feature of blockchain. Thus, all the information submitted by an agent is visible to all others. The peer-consistency mechanisms guarantee their incentive compatibility assuming that an agent can only form a belief about what her peers are going to report but doesn't know the actual report of peers. We ensure this in Infochain by making the agents follow a commit-reveal protocol:

1. **Commit:** An agent writes her commitment  $keccak256(y, k)$  on the chain, where  $y$  is the agent's answer for a given question and  $k$  is her secret key.
2. **Reveal:** Once all agents who have selected a question, have finished submitting their commitments for the question or the commitment phase expires, they can reveal their respective secret keys and answers. If the commitment of an agent matches her revealed answer, the answer is written on the chain, otherwise it is discarded.

#### 4.6.2 Cost Optimizations

Performing computations on Ethereum's Virtual Machine (EVM) remains an expensive affair. Computation costs on EVM are roughly  $10^8$  times higher than AWS<sup>5</sup>. [101] provides a good summary about the costs of basic arithmetic operations and writing operations for different data types. Agents who provide information must be compensated for this cost, increasing the overall cost of information acquisition. We discuss below several non-trivial ways of implementing three different peer-consistency mechanisms in Solidity so that the costs can be minimized.

1. **Optimizing Writing Cost:** To minimize the costs of writing on the chain, agents on Infochain combine multiple answers in the form of a bit vector. This is motivated by

<sup>5</sup><https://aws.amazon.com/blockchain/>

two observations. First, the answers are revealed simultaneously and thus, they do not require separate commitments. Second, the EVM operates on 256 bit words, thus a single bit vector is much cheaper to write than other formats.

**Proposition 5.** *With the above scheme, each 256-bit commitment can contain up to 42 answers.*

*Proof.* Given a hash function  $\mathcal{H}$  with a  $3k$  bit output, to commit the  $k$  bit message  $m$ , Alice generates a random  $k$  bit string  $\mathcal{S}$  and sends Bob  $\mathcal{H}(\mathcal{S}||m)$ . The probability that any  $\mathcal{S}'$ ,  $m'$  exist where  $m' \neq m$  such that  $\mathcal{H}(\mathcal{S}'||m') = \mathcal{H}(\mathcal{S}||m)$  is  $\approx 2^{-k}$ . The size of the message sent is limited to one third the size of the output of the hashing function, thus 85 bits. Each answer requires 2 bits: the first determines if the question was answered and the second is the answer. Therefore each commitment can contain 42 answers.  $\square$

This optimization helps both commit and reveal phases.

2. **Optimizing Computation Cost:** To reduce the cost of computing the rewards, a set of so-called intermediary values is introduced. These values naturally appear at intermediary states of reward computation. They will be precomputed and reused for each agent. What these intermediary values are, depends on the peer-consistency mechanism. This approach allows for the computation to traverse the data a minimum number of times. Since all rewards are computed at the same time, these intermediary values don't need to be written on the blockchain and can be kept in memory instead.

For an example, consider the PTSC mechanism, which requires relative frequency  $R_i(y)$  of the value  $y$  while excluding the answer given by agent  $i$ . This quantity need not be calculated from scratch for every agent or when every new answer is submitted and neither it is required to be written on the chain. The intermediary values (for e.g. running average) can be kept in memory and used to calculate or update  $R_i(y)$  as required.

### 4.6.3 Random Peer Selection

In peer-consistency, we can use only one or a few randomly selected peers for reward calculation instead of all peers. This is because, in expectation, the rewards of the agents remain unchanged and thus, the mechanisms with randomly selected peers also offer the same incentive compatibility (except that the variance in rewards increases). This is an interesting tradeoff between computation cost and variance in rewards. However, random peer selection on blockchain is subtle mainly due to the fact that nothing on the chain is a "secret", including the seed for random number generation. If random peers can be known in advance, it may increase the risk of collusion between the agents compromising the incentive compatibility of the mechanisms. In Infochain, we use the block timestamps as well as the mining difficulty level as the seed. This avoids using any trusted third party for random peer selection. The approach works under the assumption that the miners will not try to cheat the smart-contract,

which is a reasonable assumption given that the miners have no incentive to do so since they risk losing their mining rewards. The assumption can be violated in extreme scenarios where the financial activity on Infochain (for e.g. the incentive amounts) exceed the mining rewards.

#### 4.6.4 Negative Payments

The DG mechanism and the PTSC allow negative payments, which is implemented in Infochain by making agents submit refundable deposits. Information requesters also deposit the payment budget and an additional refundable deposit. Any outstanding deposits of the agents and the requester are returned after the payments and computations costs are settled.

## 4.7 Experiments

We now discuss the results of some experiments performed on Infochain. The performance measure of interest in this discussion will be the total amount of gas used. Gas is a unit measuring the computational work of running transactions or smart contracts in the Ethereum network and is a good proxy for the cost in USD. Infochain has been deployed and tested on the Ropsten Test Network, one of the commonly used public testing framework for Ethereum smart contracts. To have no limitations in terms of gas, the results reported in this chapter have been generated on a local instance of Ethereum.

### 4.7.1 Dataset

For this experiment, we used the response time attribute from the same dataset that was described earlier in Section 3.7.1. The dataset acts as the ground truth data that the information requester is interested in eliciting from self-interested agents. We simulated agent behavior as follows: 50% of the agents report truthfully, 25% report randomly (i.e. independent of the ground truth) and the rest report in an adversarial way (i.e. opposite of the truth).

### 4.7.2 Results

In Figure 4.4a, we show the reduction in writing cost due to the proposed optimization discussed in Section 4.6.2 as compared to the baseline implementation (without any proposed optimizations). Tasks in the figures refer to the questions that the agents answer. As expected, the reduction becomes more significant as agents answer more questions since the optimization can pack more and more answers into a single write operation. It may be worth noting that the optimization doesn't make writing cost independent of the number of answers as the figure may suggest. Since the number of questions in the figure doesn't exceed 42, the cost remains same as number of questions increase. Figure 4.4b shows the reduction in computation cost due to the proposed optimizations with varying number of agents and number of questions per agent. The figure was plotted based on the numbers obtained with the PTSC

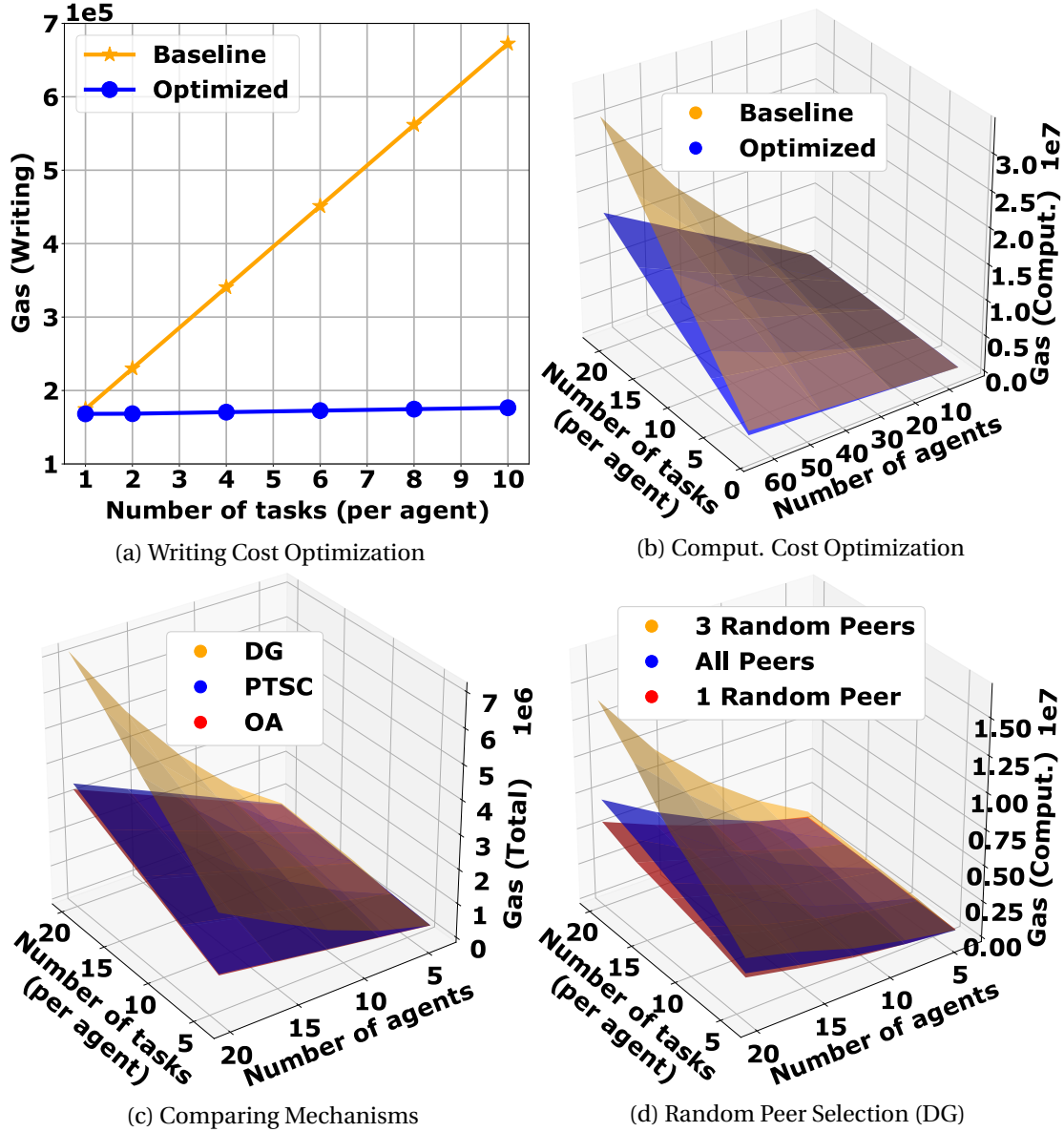


Figure 4.4: Experimental Results

mechanism but we observed a similar trend for the OA and the DG mechanisms. We next compare the cost of the three mechanisms in Figure 4.4c. While the OA mechanism and the PTSC mechanism have similar cost, the DG mechanism is more costly. This is due to the fact that DG mechanism involves more operations, particularly for keeping track of questions that are not shared between agents. Finally, Figure 4.4d shows the effect of using randomly selected peers for reward computation in the DG mechanism. We note that there may be multiple ways to implement sampling without replacement; for e.g., 1) randomly select a peer, check if it is already in the list of previously selected peers and repeat; and 2) sample from the list of not selected peers, update the list of not selected peers and repeat. The first method is not suitable

for blockchains as there is no upper bound on the number of necessary random selections and thus the transaction may run out of gas. The results presented here correspond to the second method. As shown in Figure 4.4d, the cost is guaranteed to reduce if we randomly select only one peer per agent. But when multiple peers are to be selected (which is required to reduce variance in rewards), the cost may increase to a level higher than the cost of using all peers without any random selection. The reason for this is that as we select more random peers, the cost of implementing random sampling exceeds the cost of simple implementation of just using all the peers.

## **4.8 Chapter Summary**

In this chapter, we studied settings motivated by polls and other crowdsourced data where the agents reporting the data have a conflict of interest with the aggregate statistic of the reported data. Such scenarios occur for example in reporting environmental data, where some reports might downplay pollution, in polls such as the LIBOR, where reporting agents have direct financial interest in the poll result, or our running example of self-reported Quality of Service measurements, where reporters may hope for refunds. We showed

- (i) how a detail-free peer-consistency mechanism, the PTSC mechanism, can be implemented to guarantee that truth-telling is an equilibrium of the induced game, in spite of the outside incentives to the contrary,
- (ii) how the presence of honest agents, which can remain anonymous, eliminates the undesired equilibrium where all agents report the outcome that benefits their outside incentive; and
- (iii) lower bounds on the relative saving in the net payments achieved by the mechanism, which approach optimality as the number of agents grows large.

We only considered a scenario where the outside incentives favor the same misreport for all agents, and do so with a particular dependence on the outcome. In ongoing work, we are considering different forms of outcome dependence, in particular threshold functions that require that the outcome exceeds a given threshold for the users to get refunds, and it turns out that these lead to different results. In the future, it would also be interesting to consider cases where agents have different and possibly opposing interests, such as in polls where different populations want different outcomes to win. Given that PTSC provides guarantees for non-binary signal spaces too, it would also be interesting to study similar problem beyond the binary answer setting. However, that seems to require somewhat different formalization for the correct determination of the outcome and the compensation schemes.

We presented a novel system called Infochain that implements decentralized, trustless and transparent oracles on the Ethereum blockchain. Contrary to earlier proposals on decentralized crowdsourcing systems, Infochain addresses the issue of truthfulness by implementing

## **Chapter 4. Infochain: A Decentralized, Trustless and Transparent Oracle on Blockchain**

---

game-theoretic peer-consistency mechanisms. For the first time, we discussed issues that arise in implementing these mechanisms in blockchain. The chapter also presents an important new criterion for comparing or evaluating these mechanisms by their implementation complexity on the Ethereum blockchain.

## 5 Tackling Peer-to-Peer Discrimination in the Sharing Economy

This chapter is based on the following publication:

N Goel, R Maxime, B. Faltings. **Tackling Peer-to-Peer Discrimination in the Sharing Economy.** In *Proceedings of the ACM Web Science Conference (WebSci)*, 2020.

### 5.1 Introduction and Related Work

Since the creation of eBay in 1995, the basic idea of peer-to-peer sharing of goods and services has led to many successful commercial platforms on the web. This way of distributing goods and services is often called as the sharing economy. The field saw a dazzling boom in the early 2010's when the popularity of Uber and Airbnb began to soar. Today several sharing economy platforms are operational on the web in diverse areas such as travel, real estate, transport, labor, finance and technology etc. The platforms offer an attractive alternative to both the 'producers' and the 'consumers' over the traditional ways of doing business due to being more easily accessible, sustainable and decentralized in nature. However, several recent studies have highlighted some very important ethical challenges faced by these platforms. For example, [28] found that Airbnb booking requests from the researchers (posing as guests) were 16 percent less likely to be accepted when the researchers made the requests from guests accounts with distinctively African American names relative to the case when they used identical guests accounts with distinctively white names. Similarly, [29] found that prices of properties on Airbnb offered by black hosts tend to be significantly lower than their white counterparts, even while keeping other relevant factors constant. We conjecture that less demand or trust for properties offered by black hosts is one of the reasons for the lower prices. Beyond Airbnb, [34] have observed racial discrimination by Uber drivers via more frequent cancellations against passengers when the researchers used African American sounding names for passenger accounts. Thus, the discrimination exists both ways. Hosts and drivers (providers) racially discriminate among guests and passengers (consumers) and vice-versa. It is deeply concerning that the existing social biases are finding their way into web

based platforms too. A combination of biased human feedback and large scale algorithmic decision-making on the web can cause further social segregation of historically disadvantaged groups. Thus, the problem needs an urgent attention and solution.

[2] (from Airbnb and Stanford) conducted an extensive user study on real Airbnb users and claimed that reputation systems offset the real world social biases by building trust between different users. The design of this user study was motivated from the concept of trust based investment games in economics. This is indeed a very positive finding. Trust between the users is the fundamental reason why Airbnb works in the first place [35]. However, it may be noted that even though the study was conducted with real Airbnb users, the reputation scores used in the study were generated synthetically. The reputation systems must themselves be non-discriminatory for them to actually work in the expected manner. A reputation system that discriminates against people based on race or gender will only further reinforce the bias. Unfortunately, the reputation systems on these platforms are often discriminatory towards different races and genders. This was analyzed in great detail by [41] for freelancer marketplaces like taskrabbit.com and fiverr.com. We found a similar trend on Airbnb too in our study. The findings are not very surprising because reputation systems are based on aggregating the rating provided by users (humans) to one another and human society has a long history of racial bias and discrimination. Unfortunately, it is a non-trivial problem to make reputation system non-discriminatory because the goal of the reputation systems is really to discriminate between users. However, this discrimination must be based on relevant attributes and not on sensitive attributes like race or gender. Thus, the objective is to make reputation systems racially non-discriminatory while retaining the other useful information they provide. This problem is similar to the problem of making machine learning systems non-discriminatory, where there is a trade-off between increasing the discriminative power (classification accuracy) of the classifiers and reducing racial discrimination in their decisions at the same time. The latter problem has received a lot of attention recently [18, 27, 31, 39, 42, 65, 117, 118, 119]. However, the solutions are specific to machine learning (mostly classification algorithms) and don't apply to reputation systems where humans are directly responsible for the reputations scores.

### 5.1.1 Our Contributions

In this chapter, we propose two solutions to make reputations systems on sharing economy platforms more fair and non-discriminatory. The first solution is to incentivize users (example guests on Airbnb) to find potentially high quality service providers (example hosts on Airbnb) from the disadvantaged group, evaluate them through deeper inspection (by using their service) and provide a truthful review of the service. We show that a game theoretic peer-consistency mechanism called the Peer Truth Serum for Crowdsourcing [97] can use the knowledge about the sensitive attribute of the service providers to ensure desired incentive compatibility in this scenario. This solution differs from the idea of offering explicit incentives just to explore the unexplored services (for example, see [44]). We provide the incentives



for exploration while truthfully rating the service, ensuring that the incentive mechanism doesn't cause a "reverse discrimination" and doesn't make it easier for the disadvantaged group to get business. If the design of the platform doesn't allow an incentive mechanism to be implemented, we propose a second solution. This solution applies to any reputation system irrespective of the reputation aggregation algorithm and the rating behavior of the users. We transform the aggregated reputation scores, such that the transformed scores are non-discriminatory to desired level while ensuring as little loss in their informativeness as possible. We model the problem as a constrained convex optimization problem and learn optimal transformation parameters that minimize information loss while respecting the constraints on the covariance between transformed scores and sensitive attribute(s).

## 5.2 Airbnb Case Study

### 5.2.1 Dataset

As discussed in the introduction, the phenomenon of peer-to-peer discrimination on the sharing economy platforms is already well-documented in the literature. Since we would need a real dataset to evaluate our proposed solutions and the datasets used in prior studies are not publicly available, we collected a new dataset for this work. We used the data available on Inside Airbnb <sup>1</sup>, which is "an independent, non-commercial set of tools and data that allows you to explore how Airbnb is really being used in cities around the world". We collected data for the listings in the New York City. The attributes that are of interest to us in this data include the aggregate rating (reputation scores) of the hosts, the prices of the listings, the profile pictures of the hosts and several other characteristics of the listings (for example number of bedrooms, bathrooms, guests, accommodates, min nights, reviews, reviews per month, location coordinates etc). We also add two labels to each listing: the ethnicity of the host and whether the listing is in a black majority neighborhood. To get the ethnicity of the host, we used the profile picture of the hosts and a face recognition library API called Kairos<sup>2</sup>. Kairos API, given the URL of an image, returns information about the people detected in the image, including a confidence score between 0 and 1 for five possible ethnicities: asian, black, white, hispanic or other. For each listing in our dataset, we take the ethnicity as the one having the maximal confidence score from Kairos results, and we remove the listing if this confidence score is not higher than a threshold (fixed at 0.7). This thresholding ensures that we filter out the listings for which Kairos couldn't detect the ethnicity of the hosts with enough confidence. To determine whether a given listing is in a black majority neighborhood or not, we used the coordinates of the listing and an additional piece of data (also available on Inside Airbnb) that contains the coordinates of the boundaries of the neighborhoods (for example, Harlem, Queens Village, Jamaica etc) in New York City. Using this information and a spatial analysis library in Python (Shapely), we were able to determine the exact neighborhood of each of the listings. We then used census data to determine whether a given neighborhood

---

<sup>1</sup><https://insideairbnb.com>

<sup>2</sup><https://kairos.com>

is black majority or not. This classification is also available online.<sup>3</sup> After all these pre-processing steps, we finally get a dataset of 8218 listings on Airbnb from New York City. Based on host ethnicity, 5716 listings are from white hosts and 2502 from non-white hosts (due to comparatively small proportions of other ethnicities in the dataset, we merged all non-white ethnicities). 3748 listings are in black-majority neighborhoods and 4470 are in other neighborhoods. Note that the imbalance in the dataset is a feature of the real-world. We had also collected similar datasets from Amsterdam, Geneva and San Francisco but the datasets were even more imbalanced (very few listings from minority groups) and hence, we skip discussion of those datasets in this chapter. It may also be noted that Kairos also allows us to find the age and gender of the hosts. We skip discussion about the distribution of these attributes as they don't lead to any interesting findings w.r.t. discrimination.

### 5.2.2 Data Analysis

As discussed in the introduction, there have also been user studies on Airbnb and Uber that go beyond just static data analysis and present more compelling evidence of discrimination. But in this chapter, we will restrict ourselves to static data analysis. In our data, there are two main attributes about a listing (and the corresponding host) that we focus on: the average rating (reputation scores) of the hosts and the prices of the listings. The average ratings are directly controlled by guests while prices are indirectly controlled by guests (due to lower demand and trust). Thus, if we find a significant difference in the values of these attributes for different ethnic groups, then it can be a potential case of bias. Table 5.1 shows the difference in average prices of the listings and the average rating of white and non-white hosts. Table 5.2 shows the difference in the listings of the hosts in black majority and other neighborhoods.

Table 5.1: Ratings and prices for different host ethnicity

	White Hosts	Others	Relative Difference
Avg. Price	\$139.12	\$105.51	31%
Avg. Rating	93.97	92.62	1.5%*

---

<sup>3</sup>[https://en.wikipedia.org/wiki/List\\_of\\_African-American\\_neighborhoods](https://en.wikipedia.org/wiki/List_of_African-American_neighborhoods)

Table 5.2: Ratings and prices for different neighborhoods

	Others	Black Majority	Relative Difference
Avg. Price	\$155.66	\$98.64	57%
Avg. Rating	94.02	93.20	0.9%*

\*These values are actually more significant than they seem. Ratings on Airbnb are almost always bigger than 85 (0.1 quantile is at 84.7), so the range of the ratings as shown above is not really 0-100 and the relative difference could be much higher after scaling (around 18% and 10% respectively if we consider the range to be between 85 and 100).

To further confirm the bias, we perform a regression analysis on the prices of the listings and ratings of the hosts (as target variables) with all the observable features that we could get from Inside Airbnb, including ethnicity of the host and neighborhood type. A similar approach was followed in [41] for confirming bias on taskrabbit and fiverr. The regression results obtained using Python's statsmodels library (linear model, OLS) are shown in Table 5.3 and 5.4. The 'coef' columns shows the linear relationship between observed variables and the price (or ratings) and the ' $P > |t|$ ' column shows the p-values for the relationship. This confirms that even after accounting for the observable features, ethnicity of the host (with positive correlation for white hosts) and the majority ethnicity of the neighborhoods (with negative correlation for black majority areas) have a statistically significant effect on the ratings and prices.

Another point to note in Table 5.3 is that, as expected, ratings also have a statistically significant positive effect on prices. While the platform itself has no direct control over the prices, it can definitely design better reputation systems which would then affect prices as well. In the rest of the chapter, we will focus only on the ratings.

**Remark.** We will be using the example of guests discriminating among hosts throughout the chapter, but the solutions proposed in the chapter are more general and apply for tackling discrimination in the reverse direction as well (most platforms like Airbnb and Uber have two-way reputation systems but it is more difficult to collect data about the reputation scores of guests and passengers).

Table 5.3: Regression Analysis for Price

	<b>coef</b>	<b>std err</b>	<b>t</b>	<b>P&gt; t </b>
const	-69.8484	19.962	-3.499	0.000
<b>accommodates</b>	26.7803	1.646	16.268	0.000
<b>bathrooms</b>	31.6218	4.180	7.565	0.000
<b>bedrooms</b>	16.0144	3.346	4.786	0.000
beds	-7.0650	2.845	-2.483	0.013
<b>guests</b>	8.5204	1.756	4.852	0.000
<b>min nights</b>	0.4624	0.102	4.547	0.000
reviews	-0.0011	0.054	-0.020	0.984
<b>reviews/month</b>	-7.0454	1.204	-5.852	0.000
<b>rating</b>	0.9372	0.204	4.588	0.000
<b>black majority area</b>	-55.7336	3.326	-16.755	0.000
<b>white host</b>	14.3567	3.608	3.979	0.000

Table 5.4: Regression Analysis for Ratings

	<b>coef</b>	<b>std err</b>	<b>t</b>	<b>P&gt; t </b>
const	93.4629	0.313	298.216	0.000
accommodates	-0.1241	0.090	-1.375	0.169
<b>bathrooms</b>	-0.6018	0.226	-2.659	0.008
bedrooms	0.1657	0.181	0.916	0.360
beds	-0.2936	0.154	-1.912	0.056
<b>price</b>	0.0027	0.001	4.588	0.000
<b>guests</b>	0.2543	0.095	2.680	0.007
<b>min nights</b>	-0.0152	0.005	-2.772	0.006
<b>reviews</b>	-0.0100	0.003	-3.432	0.001
<b>reviews/month</b>	0.3918	0.065	6.030	0.000
<b>black majority area</b>	-0.5371	0.182	-2.943	0.003
<b>white host</b>	1.0296	0.195	5.291	0.000

### 5.3 Bias Free Rating Elicitation

Online reputation systems involve two main steps. In the first step, the users provide ratings and in the second step, the platform aggregates the ratings into reputation scores. In this section, we intervene in the first step and discuss how an incentive mechanism can elicit fair ratings from users.

#### Incentive Mechanism Design Goals.

1. Users should be encouraged to try the services of high quality individuals belonging to the disadvantaged class.
2. Users should provide truthful ratings about the received quality of service. Providing a truthful rating about the service is important because we don't want to offer uncondi-

tional benefits for any individual belonging to any class.

While there may be several ways to achieve the first goal alone, achieving the second goal together with the first goal is a hard problem because it also requires that user must reveal their private information (the quality of service that they received from the individual) truthfully, even though there is no way to verify what the user is saying is indeed true. The latter problem is known as the problem of “information elicitation without verification” and there is a rich literature on mechanisms for this problem. The mechanisms are commonly referred to as the “peer-consistency” mechanisms [30]. The examples of these mechanisms include the original peer-prediction method of [82] and the Bayesian Truth Serum of [93]). The broad idea in these mechanisms is to “match” the information provided by different users and reward the users based on agreement between the two pieces of information. We next show that a state-of-the-art peer-consistency mechanism called the Peer Truth Serum for Crowdsourcing [97] achieves both design goals listed above, if it has access to the sensitive attribute of the individuals on the platform.

#### 5.3.1 Preliminaries and Notation

Let the individuals providing services on a platform be categorized into two classes, based on the value  $z$  of their sensitive attribute  $Z$ . For example, on Airbnb, hosts can be categorized into black hosts and white hosts. In the case of freelancer platforms like taskrabbit, fiverr [41], workers can be categorized into male and female workers. The sensitive attribute is not necessarily required to be binary one and there can also be multiple sensitive attributes. The discussion in the chapter can also be extended to these general cases. The users of the platform, who take the services of these individuals, have private prior beliefs about the underlying quality of the services provided by individuals belonging to different classes. The prior belief of a user may be different for different classes. For example, a user may believe that male workers are able to provide better service or that white hosts provide better accommodation. We formalize beliefs of users as probability distributions about the quality of service. Let the quality of service be expressed using a discrete signal  $Q$  that can take values in  $\{1, 2, \dots, k\}$ , 1 being worst and  $k$  being best. Then, prior belief of a user about the quality of service provided by an individual belonging to class  $Z = z$  is given by  $P_z(Q = q)$ . Once the user personally observes the service quality  $q'$  offered by an individual  $\Psi$  (for example after hiring  $\Psi$  or staying in the accommodation offered by  $\Psi$ ), he updates his belief about this particular individual  $\Psi$ . This is expressed by his posterior belief  $P_\Psi(Q = q|q')$ : given that he received a service of quality  $q'$  himself, the probability that any other user on the platform will receive a service of quality  $q$  from the same individual  $\Psi$ .

#### 5.3.2 The Peer Truth Serum for Crowdsourcing [97]

Let  $r$  denote the rating given by a user  $i$  to an individual  $\Psi$  belonging to class  $Z = z$  and let  $r'$  denote the rating given by another (randomly selected) user  $j$  to the same individual. Then,

the Peer Truth Serum rewards user  $i$  with  $\tau_i$  given by:

$$\tau_i = \beta(z) \left[ \frac{\mathbb{1}_{r=r'}}{R_{iz}(r)} - 1 \right]$$

Here  $\mathbb{1}_{r=r'}$  is an indicator function which returns 1 if the ratings  $r$  and  $r'$  match and 0 otherwise.  $R_{iz}(r)$  is the relative frequency of  $r$  in the ratings received by all individuals of class  $Z = z$ , excluding the ratings given by user  $i$ . More formally,  $R_{iz}(r) = \frac{\text{num}_{iz}(r)}{\sum_{\bar{r} \in \{1,2,\dots,k\}} \text{num}_{iz}(\bar{r})}$ , where  $\text{num}_{iz}(r)$  is a function that counts occurrences of  $r$  in the ratings of all individuals (except  $i$ ) of class  $Z = z$ .  $\beta(z)$  is a strictly positive scaling constant for the class  $Z = z$  such that the constant for the disadvantaged class is sufficiently bigger than the constant for the other class. Rewards can also be calculated by taking average by matching the ratings with multiple other users instead of single user (to reduce variance).

### 5.3.3 Belief Update Assumption

We will make a weak and standard assumption (the self-predicting assumption [97]) about the way users update their belief after they observe a quality of service. We assume:

$$\frac{P_\Psi(Q = q|q)}{P_z(Q = q)} > \frac{P_\Psi(Q = q'|q)}{P_z(Q = q')} \quad \forall q, q' \in \{1, 2, \dots, K\}$$

The assumption says that the relative change in posterior (over the prior) about what quality other users will observe from individual  $\Psi$  is the highest for the quality that the user himself observed. The assumption is easiest to understand in binary (good or bad quality) settings. If the user herself observed a good service, his belief about others receiving a good service from this individual doesn't decrease (or remain exactly same) as compared to his prior.

### 5.3.4 Game-Theoretic Properties

1. **Truthful Equilibrium**[97]. Under the self-predicting assumption on workers' beliefs (which can be heterogeneous and unknown), the mechanism induces a truthful Bayes-Nash equilibrium: if other users submit truthful ratings, it is the best strategy for any user to submit truthful rating. Further, the expected reward in the truthful equilibrium is strictly positive.

**Proof Sketch:** A rational user seeks to maximize the expected reward. The numerator in the first term of expected reward is the posterior belief of the user about another user receiving a certain quality of service given his own observation and the denominator converges to his prior belief about that quality of service offered by a random individual from that class. Then, truthful equilibrium follows from self-predicting assumption. A formal proof can be found in [97]. It holds even if the rewards are scaled by constants.

2. **Robustness to Collusion** [97]. The mechanism ensures that truthful equilibrium is not

just an equilibrium but the most profitable equilibrium. So collusion strategies are not profitable. For example, a simple strategy in which users may always submit the same rating (irrespective of true quality) so that their rating always match gives zero expected reward.

**Proof Sketch:** If everyone gives same rating irrespective of what quality they actually observed, then while numerator is always 1, the denominator (relative frequency of that rating) is also always 1 (net reward is 0). More generally, for ratings provided randomly (independent of the true quality), numerator and denominator converge to same quantity [97].

3. **Higher Reward for Truthful Ratings in the *Disadvantaged Class* but No Free Ride.** The mechanism gives higher reward if the users (try the service and) truthfully rate good quality individuals from disadvantaged class but doesn't incentivize giving good rating for a bad service.

**Proof Sketch.** The scaling factor  $\beta$  is higher for the disadvantaged class but due to the truthfulness property of the mechanism, the higher scaling factor only helps when the provided rating is also truthful.

**Remarks.** Assuming that the rewards given by the mechanism are small compared to the cost paid by the users and the disutility of actually receiving a bad service, the mechanism only incentivizes the users to actively search for individuals within the disadvantaged class that are very likely to offer high quality service (for example, based on photographs /presentation of the service). It is true that in this way, the mechanism also benefits individuals from the disadvantaged class who already have high ratings, but if this is not desired, one can re-define the class of individuals taking into account other attributes also (for example prior number of reviews) and apply the mechanism at a finer level of class definition. For example, on Airbnb example, 'super hosts' status already makes this distinction. Finally, we note that the scaling constant  $\beta$  can be dynamically changed and made equal for the two groups once it has served its purpose (i.e. when there is no disparity on the platform). The mechanism then continues to incentivize truthful reporting, leading to a sustainable long-term fairness on the platform.

#### 5.3.5 Sensitive Attribute Information

Our mechanism uses the information about the sensitive attribute (class) of the individuals to calculate  $R_{iz}(r)$  using the ratings of all individuals across a class  $Z = z$  and also to scale the rewards with class dependent scaling constants  $\beta(z)$ . This information is required to ensure that the mechanism works in the desired way.

To give a concrete example, imagine that the  $R_{iz}(r)$  for some  $r$  denoting a good rating is 0.8 for the disadvantaged class and is 0.9 for the other class. Remember that  $R_{iz}(r)$  is estimate of

the prior belief. Let's assume that after a user receives a good service from an individual, her posterior belief that another user will also receive a good service from the same individual, increases by 0.02. Thus, it becomes 0.82 if the individual is from the disadvantaged class and 0.92 if the individual is from the other class. In this example, the user gets an expected reward of  $\beta(z) \left[ \frac{0.82}{0.8} - 1 \right] = 0.025\beta(z)$  or  $\beta(z) \left[ \frac{0.92}{0.9} - 1 \right] = 0.022\beta(z)$  depending on the class of the individual being rated. Let's further assume for simplicity that the same happens in case of a bad service also i.e. her posterior belief that another user will also receive a bad service from the same individual given that she herself received bad service, increases by 0.02. Thus, it becomes 0.22 if the individual is from the disadvantaged class and 0.12 if the individual is from the other class. In this example, the user gets an expected reward of  $0.1\beta(z)$  or  $0.2\beta(z)$  depending on the class.

Now imagine that we instead use an  $R_i(r)$  calculated from the ratings received for all individuals irrespective of their class, and a class independent scaling constant  $\beta$ .  $R_i(r)$  may be the average of 0.9 and 0.8, for example. Using this  $R_i(r)$  violates all three properties enumerated in section 5.3.4. The first two properties are violated because a common  $R_i(r)$  is no longer an estimate of the different prior beliefs for the two classes and the self-predicting assumption can not be used to guarantee truthfulness. The third property is violated because the scaling constant is no longer different for the difference classes. In fact, not using the sensitive attribute information in the mechanism may cause even more discrimination. The rewards no longer encourage exploring good service providing individuals from the disadvantaged class but on the contrary, may discourage doing so. In the above example, it is easy to see that truthfully rating a good service gets a negative reward of  $\beta \left[ \frac{0.82}{0.85} - 1 \right]$  for the disadvantaged class and a positive reward of  $\beta \left[ \frac{0.92}{0.85} - 1 \right]$  for the other class. On the other hand, truthfully rating a bad service gets a positive reward of  $\beta \left[ \frac{0.22}{0.15} - 1 \right]$  for the disadvantaged class and a negative reward of  $\beta \left[ \frac{0.12}{0.15} - 1 \right]$  for the other class. Indeed, it is not necessary that a similar difference in rewards for the two classes will always be observed (depending on different priors and belief update parameters) but the example clearly shows that it is possible that the bias may be reinforced if the sensitive attribute information is not used in the mechanism. Hence, it is important to use this information to achieve the desired outcome using the incentive mechanism. This is similar to using sensitive attribute in fair machine learning (at training and/or prediction time).

### 5.4 Bias Correction

While the incentive mechanism proposed in the previous section is an attractive method to make reputation systems fair, it is possible that the design and business model of some sharing economy platforms may not permit the implementation of such a mechanism since many platforms may be reluctant to pay for reviews. In such cases, an ideal solution would be to



somehow estimate the bias of the users and discount their opinions according to their bias parameters while aggregating the ratings into the reputation scores. However, such a solution can only work if:

- There is enough data available about every user. This means that every user must provide enough ratings across classes.
- The data from users follows a consistent probabilistic distribution so that their biased behavior can be learned. In our case, this requires that users provide ratings in a consistent way.

Unfortunately, most users on the web provide very few ratings making it hard to make any inference about their bias parameters. Further, it is difficult to model human rating behavior using a simple probabilistic model. Hence, we take an alternative approach and propose a “post-aggregation” correction technique.

#### 5.4.1 Post-Aggregation Transformation

We apply a transformation on the aggregated reputation scores of all individuals such that the transformed scores are non-discriminatory, while ensuring that the information loss due to transformation is minimum. More formally, let  $x = \{x_1, x_2, \dots, x_n\}$  be the originally aggregated reputation scores of individuals  $\{1, 2, \dots, n\}$  and let  $\tilde{x} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n\}$  be their transformed scores. Let  $l(\tilde{x}, x)$  be a loss function measuring the amount of information lost due to the transformation and let  $d(\tilde{x}, z)$  be a function measuring the discrimination in the transformed scores,  $z = \{z_1, z_2, \dots, z_n\}$  being the values of the sensitive attribute. Then, we have the following problem:

$$\begin{aligned} & \text{minimize} && l(\tilde{x}, x) \\ & \text{subject to} && d(\tilde{x}, z) \leq \delta \end{aligned} \tag{5.1}$$

Here,  $\delta \geq 0$  is the allowed threshold of discrimination in the transformed scores.  $l$  and  $d$  can be chosen according to the domain of application and the computational considerations. In our work, we make the following assumptions: (1)  $l$  is the mean squared error(MSE), which is a common choice for measuring difference in real valued ratings (for example, to evaluate recommendation systems algorithms), (2)  $d$  is the absolute value of covariance between the transformed scores and the sensitive attribute values, and (3)  $\tilde{x}_i = a \cdot x_i + b$ . With these assumptions, we get a convex optimization problem in  $a$  and  $b$ , which can be solved efficiently using existing tools. The sensitive attribute doesn’t have to be binary and it is easy to accommodate non-binary categorical sensitive attributes using one-hot representation of the sensitive attribute (a common trick used in data analysis and machine learning).

**Remarks.** The choice of covariance as a proxy to measure bias was explored by [117] in context of machine learning classifiers. In our settings, correlation would be a more appropriate metric than covariance due to scale invariance, but it makes the problem non-convex. Nevertheless,

as we will show, even covariance turns out to give good performance in our experiments. Finally, we note that instead of using identical transformation parameters  $a, b$  for all individuals, one could define  $\tilde{x}$  to have individual specific parameters  $a_i, b_i$  (at the cost of increasing the number of optimization parameters). It is also possible to consider more complex transformation functions (for example,  $\tilde{x}_i = a \cdot x_i^2 + b \cdot x_i + c$ ). However, it may not always be useful if  $d(\tilde{x}, z)$  is still measured using covariance (covariance captures only the linear dependence between two variables). For example, even with  $\tilde{x}_i = a \cdot x_i + b$  and covariance as the measure of bias, it is easy to show mathematically that the closed form solution for  $a$  is 1 in our optimization problem. The same is true for coefficients of higher order terms. This was also observed in our experiments. Thus, we only discuss simple transformations of the form  $a \cdot x_i + b$  in this chapter. It remains an interesting future work to explore whether other transformation functions  $\tilde{x}$  (together with more advanced measures  $d(\tilde{x}, z)$  of discrimination) can achieve better performance, while keeping the problem convex.

### 5.4.2 Range Scaling

On many platforms (including Airbnb), aggregated reputation scores always lie within a fixed range  $[L, U]$  (for example,  $[0, 5]$  or  $[0, 100]$ ). It is fairly easy to address this in our proposed solution. One natural fix is to include additional constraints on the parameters  $a, b$  in the optimization problem such that  $L < \tilde{x}_i = a \cdot x_i + b < U$  for any  $L < x_i < U$  and constants  $L, U$ . An even simpler approach is to apply a range scaling on the transformed scores (after optimization) so that the transformed scores lie in desired range  $[l, u]$ .

$$\tilde{x}_i^{scaled} = (u - l) \frac{\tilde{x}_i - \min(\tilde{x})}{\max(\tilde{x}) - \min(\tilde{x})} + l$$

In our experiments, we set  $l = \min(x), u = U$ . This ensure that the the minimum value of the transformed scores doesn't go below the minimum value of the original reputation scores. It may be noted that the scaling is a constant linear function applied to all scores; hence, the covariance between the scaled transformed scores and the sensitive attribute will stay the same as it was before scaling, and it doesn't alter the discrimination removal achieved by the constrained optimization step.

**Remark.** Assuming the conclusions of the user study conducted by [2], a one time correction in reputation scores should bring fairness on the platform by building trust between the users. In a less optimistic (and perhaps more realistic) scenario, the correction can be applied only at infrequent intervals, eventually leading to an ideal setting where no more corrections are required and reputation scores are fair by default.

### 5.4.3 Experiments

We implemented the above approach in Python (using Scipy's Sequential Least Squares Programming) and tested it on our Airbnb dataset. The results presented here are for the case

when host ethnicity was assumed to be the sensitive attribute but similar trends were observed when neighborhood ethnicity was assumed to be the sensitive attribute. The value of  $\delta$  was set to  $10^{-5}$ .

Table 5.5: MSE and Covariance after the transformation

	$MSE(x, \tilde{x})$	$cov(\tilde{x}, z)$
Before Transformation	0	0.228
After Transformation	0.246	$10^{-5}$

Table 5.5 shows that the covariance between the reputation scores and the ethnicity was reduced to  $1e-05$  as specified by the constraint ( $\delta$ ) and MSE increased to 0.246. While it is certainly good that covariance is now close to 0, it is not clear whether the increase in MSE is acceptable or not. Even a naive transformation technique (for example which assigns random reputation scores to individuals independent of their true reputation) could also achieve a zero covariance but that would clearly not be an acceptable solution. Thus, we perform a regression analysis on the transformed reputation scores (exactly as we did in Section 5.2). Table 5.6 shows that the p-value corresponding to the host ethnicity is now  $0.803 \gg 0.05$ , which means ethnicity is now an insignificant feature, while p-values for other relevant remain unchanged. This shows that our technique retained the desired information while removing discrimination. We note that if there are multiple sensitive attributes, our optimization framework can easily accommodate multiple constraints (one for each sensitive attribute). For example, if both host ethnicity as well as neighborhood's majority ethnicity are specified as sensitive attributes in the constraints, then the scores transformed using the two constraints would show no relation with both attributes.

We now show some additional experimental results on synthetic datasets.

- **Power Law Distribution.** We assumed that the sensitive attribute can now take 3 different values (for example black, white and asian hosts) and generated 5000, 10,000 and 15,000 reputation scores (one for each synthetic host) in the range 0-10 for these three classes. We generated the scores using power-law distribution (with parameters 3, 6 and 10 respectively) to closely model the distribution observed in Airbnb data. Table 5.7 shows the results of the transformation for  $\delta = 0.01$ .
- **Normal Distribution.** Similar observations were made when the data was generated using truncated normal distributions (10,000 scores from  $\mathcal{N}(5, 1)$  and 10,000 from  $\mathcal{N}(8, 2)$ ). Instead of presenting duplicate trends, we instead show some other interesting observations on the data generated using normal distribution. Figure 5.1 shows that the our threshold constant  $\delta$  in the optimization problem provides the platform direct control over the extent to which reputation scores can be altered, leading to different values of covariance and MSE. The flat parts of the curves show that if  $\delta$  is set to a value equal to or higher than the covariance of the original scores, then as expected our

Table 5.6: Regression Analysis for Transformed Ratings

	coef	std err	t	P> t
const	94.2129	0.313	300.610	0.000
accommodates	-0.1241	0.090	-1.375	0.169
<b>bathrooms</b>	-0.6018	0.226	-2.659	0.008
bedrooms	0.1657	0.181	0.916	0.360
beds	-0.2936	0.154	-1.912	0.056
<b>price</b>	0.0027	0.001	4.588	0.000
<b>guests</b>	0.2543	0.095	2.680	0.007
<b>min nights</b>	-0.0152	0.005	-2.772	0.006
<b>reviews</b>	-0.0100	0.003	-3.432	0.001
<b>reviews/month</b>	0.3918	0.065	6.030	0.000
<b>black majority area</b>	-0.5371	0.182	-2.943	0.003
white host	-0.0486	0.195	-0.250	0.803

Table 5.7: Synthetic Data Results (Power Law Distribution)

	$MSE(x, \tilde{x})$	$cov(\tilde{x}, z)$
Before Transformation	0	0.264
After Transformation	0.117	0.01

algorithm does not transform the scores and return the original scores. Figure 5.2 shows that the proposed technique scales linearly with the number of reputation scores to be transformed. We transformed upto 1M reputation scores (equal number of samples from two truncated normals) in under 4 minutes on a normal PC.

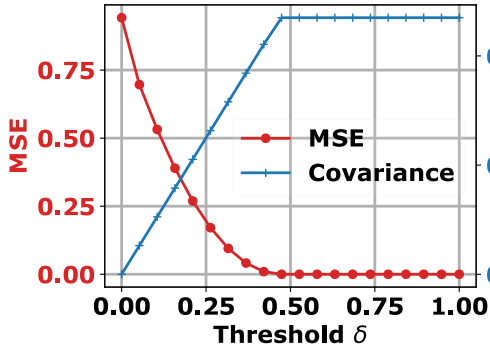


Figure 5.1: Controlling  $\delta$

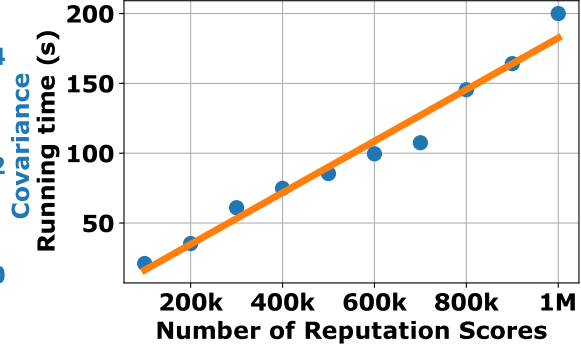


Figure 5.2: Running Time

## 5.5 Chapter Summary

In this chapter, we considered the problem of making reputation systems on the sharing economy platforms more fair and proposed two solutions: an incentive mechanism and a bias correction technique. The incentive mechanism encourages users to try the service of

individuals belonging to the disadvantaged class and at the same time also elicits truthful ratings about the quality of service received. This ensures that disadvantaged class doesn't receive unconditional benefit and the aggregated reputation scores compensate for any disparate benefit in the long term. Eventually, when individuals have reputation scores that truly reflect their quality irrespective of their class, the incentive mechanism can be easily modified through the scaling constant to stop offering different incentives for different classes, and it can continue to offer incentives for truthful reporting. The bias correction is also meant to be a similar short-term intervention, albeit a more directly controlled one. Even if the platform uses the bias correction solution instead of the payment mechanism, the mechanism can still be used to provide feedback to the raters through artificial currency or points.

There also remain several open questions. Usually, the platforms not only display the aggregated reputation scores but also each of ratings given by the users. On one hand, this further helps in emphasizing that an incentive mechanism is an ideal way to make reputation systems fair but on the other hand, it also means that a post-aggregation bias correction technique loses its utility since the bias is corrected only for the aggregated reputation scores. There is no trivial way to hide biased ratings without access to sufficient data to estimate biased behavior of some of the users. Another open problem is to elicit truthful textual reviews or to filter biased ones.



## 6 Crowdsourcing with Fairness, Diversity and Budget Constraints

This chapter is based on the following publication:

N Goel, B Faltings. Crowdsourcing with Fairness, Diversity and Budget Constraints. In Proceedings of the AAAI/ACM Conference on AI, Ethics and Society (AAAI/ACM AIES), 2019.

### 6.1 Introduction

Algorithmic decision-making is gaining popularity in many diverse application areas of social importance. Examples include criminal recidivism prediction, stop-and-frisk programs, university admissions, bank loan decisions, screening job candidates, fake news control, information filtering (personalization) and search engine rankings etc. Recently, questions were raised about the fairness of these algorithms. An investigation [94] found COMPAS (a popular software used by courts to predict criminal recidivism risk) racially discriminatory. Other software systems have also been found to be biased against people of different races, genders and political views [9, 63, 71, 90]. This has led to a widespread and legitimate concern about the potential negative influence of such systems on the society [8, 92]. One of the several reasons of algorithmic bias is the bias in the training datasets. In order to achieve algorithmic fairness, the issue of data fairness also needs to be addressed. In many interesting cases, data is directly or indirectly influenced by some kind of human feedback. The influence is obvious and direct if human assigned labels are used as a proxy for ground truth labels. However, human feedback can also indirectly influence the so-called “ground-truth” datasets (when the labels are not human assigned but observed in reality). This is because the ground truth labels can only be collected for a finite number of data points and the selection of data points is often influenced by humans. For example, there are no ground truth labels available for recidivism of people who were never released by the judges. There can be many other ways in which data can be biased. In this chapter, we focus only on the direct influence of human feedback on data fairness i.e. the case in which humans assign labels for data. In this thesis, we view **crowdsourcing data fairness** as an independent objective, separate from algorithmic

fairness. Algorithmic fairness is a more complex problem and needs further considerations beyond data fairness.

Crowdsourcing is increasingly used to collect training data labels. Inevitably, crowdworkers have different biases, which are then reflected in the labels collected from the workers. A very recent study [26] conducted on Amazon Mechanical Turk showed that the crowdworkers were equally racially biased as COMPAS in predicting recidivism. The difference in false positive rates of crowd predictions for white and black defendants was significant and nearly equal to that of the predictions made by COMPAS. The same was true for false negative rates also. The bias didn't change much even when the crowdworkers were not explicitly displayed the race of the defendants.

We consider settings similar to [26]. Workers are asked to provide their answers (or labels) about some tasks with unknown ground truth labels. Every task has some non-sensitive details that are shown to the workers and a sensitive attribute (for example, race) that is not explicitly shown. But the sensitive attribute may potentially be correlated with the non-sensitive task details. A worker inspects the tasks assigned to her and submits labels for the tasks. Each task is assumed to have a ground truth label but the workers don't have any way of accessing the ground truth. They can only use the task details, their prior knowledge and incomplete information from other sources to make an "educated guess" about the ground truth. The examples of such tasks are "Will a defendant with given personal history recidivate within the next two years or not?" or "Will a candidate with given CV be successful in the job applied for?" or "Is given political news item fake?". The sensitive attributes in these example tasks are race, gender and political group respectively. Every worker charges a fee for answering the assigned tasks. The requester has a budget constraint on the fees that she can pay to the workers.

### 6.1.1 Related Work

**Empirical Studies :** [26] finds racial discrimination in recidivism prediction tasks on Amazon Mechanical Turk (AMT). [88] analyzes linguistic bias in labels collected through GWAP (Games with a Purpose) on AMT. [89] analyzes the linguistic bias in collaboratively produced biographies. [41] finds discrimination in reputation crowdsourcing systems in online marketplaces.

**Proposed Solutions :** In an independent and pioneering work, [107] considers the problem of fairness in human decision-making tasks like recidivism prediction, without budget and diversity constraints. Criminal cases with *known* race information arrive in batches of *known* sizes and an MDP based maximum weighted matching algorithm assigns each case to *exactly one* human judge such that the overall utility from decisions of releasing or keeping any defendant is maximized, while ensuring *demographic parity* of release decisions across two races. To the best of our knowledge this is the first and the most recent work to consider settings somewhat similar to ours but our work differs from theirs in several ways. We consider general crowdsourcing settings, in which several assumptions from their model don't hold. In particular, they assume that "true" risk scores of individual defendants are known to the



human judges and the case assignment algorithm. In general crowdsourcing settings, one can only hope to have an overall label distribution for the population. In fact, finding the label probability for individual tasks is the very objective of crowdsourcing. Further, it is not immediately clear how their work can be extended for other important fairness definitions. In their model, given true risk scores of the defendants, judges only apply different thresholds for black and white defendants to predict recidivism. The threshold parameters alone can't capture unfairness measures such as unequal error rates. Even if one does improvise the model with more parameters, it remains an open question whether the theoretical conjectures made in the paper are still likely. This is because the conjectures assume that every time a judge gives a decision, the model parameters of the judge are updated. This becomes an issue with error rate parameters since the ground truth labels are not revealed for all tasks in crowdsourcing. [84] considers a different but related problem of bias resulting from adaptive data gathering (when the choice of whether to collect more data of a given type depends on the data already collected) and propose a differentially private data collection process as a solution.

There is also a lot of work on task assignment in crowdsourcing, which doesn't consider fairness. [106] proposes a greedy knapsack approach to satisfy limits on budget and the number of tasks any worker can solve. [45, 47, 62] consider task assignment problem when workers arrive online. [10] proposes optimal gold task assignment when workers' diligence change over time.

Beyond data collection, there is also recent work on making algorithms fair and robust to bias in the training data [27, 39, 42, 66, 118] and on correcting bias in training datasets [11, 31]. Correcting bias in a given dataset requires modifying the feature values and/or the labels in the dataset. In this chapter, we aim to collect unbiased and high quality dataset to begin with, relaxing the responsibility and the overhead of such post-processing from data users (for example, data scientists and machine learning engineers).

### 6.1.2 Our Contributions

In this chapter, we make the following contributions:

1. We propose a novel algorithm for assigning tasks to workers, which optimizes the expected accuracy of labels obtained from crowd while ensuring that the collected labels satisfy desired notions of *error fairness*. The algorithm also ensures diversity of responses by limiting the probability of assigning many tasks to a single worker. Our algorithm works even when the values of the sensitive attribute of the tasks are unavailable or can't be used because of ethical/legal reasons.
2. With a novel formulation of the task assignment strategy as a probability distribution over the workers, we can cast the optimization problem as a linear program and avoid the use of integer programming or other graph matching algorithms which are popular in the task assignment literature but are harder to solve exactly and analyze. This also

makes our algorithm suitable for online settings in which the requester is not aware of the tasks in advance.

3. We use a limited number of gold tasks (tasks with known ground-truth answers) for estimating workers' parameters and then optimally assign non-gold tasks to the workers. We provide performance bounds for our algorithm and show empirical performance on a real dataset.

### 6.2 Model

Let there be a finite set of  $n$  workers and a large pool of tasks with unknown ground truth labels. The data requester randomly chooses tasks from the pool one by one and assigns each to one (or more) worker(s). The requester may not have knowledge of all the tasks in the pool (not even the number of tasks in the pool) in advance. A worker  $i$  charges a constant amount of fee  $c_i$  for every label she provides. The requester has a budget constraint for the maximum *expected* money to be spent on acquiring one label from a worker.

Let  $Z$  be a random variable denoting the sensitive attribute and  $Y$  denoting the (unknown) ground truth labels of the tasks such that  $Z, Y \in \{0, 1\}$ . For the tasks attempted by a worker  $i$ , let  $\hat{Y}_i \in \{0, 1\}$  denote the labels submitted by the worker. We denote the realizations of random variables  $Z$ ,  $Y$  and  $\hat{Y}_i$  by lower case letters  $z$ ,  $y$  and  $\hat{y}_i$  respectively and will drop the subscripts for brevity when the context is clear. We will use  $[n]$  to denote  $\{1, 2, \dots, n\}$ . The workers are modeled using their accuracy matrices as follows:

**Definition 18** (Accuracy Matrices of a Worker). *The accuracy matrices  $\mathcal{A}_{iz}$ ,  $z \in \{0, 1\}$  of a worker  $i$  are two  $2 \times 2$  row stochastic matrices such that,  $\forall y, \hat{y}_i \in \{0, 1\}$ , the entry  $\mathcal{A}_{iz}[y, \hat{y}_i]$  is the probability of the worker's label on a task being  $\hat{y}_i$  given that the sensitive attribute of the task is  $z$  and the ground truth label is  $y$ .*

The two matrices  $\mathcal{A}_{i0}$  and  $\mathcal{A}_{i1}$  define the accuracy of the worker  $i$  for tasks belonging to the two different values of the sensitive attribute. The accuracy matrix model, also known as the Dawid-Skene model [21] in the crowdsourcing literature, is strong enough to capture different errors (for e.g. false positive and false negative rates) that a worker may make for tasks belonging to a given sensitive attribute value. If a worker is unbiased in the sense that her errors don't depend on the value of sensitive attribute of the task, her two accuracy matrices are identical. Note that this model makes an implicit i.i.d. assumption on a worker's answers.

The requester uses a probabilistic policy to assign the tasks to workers and collects the labels from the workers.

**Definition 19** (Crowdsourcing Policy). *A crowdsourcing policy is an  $n$ -dimensional stochastic vector  $S$ , such that an element  $S[i]$ ,  $i \in [n]$  is the probability of assigning any task to worker  $i$ , regardless of the sensitive attribute value of the task.*

Note that the requester’s policy doesn’t depend on the value of the sensitive attribute of the task. This is an intentional modeling choice to deal with the situations in which the sensitive attribute values of the tasks may not be available. It may be due to missing data, privacy reasons or legal/ethical requirements of not using the sensitive attribute.

For any task, the requester randomly selects one (or more than one) worker(s) with probabilities specified by the crowdsourcing policy vector  $S$  and assigns the task to the selected worker(s). The labels collected from the workers are obviously not guaranteed to be error free. We can define the accuracy matrices of the crowdsourcing policy in the same way as we defined the accuracy matrices of workers.

**Definition 20** (Accuracy Matrices of a Crowdsourcing Policy). *The accuracy matrices  $\mathcal{A}_z$ ,  $z \in \{0, 1\}$  of a crowdsourcing policy are two  $2 \times 2$  row stochastic matrices such that,  $\forall y, \hat{y} \in \{0, 1\}$ , the entry  $\mathcal{A}_z[y, \hat{y}]$  is the probability that a crowdsourced label<sup>1</sup> for a task is  $\hat{y}$  given that the sensitive attribute of the task is  $z$  and the ground truth label is  $y$ .*

We use the letter  $\mathcal{A}$  to denote accuracy matrices of the crowdsourcing policy as well as that of the workers but readers can differentiate between the two by noting that  $\mathcal{A}$  has an additional subscript  $i$  when referring to the matrix of a worker  $i$ . It is easy to see that we can express the accuracy matrices of a policy in terms of the accuracy matrices of the workers as follows:

$$\mathcal{A}_z = \sum_{i=1}^n S[i] \cdot \mathcal{A}_{iz} \quad , \forall z \in \{0, 1\} \quad (6.1)$$

The requester is interested in finding a crowdsourcing policy that maximizes the expected accuracy of the collected labels while ensuring that the data is *fair*, *diverse* and is acquired within budget constraints.

Crowd diversity is a subjective property and is generally defined in terms of the demographics of crowdworkers. In this chapter, we work with a given set of crowdworkers and can’t control such a measure of diversity. For settings like these, we define diversity as follows:

**Definition 21** ( $\beta$ -Diverse Crowdsourcing Policy). *A crowdsourcing policy is called  $\beta$ -diverse if and only if  $\forall i \in [n]$ ,  $S[i]$  is upper bounded by  $\beta$ , where  $\beta$  is a diversity parameter such that  $0 \leq \beta < 1$ .*

This definition limits the influence of individual workers on the overall crowdsourced dataset and aims to distribute the influence across more workers.

Similar to diversity, fairness is also a subjective property. We use some standard definitions of fairness from the machine learning literature [7, 42, 118].

<sup>1</sup>We note that the accuracy of a crowdsourcing policy can also be defined in terms of aggregated label when multiple labels per task are collected. But such definitions depend on specific label aggregation algorithms used. However, in many cases, it may be sufficient to assume that the accuracy of a policy with aggregated labels is an increasing function of our accuracy, which is a reasonable assumption.

**Definition 22** (False Positive Rate Parity). *A crowdsourcing policy, with accuracy matrices  $\mathcal{A}_0$  and  $\mathcal{A}_1$ , is said to satisfy false positive rate parity if and only*

$$\mathcal{A}_0[0, 1] = \mathcal{A}_1[0, 1]$$

One can similarly define false negative rate parity, which requires  $\mathcal{A}_0[1, 0] = \mathcal{A}_1[1, 0]$ .

**Definition 23** (Error Rate Parity). *A crowdsourcing policy, with accuracy matrices  $\mathcal{A}_0$  and  $\mathcal{A}_1$ , is said to satisfy error rate parity if and only if it satisfies both false positive rate parity and false negative rate parity, i.e.*

$$\mathcal{A}_0 = \mathcal{A}_1$$

It is easy to see that if all workers are unbiased, any crowdsourcing policy satisfies the above fairness definitions and one only needs to select a policy that maximizes accuracy while satisfying budget and diversity constraints. In this chapter, we address the general problem scenario (when workers are not necessarily unbiased).

### 6.3 Finding Optimal Crowdsourcing Policy

Let's first assume that the accuracy matrices of all the workers are known and the requester is interested in finding the optimal crowdsourcing policy maximizing the expected accuracy under budget, fairness and diversity constraints. We model this as a constrained optimization problem. The objective function in the minimization problem is the negative of the expected accuracy of the policy variable  $S$ :

$$-\mathbb{E}[\mathcal{A}(S)] = - \sum_{z \in \{0,1\}} P(Z = z) \sum_{y \in \{0,1\}} P_z(Y = y) \sum_{i=1}^n S[i] \mathcal{A}_{iz}[y, y] \quad (6.2)$$

where  $P(Z = z)$  is the known prior probability that any random task in the pool will have sensitive attribute value equal to  $z$  and  $P_z(Y = y)$  is the known prior probability that any random task with sensitive attribute value  $z$  in the pool will have a ground truth label  $y$ .

Together with the fairness and diversity constraints, we get following optimization problem:

$$\begin{aligned} \arg \min_S \quad & - \sum_{z \in \{0,1\}} P(Z = z) \sum_{y \in \{0,1\}} P_z(Y = y) \sum_{i=1}^n S[i] \mathcal{A}_{iz}[y, y] \\ \text{subject to} \quad & \sum_{i=1}^n S[i] = 1 \\ & S[i] \geq 0, \forall i \in [n] \\ & S[i] \leq \beta, \forall i \in [n] \\ & \mathcal{A}_0[0, 1] - \mathcal{A}_1[0, 1] \leq \alpha \\ & -(\mathcal{A}_0[0, 1] - \mathcal{A}_1[0, 1]) \leq \alpha \\ & \sum_{i=1}^n S[i] \cdot c_i \leq C \end{aligned} \quad (6.3)$$

The first two constraints are due to the fact that the crowdsourcing policy vectors are probabilistic and so, all elements must be positive and sum to 1. The third is the diversity constraint as formalized in Definition 21. The forth and fifth constraints together are equivalent to  $|\mathcal{A}_0[0, 1] - \mathcal{A}_1[0, 1]| \leq \alpha$ . For  $\alpha = 0$ , we get the exact fairness constraint (false positive rate parity) as formalized in Definition 22. Other fairness constraints can also be similarly included. The last constraint is due to the maximum expected budget ( $C$ ) that can be spent on acquiring one answer from a worker.

### 6.3.1 Estimates of Worker Accuracy Matrices

Until now, we assumed that the accuracy matrices of the workers are known. However, in practice, we need to estimate them. As is common in the literature [86], we assume that the requester has some limited number of gold standard tasks. Gold tasks are the tasks for which the requester not only knows the sensitive attribute value  $z$  but also the ground truth label  $y$ . We use gold tasks to estimate unknown worker accuracy matrices. Estimating all the entries of the worker accuracy matrices requires that every worker answers some gold tasks of each “type” (the type of a task is specified by its ground truth answer and its sensitive attribute value). We assign  $N_g$  tasks of every type to each worker to estimate their accuracy matrices. This is further explained as follows. Let  $\hat{\mathcal{A}}_{iz}$  be the estimate of the worker accuracy matrices  $\mathcal{A}_{iz}$ ,  $\forall z \in \{0, 1\}$ . If a worker  $i$  answers  $k$  tasks correctly out of  $N_g$  gold tasks of type  $z = 1, y = 1$ , then

$$\hat{\mathcal{A}}_{i1}[1, 1] = \frac{k}{N_g} \text{ and } \hat{\mathcal{A}}_{i1}[1, 0] = 1 - \hat{\mathcal{A}}_{i1}[1, 1]$$

Similarly, if she answers  $k'$  tasks correctly out of  $N_g$  gold tasks of type  $z = 1, y = 0$ , then

$$\hat{\mathcal{A}}_{i1}[0, 0] = \frac{k'}{N_g} \text{ and } \hat{\mathcal{A}}_{i1}[0, 1] = 1 - \hat{\mathcal{A}}_{i1}[0, 0]$$

This process is repeated with gold tasks of type  $z = 0, y = 1$  and  $z = 0, y = 0$  to get estimates of all entries of the worker's matrices.

The optimization problem 6.3 can now be written as follows, by replacing the accuracy matrices with their estimates:

$$\begin{aligned} \arg \min_S \quad & - \sum_{z \in \{0, 1\}} P(Z = z) \sum_{y \in \{0, 1\}} P_z(Y = y) \sum_{i=1}^n S[i] \hat{\mathcal{A}}_{iz}[y, y] \\ \text{subject to} \quad & \sum_{i=1}^n S[i] = 1 \\ & S[i] \geq 0, \forall i \in [n] \\ & S[i] \leq \beta, \forall i \in [n] \\ & \hat{\mathcal{A}}_0[0, 1] - \hat{\mathcal{A}}_1[0, 1] \leq \alpha \\ & -(\hat{\mathcal{A}}_0[0, 1] - \hat{\mathcal{A}}_1[0, 1]) \leq \alpha \\ & \sum_{i=1}^n S[i] \cdot c_i \leq C \end{aligned} \tag{6.4}$$

where,

$$\hat{\mathcal{A}}_z = \sum_{i=1}^n S[i] \cdot \hat{\mathcal{A}}_{iz}, \quad \forall z \in \{0, 1\} \quad (6.5)$$

This is a linear program, which can be exactly solved in polynomial time. In practice, the simplex method [17] can be used to find the optimal solution efficiently with common optimization libraries like IBM CPLEX and SciPy. Depending on the constraints, the cost and the accuracy matrices of workers, it is possible that no feasible solution exists for the optimization problem. In this case, the requester will have no choice but to relax the constraints.

A summary of steps of our complete crowdsourcing algorithm below.

---

**Algorithm 1:** Crowdsourcing with Fairness, Diversity and Budget Constraints (CrowdFDB)

---

- 1 Assign  $N_g$  gold tasks of every type  $((z = 0, y = 0), (z = 0, y = 1), (z = 1, y = 0), (z = 1, y = 1))$  to every worker  $i$  in the provided set of  $n$  workers.
  - 2 Get estimate of every worker  $i$ 's accuracy matrices  $\hat{\mathcal{A}}_{i0}$  &  $\hat{\mathcal{A}}_{i1}$ .
  - 3 Solve the linear program to find the best crowdsourcing policy satisfying desired fairness, diversity and budget constraints.
  - 4 Pick a task randomly from the task pool of tasks with unknown ground truth labels. Randomly select one or more workers from the set of  $n$  workers, with probabilities specified by the crowdsourcing policy. Assign the task to the selected worker(s).
  - 5 Repeat Step 4 for all tasks in the pool.
- 

### 6.4 Theoretical Analysis

When worker accuracy matrices are known, our method is guaranteed to provide the optimal solution, satisfying constraints. However, when estimates of the accuracy matrices are used, two interesting questions arise:

1. Does the solution of problem 6.4 (which is optimal and satisfies fairness constraints only according to the estimated accuracy parameters) also satisfy fairness in reality?
2. How much does the requester lose in terms of actual expected accuracy of the policy because of using the estimated accuracy parameters in optimization?

**Theorem 13.** *With probability at least  $\gamma$ , the solution  $\hat{S}$  to the optimization problem 6.4 satisfies*

$$|\mathcal{A}_0[0, 1] - \mathcal{A}_1[0, 1]| \leq \alpha + \delta$$

where

$$\delta = 2\sqrt{\frac{-\ln(1 - \sqrt[2n]{\gamma}) + \ln 2}{2N_g}}; \mathcal{A}_z = \sum_{i=1}^n \hat{S}[i] \cdot \mathcal{A}_{iz}, \quad \forall z \in \{0, 1\}$$

and  $N_g$  is number of gold tasks.

The theorem states that when we use estimates of the worker accuracy matrices instead of the real matrices, the obtained solution  $\hat{S}$  doesn't violate the fairness constraints in reality by more than  $\delta$ , with probability at least  $\gamma$ .

**Theorem 14.** *Assuming that the optimal solution  $\hat{S}$  of problem 6.4 satisfies fairness constraints of problem 6.3 and the optimal solution  $S$  of problem 6.3 satisfies fairness constraints of problem 6.4, then with probability at least  $\gamma'$*

$$\mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(\hat{S})] \leq 2n\beta \sqrt{\frac{-\ln(1 - \sqrt[n]{\gamma'}) + \ln 2}{2N_g}}$$

where

$$\begin{aligned} \mathbb{E}[\mathcal{A}(S)] &= \sum_{z \in \{0,1\}} P(Z=z) \sum_{y \in \{0,1\}} P_z(Y=y) \sum_{i=1}^n S[i] \mathcal{A}_{iz}[y, y], \\ \mathbb{E}[\mathcal{A}(\hat{S})] &= \sum_{z \in \{0,1\}} P(Z=z) \sum_{y \in \{0,1\}} P_z(Y=y) \sum_{i=1}^n \hat{S}[i] \mathcal{A}_{iz}[y, y] \end{aligned}$$

The theorem provides an upper bound on the loss in real expected accuracy of the crowdsourcing policy, when we use the estimated worker matrices instead of the real accuracy matrices for optimization. Note that in both the theorems, the bounds get better with increasing number of gold tasks.

The proofs of the above theorem are not difficult and depend on a simple application of the Hoeffding's inequality. For the sake of completeness, we provide the proofs in the appendix.

## 6.5 Experimental Evaluation

### 6.5.1 Datasets

We use the following datasets in our experiments.

1. **Broward County Dataset** [94] : This dataset contains information about 7214 defendants arrested in Broward County, Florida between 2013 and 2014. The information includes race of the defendants among other non-sensitive attributes such as age, prior charges etc. The dataset also contains ground-truth whether the defendants recidivated within 2 years or not. There are 3696 black defendants and 2454 white defendants in the dataset and the base rate of recidivism is 51.43% among the black defendants and 39.36% among the white defendants.
2. **Crowd Judgment Dataset** : [26] randomly selected a subset of 1000 defendants from the Broward County dataset and asked 20 random workers on Amazon Mechanical Turk to predict recidivism for each individual. In total, 400 workers participated in their study

and each worker submitted answers for 50 different defendants. The dataset contains these answers collected from the crowd.

### Experiment Outline

The idea is to split the set of defendants into two sets. The first set acts as the gold standard set, which we use to estimate worker accuracy matrices. Once we have the estimates of the worker accuracy matrices, we can solve the optimization problem 6.4 and learn optimal crowdsourcing policy. We then pick non-gold defendants one by one and assign it to one of the 400 workers, randomly selected according to the policy. The workers' responses are then compared with the ground-truth label to evaluate fairness and accuracy of our crowdsourcing policy.

### Handling Limitations of Datasets

Unfortunately, none of the two datasets alone can be used for such experiment. The Broward County dataset contains ground truth labels but doesn't contain workers' answers. On the other hand, the Crowd Judgment dataset does contain worker answers but is very limited for the following reasons. In this dataset, tasks have already been assigned (randomly) to workers and for every defendant, we have responses of only a subset of 20 workers out of all 400 workers. If the crowdsourcing policy learned by our algorithm decides that a worker outside that subset of 20 workers should be assigned a task, then we will need to know the answer of that worker but the answer of this worker is not part of the dataset. The second reason is that every worker has submitted answers for 50 defendants, which is sufficient for getting good estimates of the accuracy parameters of the workers but not big enough to be further split into gold and non-gold sets.

To overcome these limitations, we first create a bigger synthetic dataset using the two real datasets as follows. We generate synthetic answers of all the 400 workers for all the 3696 black and 2454 white defendants in the Broward County dataset. The answers are generated using the worker accuracy parameters estimated from the entire Crowd Judgment dataset. Note that even though this is a synthetic dataset but none of the parameters of the dataset are synthetic. The worker accuracy parameters are derived from the entire real dataset of [26] and the base dataset (Broward County dataset) is used as it is. Indeed, this is not ideal but is perhaps the only option, given the limitations of the available datasets.

Worker Costs : The datasets also don't contain workers' costs. We create this information in two different ways. In the first setting, we associate a uniform cost of \$1 to each worker. In the second and more interesting setting, we probabilistically associate a cost of \$1 or \$3. The probability of a worker's cost being \$3 is equal to her average accuracy and of it being \$1 is equal to  $1 - \text{her average accuracy}$ . Thus, the higher the average accuracy of a worker, the higher is the probability that she will charge a cost of \$3.



Now this complete dataset is ready to be used in the experiment outlined earlier in this section. We compare our approach (called ‘CrowdFDB’ in the figures) with two baselines (called ‘Random’ and ‘Greedy’ [106]). The baselines are discussed below.

### 6.5.2 Baselines

1. **Random Policy** : In the random policy, all workers are equally likely to be selected (probability  $\frac{1}{400}$ ) for any task.
2. **Greedy Optimization [106]** : In this baseline, we first estimate worker accuracy matrices using gold tasks in exactly same way as we do in our algorithm. However, the optimization is done using a bounded knapsack algorithm instead of linear programming. This algorithm sorts the workers in decreasing order of their “density”, where density is defined as the ratio of the expected accuracy of a worker and her cost. The expected accuracy of a worker can be calculated in the same way as we do for our algorithm i.e.  $\sum_{z \in \{0,1\}} P(Z=z) \sum_{g \in \{0,1\}} P_z(Y=g) \hat{\mathcal{A}}_{iz}[y, y]$ . Then, the algorithm assigns as many tasks as possible to the highest density worker without violating the diversity constraint. If  $T$  is the total number of tasks to be assigned, then a worker can be assigned at most  $\beta T$  tasks. Note that, unlike our algorithm, this baseline has to know the total number of tasks in advance to enforce diversity constraint. Once this worker has reached its capacity, the algorithm starts assigning tasks to the worker with next highest density and continues this for all tasks. However, it may be noted that the Greedy approach was originally proposed for a bit different setting, in which there is an overall crowdsourcing budget and the goal of the requester is to get as many tasks done as possible in that budget, maximizing total utility/accuracy and respecting work limits of the workers.

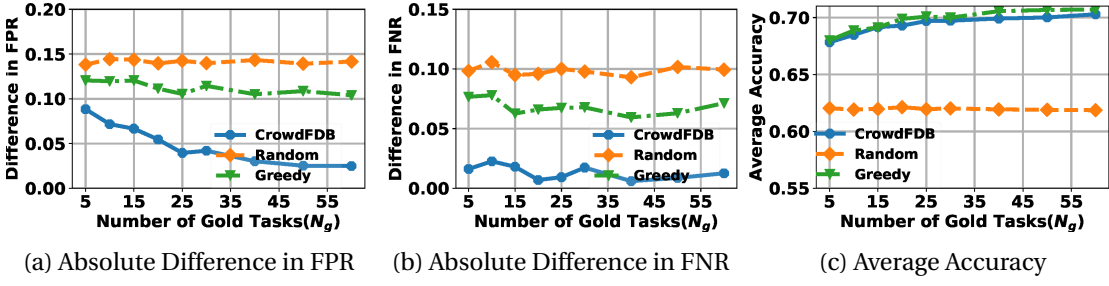
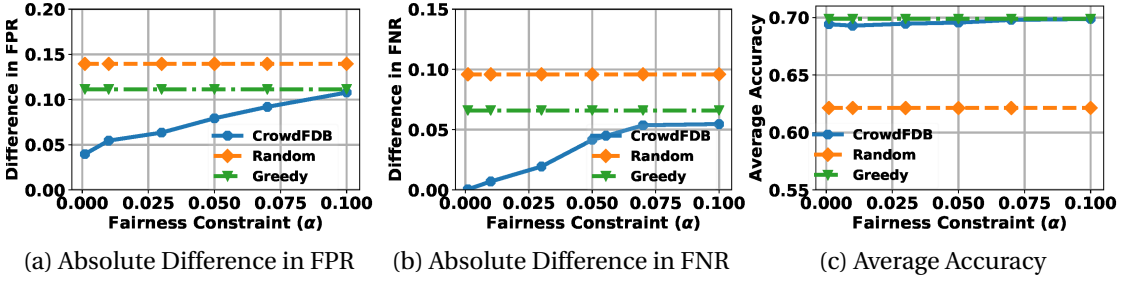
To give extra advantage to the baselines, we don’t put an *explicit* budget constraint for them and observe whether they can compete with the fairness and accuracy of our algorithm, which operates under budget constraint.

### 6.5.3 Observations

We use equal error rate parity (Definition 23) as the desired fairness. All results reported in the chapter are averages over 100 repeated runs. Parameter  $\beta$  was set to 0.01 in the first set of experiments. In the uniform costs settings,  $C$  was set to \$1 and in non-uniform settings,  $C = \$1.5$ .

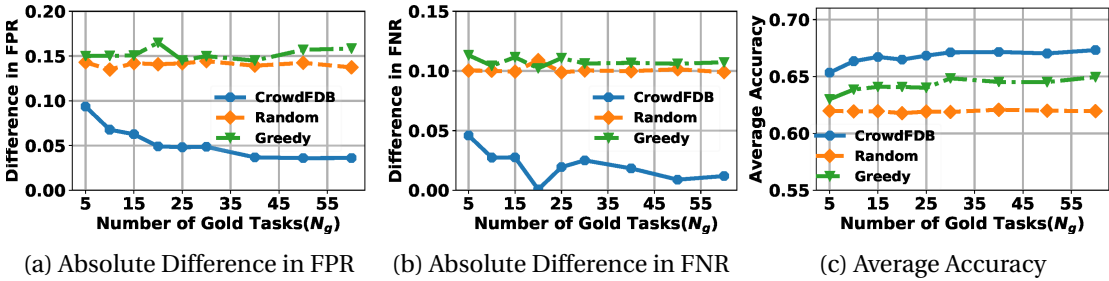
#### Uniform Costs

In Figure 6.1, we keep the fairness constraint  $\alpha$  to be fixed (0.01) and observe the effect of increasing number of gold tasks ( $N_g$ ). Figures 6.1a and 6.1b show that as we increase  $N_g$ , the fairness i.e. the absolute difference in FPR (and FNR) for black and white populations, gets closer and closer to  $\alpha$ . In other words, the  $\delta$  of Theorem 13 gets closer to 0 as expected.

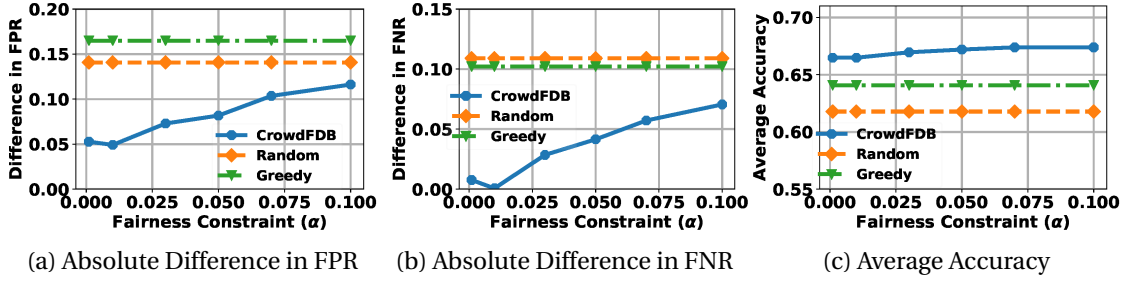

 Figure 6.1: Varying  $N_g$  (Number of gold tasks), Settings : Uniform Costs,  $\beta = 0.01, \alpha = 0.01$ 

 Figure 6.2: Varying  $\alpha$  (Fairness Constraint), Settings : Uniform Costs,  $\beta = 0.01, N_g = 20$ 

Moreover, the margin between our algorithm and the baselines also increases. However, meeting the fairness constraints alone is not enough. This could also be done by a bad algorithm that collects equally wrong labels for both white and black populations. Hence, accuracy of the collected labels is also an important measure. Figure 6.1c shows that our algorithm has an accuracy competitive to the Greedy baseline method, which is a highly efficient baseline in the literature for accuracy optimization. Our algorithm can achieve same level of accuracy while also providing fairness. In Figure 6.2, we keep  $N_g$  fixed (20) and observe the effect of increasing value of  $\alpha$ . As value of  $\alpha$  increases, the fairness constraints are more relaxed and the algorithm can obtain better accuracy.

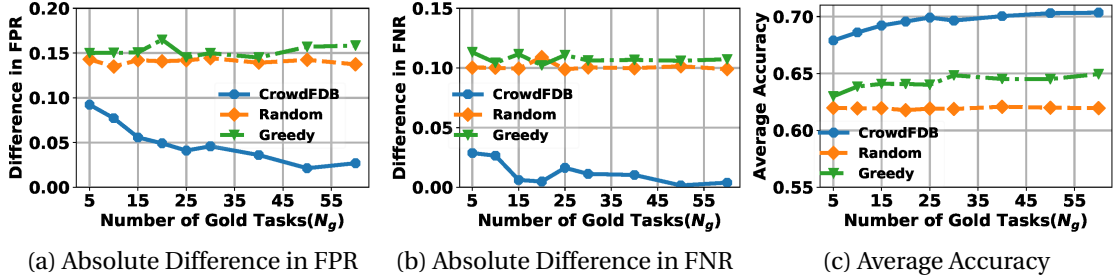
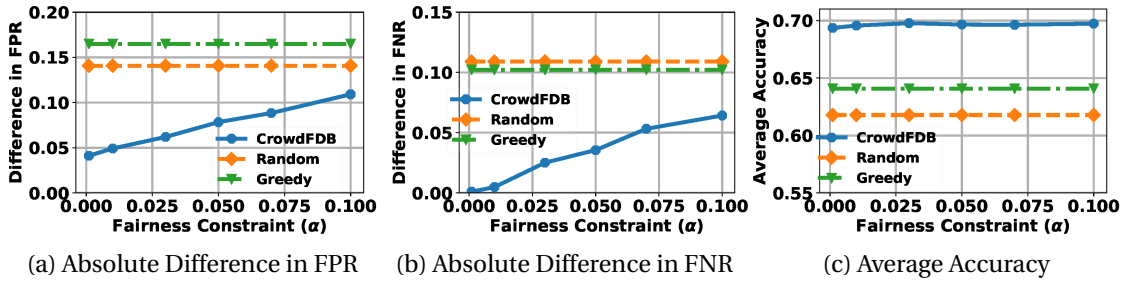
### Non-Uniform Costs


 Figure 6.3: Varying  $N_g$  (Number of gold tasks), Settings : Non-Uniform Costs,  $\beta = 0.01, \alpha = 0.01, C = 1.5$ 

In the non-uniform costs settings, we observe similar patterns in Figure 6.3 and 6.4. There are a few notable differences. The accuracy of our algorithm as well as the Greedy baseline

Figure 6.4: Varying  $\alpha$  (Fairness Constraint), Settings : Non-Uniform Costs,  $\beta = 0.01$ ,  $N_g = 20$ ,  $C = 1.5$ 

are lower. Our algorithm doesn't select more accurate workers because of budget constraints and the Greedy baseline also finds the density of the more accurate workers comparatively lower due to their high costs and prefers to choose other high density workers. In this case, our algorithm beats Greedy in not just fairness but also in accuracy by better utilizing the available budget.

Figure 6.5: Varying  $N_g$  (Number of gold tasks), Settings : Non-Uniform Costs,  $\beta = 0.01$ ,  $\alpha = 0.01$ ,  $C = 2.5$ Figure 6.6: Varying  $\alpha$  (Fairness Constraint), Settings : Non-Uniform Costs,  $\beta = 0.01$ ,  $N_g = 20$ ,  $C = 2.5$ 

Figures 6.5 and 6.6 show the results with  $\beta = 0.01$ ,  $C = 2.5$ . Figures 6.7 and 6.8 show the results with  $\beta = 0.005$ ,  $C = 2.5$ . Decreasing the value of parameter  $\beta$  makes the algorithms (ours and the Greedy baseline) more constrained in assigning tasks to the workers that they find to be better. This hits the accuracy of both the algorithms but the general trends discussed above (w.r.t. fairness and accuracy with different  $N_g$  and  $\alpha$ ) remain the same. The effect of increasing budget from 1.5 to 2.5 is that our algorithm can get better accuracy but there is no effect on the performance of other baselines as expected, since there was no explicit budget constraint placed on them.

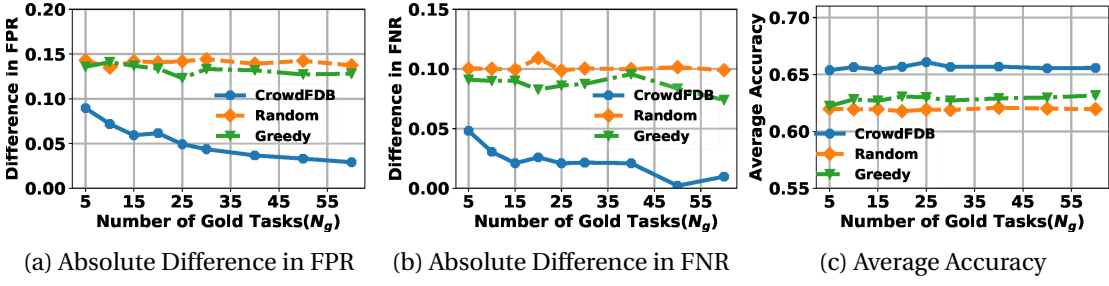


Figure 6.7: Varying  $N_g$  (Number of gold tasks), Settings : Non-Uniform Costs,  $\beta = 0.005$ ,  $\alpha = 0.01$ ,  $C = 2.5$

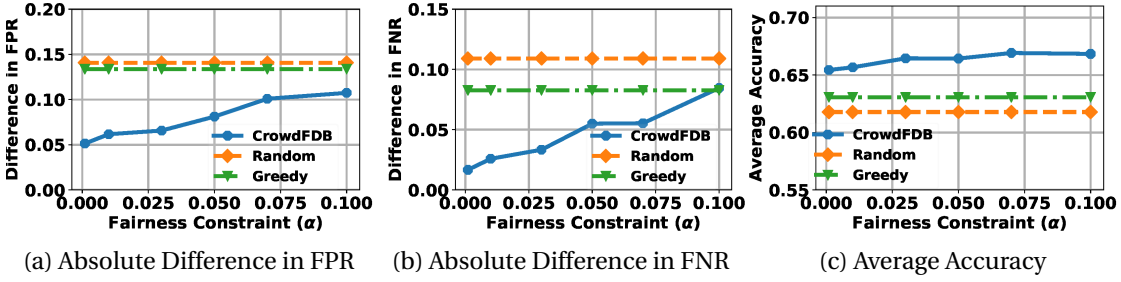


Figure 6.8: Varying  $\alpha$  (Fairness Constraint), Settings : Non-Uniform Costs,  $\beta = 0.005$ ,  $N_g = 20$ ,  $C = 2.5$

**Remark:** It is possible to change Definition 19 of the crowdsourcing policy such that the probability to assigning any task to a worker  $i$  depends on the sensitive attribute of the task. As noted earlier, we opted for the definition in which the policy doesn't depend on the sensitive attribute of the task due to legal, ethical, privacy or even missing data concerns. Empirically, we observed that a sensitive-attribute dependent policy doesn't really improve the accuracy by much as compared to our sensitive-attribute independent policy.

## 6.6 Chapter Summary

In this chapter, we addressed the problem of data fairness in crowdsourcing. We proposed a novel crowdsourcing algorithm that learns an optimal selection probability distribution over the available set of workers to maximize the expected accuracy of collected data, while ensuring that the errors in the data are not unfairly discriminatory towards any particular social group. There also remain many challenges to be addressed. These include estimating accuracy without requiring gold standard tasks, relaxing the assumption about knowledge of prior label distribution and the i.i.d. assumption about workers' answers.

Another interesting challenge in slightly different settings is to define data fairness for the case of subjective tasks, which have no ground truth labels and thus, no clear notion of errors. Ensuring fairness in subjective data collection is also likely to create a challenging problem of lying incentives for workers.

## 7 Concluding Remarks

Artificial intelligence has made a remarkable progress in the past few years and it has enabled a lot of new applications that were only possible in science fiction earlier. However, a wider social adoption of the technology will depend on whether it can be provably trustworthy and satisfy new regulations. With data playing a key role in the modern AI systems, it is imperative to be concerned about the trustworthiness of data and data collection mechanisms. In this thesis, we discussed in detail three core questions around this issue: how to get highest quality data for AI, how to ensure that the data collection process is transparent and fair for the data providers, and how to ensure that the collected data is fair for the users affected by the applications that make use of this data.

We presented two novel game theoretic mechanisms to elicit high quality data. The first is the Deep Bayesian Trust mechanism to collect high quality labels for tasks that have objective ground truth answers. The examples of such tasks include vision, natural language understanding, information retrieval etc. This mechanism is thus helpful, for example, in collecting data for supervised machine learning. This is the first peer based mechanism that ensures a dominant notion of incentive compatibility in non-binary answer space settings. This work is also the first to formally define a notion of fairness in crowd incentive mechanisms and propose a peer based mechanism satisfying the definition. The second mechanism is the Personalized Peer Truth Serum for collecting personal data such as health and activity data. This mechanism is thus useful, for example, in collecting data for unsupervised machine learning. This is the first peer based mechanism that doesn't need peer relationship to be defined based on shared tasks, but instead uses the data reported by people to find peer relationship.

We addressed the problem of building decentralized and transparent data oracles for blockchain applications. We identified major challenges in doing so - 1) getting truthful data from independent rational agents, particularly when agents have incentives originating from the applications to provide wrong data, and 2) implementing a truthful mechanism without requiring a central authority. We showed that peer-consistency mechanisms can be used to get truthful data for building such oracles in a profitable way. For the first time, we designed

and implemented a completely decentralized, trustless and transparent oracle using peer-consistency mechanisms in Ethereum. We addressed several non-trivial challenges that arise in implementing peer-consistency mechanisms in Ethereum and empirically evaluated the proposed optimizations to reduce gas costs.

Given that sharing economy platforms are increasingly being used as alternate and more accessible ways to earn livelihoods, we considered an important problem of discrimination on such platforms. Since reputation systems are often considered as a medium of building trust on web-based platforms and recent studies have found them useful in offsetting social biases, we focused on the challenge of making reputation systems more fair and non-discriminatory. We proposed two solutions for this: 1) a game-theoretic incentive mechanism to encourage providing more opportunities to the disadvantaged class and truthful reporting of the reviews of the received service, and 2) a post-aggregation bias correction technique to correct the bias in the reviews. The interesting observation about the incentive mechanism is that it doesn't give any unconditional advantage to the disadvantaged class (and thus avoid so-called 'reverse discrimination') since the rewards are tied to truthful reviews of the experienced service. An important consideration however is that the incentive mechanism must use the sensitive attribute information, instead of ignoring it. We hope that our proposed approaches will act as short-term interventions [49] to bring long-term and self-sustained fairness on such platforms.

Finally, we discussed the issue of preventing sensitive attribute-based bias of crowdworkers from appearing in the crowdsourced judgments. We showed that a novel task allocation algorithm can be used to satisfy fairness and diversity constraints in crowdsourcing under budget.

### **Future Directions:**

There remain several interesting directions for future work. In Chapter 2, there are two immediate questions: 1) can DUSIC be guaranteed with only a finite and small number of shared tasks between the agents?, and 2) can DUSIC be further strengthened to DSIC? Based on preliminary investigation, we conjecture that the answer to the first question is very likely to be positive. Having a smaller number of tasks introduces error in the estimated transitive trust and the error accumulates as the distance from the oracle increases. However, as long as this error is such that the agents can still be classified into three classes (agents with random answers, agents with answers positively correlated with ground truth, and agents with answers negatively correlated with ground truth) with sufficient accuracy, it is possible to ensure DUSIC. The issue about error accumulation can be fixed by using oracle after every few rounds of crowdsourcing. If the first question is answered positively, the second question becomes less significant from a practical perspective but is still interesting from theoretical perspective.

In Chapter 3, apart from the obvious questions about stronger game-theoretic guarantees, perhaps the most interesting open question is whether the proposed incentive scheme is easy to understand for real-world agents and if the agents react to it in the expected way. In

---

collaboration with behavioral economists, we are in the early stages of a large scale user-study to investigate this question formally. We have designed a user-study to elicit fitness data from people. Since generating and precisely measuring fitness data requires effort, an incentive scheme is required to collect high quality data.

In Chapter 4, we only considered a scenario where the outside incentives favor the same misreport for all agents, and do so with a particular dependence on the outcome. In ongoing work, we are considering different forms of outcome dependence, in particular threshold functions that require that the outcome exceeds a given threshold for the users to get refunds, and it turns out that these lead to different results. In the future, it would also be interesting to consider cases where agents have different and possibly opposing interests, such as in polls where different populations want different outcomes to win. Given that PTSC provides guarantees for non-binary signal spaces too, it would also be interesting to study similar problem beyond the binary answer setting.

In Chapter 5, an interesting open question is designing mechanisms to elicit truthful and fair text reviews. We also noted in this chapter that the sensitive attribute is required in our incentive mechanism for it to work in the desired way. While it is quite common in fair machine learning literature to use the sensitive attribute for ensuring fairness in decisions, it can sometimes be counterintuitive and difficult to explain from a legal and ethical perspective. It remains an interesting future work to design incentive schemes that don't require any knowledge about the sensitive attribute. Interestingly, the algorithm proposed in Chapter 6 doesn't require knowing the sensitive attribute of the tasks while assigning them to crowdworkers. We also noted that knowing the sensitive attribute had only a little additional advantage from an accuracy perspective. This observation further provides hope that it may be possible to design a similar sensitive-attribute independent solution for the problem of Chapter 5 as well.

While we viewed fairness in crowdsourced judgments as an independent objective in Chapter 6, it will also be interesting to see how fairness in crowdsourced data affects the fairness in algorithms and applications that eventually use that data.

Finally, we would like to emphasize the importance of viewing AI algorithms not as independent and isolated systems but as part of a bigger system where humans are the key players. To truly achieve our goal of building trustworthy AI, the complex nature of all the different directions [40] of the interaction between humans and AI must be taken into account.





# A Appendix

## A.1 Missing Proofs for Chapter 2

### A.1.1 Proof of Lemma 1

*Proof.* Let's first write the expression for the probability  $P(Y_j = y_j | Y_i = y_i)$  by applying Bayes' rule.

$$\begin{aligned} P(Y_j = y_j | Y_i = y_i) &= \sum_{g \in [K]} P(Y_j = y_j, G = g | Y_i = y_i) \\ &= \sum_{g \in [K]} P(Y_j = y_j | G = g, Y_i = y_i) \cdot P(G = g | Y_i = y_i) \end{aligned}$$

Since answers of  $i$  and  $j$  are conditionally independent given the ground truth, we have  $P(Y_j = y_j | G = g, Y_i = y_i) = P(Y_j = y_j | G = g)$ . This gives the following:

$$P(Y_j = y_j | Y_i = y_i) = \sum_{g \in [K]} P(Y_j = y_j | G = g) \cdot P(G = g | Y_i = y_i)$$

Now we apply the Bayes' rule again and expand the term  $P(G = g | Y_i = y_i)$ . This gives:

$$P(Y_j = y_j | Y_i = y_i) = \sum_{g \in [K]} P(Y_i = y_i | G = g) \cdot \frac{P(Y_j = y_j | G = g) \cdot P(G = g)}{P(Y_j = y_j)}$$

Note that  $P(Y_i = y_i | G = g) = T_i[g, y_i]$  and  $P(Y_j = y_j | G = g) = T_j[g, y_j]$ . We thus get,

$$P(Y_j = y_j | Y_i = y_i) = \sum_{g \in [K]} T_i[g, y_i] \cdot \frac{T_j[g, y_j] \cdot P(G = g)}{P(Y_j = y_j)}$$

Assuming  $|Q^i \cap Q^j| \rightarrow \infty$ , we now use the law of large numbers and the continuous mapping

theorem to replace  $P(Y_j = y_j | Y_i = y_i)$  with empirical distribution  $\omega(Y_j = y_j | Y_i = y_i)$  and  $P(Y_j = y_j)$  with empirical distribution  $\omega(Y_j = y_j)$ . This finally gives,

$$\omega(Y_i = y_i | Y_j = y_j) = \sum_{g \in [K]} T_i[g, y_i] \cdot \left( \frac{T_j[k, y_j] \cdot P(k)}{\omega(Y_j = y_j)} \right)$$

□

### A.1.2 Proof of Theorem 1

*Proof.* As  $|Q^i \cap Q^j| \rightarrow \infty$ , using lemma 1, the reward  $R'_i$  of any worker  $i$  in the Deep Bayesian Trust Mechanism is given by :

$$\begin{aligned} R'_i &= \beta \left[ \left( \sum_{g \in [K]} T_i[g, g] \right) - 1 \right] \\ &= \beta \left[ \left( \sum_{g \in [K]} \sum_{m \in [K]} A_i[g, m] S_i[m, g] \right) - 1 \right] \\ &\quad \text{(Using Proposition 1)} \end{aligned}$$

For binary answer space ( $K = 2$ ), this can be expanded as :

$$\begin{aligned} R_i &= \beta \left[ A_i[1, 1] S_i[1, 1] + A_i[1, 2] S_i[2, 1] + A_i[2, 1] S_i[1, 2] + A_i[2, 2] S_i[2, 2] - 1 \right] \\ &\quad \text{Rearranging the terms,} \\ R_i &= \beta \left[ \left[ A_i[1, 1] S_i[1, 1] + A_i[2, 1] S_i[1, 2] \right] + \left[ A_i[1, 2] S_i[2, 1] + A_i[2, 2] S_i[2, 2] \right] - 1 \right] \end{aligned}$$

Assuming  $A_i[1, 1] + A_i[2, 2] > 1$ s, we get that  $A_i[1, 1] > A_i[2, 1]$  and  $A_i[2, 2] > A_i[1, 2]$ .

Now, note that  $\left[ A_i[1, 1] S_i[1, 1] + A_i[2, 1] S_i[1, 2] \right]$  is a convex combination of  $A_i[1, 1]$  and  $A_i[1, 2]$  with  $S_i[1, 1]$  and  $S_i[2, 1]$  being the convex coefficients. Since  $A_i[1, 1] > A_i[2, 1]$ , this convex sum is maximized by using  $S_i[1, 1] = 1$  and  $S_i[1, 2] = 0$ . A similar argument follows for the second independent term in the reward. Thus, the total reward is maximized by the identity strategy matrix. The reward with  $S_i = I$  is thus,

$$R_i = \beta \left[ A_i[1, 1] + A_i[2, 2] - 1 \right]$$

which is strictly positive.

The above analysis implies that whenever worker does solve the tasks, it is her best reporting strategy to report the answer as they are. Now we just need to ensure that investing effort is also the best effort strategy. If the worker doesn't invest effort and report heuristically, the value

of the term  $\left(\sum_{g \in [K]} T_i[g, g]\right) - 1$  is 0. This will be proved in the proof of Theorem 2. However, the worker saves the cost of effort too in this case and she neither earns anything nor loses anything. But when worker invest effort, she earns  $R'_i$  from the mechanism and loses  $C^E$  in the form of cost of effort. For truthful strategy ( $e_i = 1, S_i = I$ ) to be the dominant uniform strategy, we need the following condition:

$$\beta \left[ A_i[1, 1] + A_i[2, 2] - 1 \right] - C^E > 0$$

This is true when

$$\beta > \frac{C^E}{A_i[1, 1] + A_i[2, 2] - 1}$$

**For non-binary answer space** ( $K > 2$ ), the proof follows similarly assuming  $A_i[k, k] > A_i[k', k], \forall k' \neq k$ . The reward for truthful strategy is  $R'_i = \beta \left[ \sum_{g \in [K]} A_i[g, g] - 1 \right]$  which is also strictly positive under the same assumption.  $\square$

### A.1.3 Proof of Theorem 2

*Proof.* As  $|Q^i \cap Q^j| \rightarrow \infty$ , using lemma 1, the reward  $R_i$  of any worker  $i$  in the Deep Bayesian Trust Mechanism is given by :

$$R'_i = \beta \left[ \left( \sum_{g \in [K]} T_i[g, g] \right) - 1 \right]$$

We know that a worker's strategy is called heuristic if either  $e_i = 0$  or  $e_i = 1$  and  $S_i$  has identical rows. In both cases, it is easy to see (using Proposition 1) that the sum of diagonal entries of her trustworthiness matrix sum to 1, which implies,

$$\left( \sum_{g \in [K]} T_i[g, g] \right) - 1 = 1 - 1 = 0$$

$\square$

### A.1.4 Proof of Theorem 3

*Proof.* The proof follows from Lemma 1, which ensures that reward of worker  $i$  converges in the limit to

$$\beta \cdot \left( \sum_{g \in [K]} T_i[g, g] \right) - 1$$

By definition,  $T_i[g, g], \forall g \in [K]$  measure the accuracy of the answers reported by the worker. Thus, even though the mechanism has only estimates of the accuracy of the workers' answers

and the estimates are indeed obtained using the answers of the peers and their trustworthiness but the consistency property of these estimates ensures the asymptotic fairness of the mechanism.  $\square$

## A.2 Missing Proofs for Chapter 3

### A.2.1 Proof of Theorem 4

*Proof.* The attribute score of agent  $i$  is given by :

$$\log \frac{f(y|\hat{\mu}_{L_{ij}}, \hat{\sigma}_{L_{ij}}^2)}{\sum_{k=1}^K \hat{\alpha}_k \cdot f(y|\hat{\mu}_{L_{kj}}, \hat{\sigma}_{L_{kj}}^2)}$$

Given that all other agents report truthfully, the attribute score becomes :

$$\log \frac{f(y|\mu_{L_{ij}}, \sigma_{L_{ij}}^2)}{\sum_{k=1}^K \alpha_k \cdot f(y|\mu_{L_{kj}}, \sigma_{L_{kj}}^2)}$$

This is because the maximum likelihood estimates  $\{\hat{\mu}_{L_{ij}}, \hat{\sigma}_{L_{ij}}^2\}$  converge to  $\{\mu_{L_{ij}}, \sigma_{L_{ij}}^2\}$  as  $n, |N_i| \rightarrow \infty$  under the assumptions of conditional independence and statistical similarity. Similarly for  $\hat{\alpha}_k$ . We can write the score as:

$$r_{ij} = \log \frac{P(X_{ij} = y|L_{ij})}{P(X_{ij} = y|G_j)}$$

The expected attribute score  $R$  of agent  $i$ , who observed  $X_{ij} = x$  and reported  $X_{ij} = y$  is then:

$$R = \int_{L_{ij}, G_j} P(L_{ij}, G_j | X_{ij} = x) \log \frac{P(X_{ij} = y | L_{ij})}{P(X_{ij} = y | G_j)} dL_{ij} dG_j$$

where  $P(L_{ij}, G_j | X_{ij} = x)$  is agent's posterior belief about  $L_{ij}$  and  $G_j$  conditional on observing  $X_{ij} = x$ . Under the assumption that attribute value  $X_{ij}$  is conditionally independent of global factors  $G_j$  given the personal factors  $L_{ij}$ , i.e.,  $P(X_{ij} = y | L_{ij}) = P(X_{ij} = y | G_j, L_{ij})$ , we get

$$R = \int_{L_{ij}, G_j} P(L_{ij}, G_j | x) \cdot \log \frac{P(y | G_j, L_{ij})}{P(y | G_j)} dL_{ij} dG_j \quad (\text{A.1})$$

However, we know (using Bayes' rule) that,

$$\frac{P(y | G_j, L_{ij})}{P(y | G_j)} = \frac{P(L_{ij} | y, G_j)}{P(L_{ij} | G_j)} \quad (\text{A.2})$$

Using Equations A.1 and A.2,

$$\begin{aligned} R &= \int_{L_{ij}, G_j} P(L_{ij}, G_j | x) \cdot \log \frac{P(L_{ij} | y, G_j)}{P(L_{ij} | G_j)} dL_{ij} dG_j \\ &= \int_{L_{ij}, G_j} P(L_{ij} | x, G_j) \cdot P(G_j | x) \log \frac{P(L_{ij} | y, G_j) \cdot P(L_{ij} | x, G_j)}{P(L_{ij} | G_j) \cdot P(L_{ij} | x, G_j)} dL_{ij} dG_j \end{aligned}$$

which can be rearranged as,

$$\begin{aligned} R &= \int_{G_j} P(G_j | x) \left[ \int_{L_{ij}} -P(L_{ij} | x, G_j) \log \frac{P(L_{ij} | x, G_j)}{P(L_{ij} | y, G_j)} dL_{ij} \right. \\ &\quad \left. + \int_{L_{ij}} P(L_{ij} | x, G_j) \log \frac{P(L_{ij} | x, G_j)}{P(L_{ij} | G_j)} dL_{ij} \right] dG_j \quad (\text{A.3}) \end{aligned}$$

for brevity,

$$R = \int_{G_j} P(G_j | x) [-KL_1 + KL_2] dG_j$$

where,  $KL_1$  and  $KL_2$  are KL-divergences and hence non-negative. It is easy to see that  $R$  is uniquely maximized when  $KL_1 = 0$ , which happens only when  $y = x$ . The expected attribute score at  $y = x$  is

$$R_{Truth} = \int_{G_j} P(G_j | x) KL_2 dG_j \quad (\text{A.4})$$

which is strictly positive. □

### A.2.2 Proof of Theorem 5

*Proof.* It is easy to see that if agents collude globally to report attribute  $j$  according to a certain strategic probability distribution  $\pi_{heur}$  independent of their clusters, then both numerator and denominator in the log term will converge to some common values. The attribute score of agent  $i$  is then given by :

$$r_{ij} = \log 1 = 0$$

□

### A.2.3 Proof of Theorem 6

*Proof.* When others are not truthful and report according to function  $g$ , the attribute score of agent  $i$  for reporting  $y$  as the value of  $X_{ij}$  is given by :

$$\log \frac{f(y|\bar{\mu}_{L_{ij}}, \bar{\sigma}_{L_{ij}}^2)}{\sum_{k=1}^K \alpha_k \cdot f(y|\bar{\mu}_{L_{kj}}, \bar{\sigma}_{L_{kj}}^2)}$$

This is because the maximum likelihood estimates  $\{\hat{\mu}_{L_{ij}}, \hat{\sigma}_{L_{ij}}^2\}$  converge to  $\{\bar{\mu}_{L_{ij}}, \bar{\sigma}_{L_{ij}}^2\}$  in this case, where,  $\bar{\mu}_{L_{ij}} = a \cdot \mu_{L_{ij}} + b$  and  $\bar{\sigma}_{L_{ij}}^2 = a^2 \cdot \sigma_{L_{ij}}^2$ . We can also write this as :

$$r_{ij} = \log \frac{P(g^{-1}(y)|L_{ij})}{P(g^{-1}(y)|G_j)}$$

where  $g^{-1}(y) = \frac{y-b}{a}$ . Using a procedure similar to that followed in the proof of Theorem 1, it can be shown that the expected attribute score of agent  $i$ , who observed  $X_{ij} = x$  and reported  $X_{ij} = y$  is given by :

$$R = \int_{G_j} P(G_j|x) \left[ \int_{L_{ij}} -P(L_{ij}|x, G_j) \log \frac{P(L_{ij}|x, G_j)}{P(L_{ij}|g^{-1}(y), G_j)} dL_{ij} + \int_{L_{ij}} P(L_{ij}|x, G_j) \log \frac{P(L_{ij}|x, G_j)}{P(L_{ij}|G_j)} dL_{ij} \right] dG_j$$

which is maximized when  $g^{-1}(y) = x$ , i.e.  $y = ax + b$ . In other words, it is the best response for agent  $i$  to also report according to  $g(x)$  when others do so. Moreover, at  $g^{-1}(y) = x$ , the expected attribute score is the same as  $R_{Truth}$  as obtained in Equation A.4.  $\square$

### A.2.4 Proof of Theorem 7

*Proof.* The expected score of agent  $i$  for truthfully reporting attribute  $j$ , as derived in Equation A.4 is

$$R_{Truth} = \int_{G_j} P(G_j|x) \int_{L_{ij}} P(L_{ij}|x, G_j) \log \frac{P(L_{ij}|x, G_j)}{P(L_{ij}|G_j)} dL_{ij} dG_j$$

Taking expectation over possible values of observations  $x$ , we find the ex-ante expected payment (EP) of agent  $i$  for telling truth.

$$\begin{aligned}
EP &= \int_x P(x) \int_{G_j} P(G_j|x) \int_{L_{ij}} P(L_{ij}|x, G_j) \log \frac{P(L_{ij}|x, G_j)}{P(L_{ij}|G_j)} dL_{ij} dG_j dx \\
&= \int_x \int_{G_j} P(x, G_j) \int_{L_{ij}} P(L_{ij}|x, G_j) \log \frac{P(L_{ij}|x, G_j)}{P(L_{ij}|G_j)} dL_{ij} dG_j dx \\
&= \int_x \int_{G_j} \int_{L_{ij}} P(x, G_j) P(L_{ij}|x, G_j) \log \frac{P(L_{ij}|x, G_j)}{P(L_{ij}|G_j)} dL_{ij} dG_j dx \\
&= \int_x \int_{G_j} \int_{L_{ij}} P(x, L_{ij}, G_j) \log \frac{P(L_{ij}|x, G_j)}{P(L_{ij}|G_j)} dL_{ij} dG_j dx \\
&= \int_x \int_{G_j} \int_{L_{ij}} P(x, L_{ij}, G_j) \log \frac{P(G_j) \cdot P(L_{ij}, x, G_j)}{P(x, G_j) \cdot P(L_{ij}, G_j)} dL_{ij} dG_j dx
\end{aligned}$$

Thus,

$$EP = I(X_{ij}; L_{ij}|G_j) \quad (\text{A.5})$$

where  $I(X_{ij}; L_{ij}|G_j)$  is the conditional mutual information of  $X_{ij}$  and  $L_{ij}$  given  $G_j$ .  $\square$

### A.2.5 Proof of Theorem 8

*Proof.* To prove this theorem, let's first assume that a clustering algorithm exists which can correctly assign each true report to the cluster from which it has been generated. In this case, if all other agents except  $i$  are truthful, the mechanism has access to the correct clusters of each of the truthful agents. The attribute score of agent  $i$  depends on which cluster the agent gets assigned to based on its report. If the agent gets assigned to its correct cluster, its expected attribute score from Equation A.3 is given by :

$$\begin{aligned}
R &= \int_{G_j} P(G_j|x) \left[ \int_{L_{ij}} -P(L_{ij}|x, G_j) \log \frac{P(L_{ij}|x, G_j)}{P(L_{ij}|y, G_j)} dL_{ij} \right. \\
&\quad \left. + \int_{L_{ij}} P(L_{ij}|x, G_j) \log \frac{P(L_{ij}|x, G_j)}{P(L_{ij}|G_j)} dL_{ij} \right] dG_j
\end{aligned}$$

which is already shown to be maximized in expectation by truthful reporting and is strictly positive. On the other hand, if the agent uses a misreport to get assigned to a different cluster, its expected attribute score is strictly negative. This is because the expected score from

Equation A.3 can be written as follows :

$$R = \int_{G_j} P(G_j|x) \left[ \int_{L'_{ij}} -P(L'_{ij}|x, G_j) \log \frac{P(L'_{ij}|x, G_j)}{P(L'_{ij}|y, G_j)} dL'_{ij} + \int_{L'_{ij}} P(L'_{ij}|x, G_j) \log \frac{P(L'_{ij}|x, G_j)}{P(L'_{ij}|y, G_j)} dL'_{ij} \right] dG_j$$

where  $L'_{ij}$  are the personal factors of the cluster that the agent gets assigned to. Note that  $x$  is conditionally independent of  $L'_{ij}$  given  $G_j$  i.e.  $P(L'_{ij}|x, G_j) = P(L'_{ij}|G_j)$ . This makes the only non-negative term in the above expression 0, leaving only with a term that is always negative for a misreport.

$$R = \int_{G_j} P(G_j|x) \int_{L'_{ij}} -P(L'_{ij}|x, G_j) \log \frac{P(L'_{ij}|x, G_j)}{P(L'_{ij}|y, G_j)} dL'_{ij} dG_j$$

We have assumed here that our always correct clustering algorithm assigns a non-truthful report to the correct cluster of its corresponding true observation with vanishing probability.

Hence, if others are truthful and a clustering algorithm exists to assign true reports to their correct clusters, it is the best strategy for agent  $i$  to also be truthful.

Now, it is easy to follow the clustering error tolerance of the mechanism from the above reasoning. Let's assume that the clustering algorithm makes an error  $\varepsilon$  in assigning any given true report to its correct cluster. We further assume that the error  $\varepsilon$  is small enough such that as  $|N_k| \rightarrow \infty, \forall k$ , the cluster parameters estimated are correct i.e.  $\hat{\alpha}_k, \forall$  converge to  $\alpha_k$  and  $\{\hat{\mu}_{kj}, \hat{\sigma}_{kj}^2\}, \forall k$  converge to  $\{\mu_{kj}, \sigma_{kj}^2\}$ .

If others are truthful, the expected attribute score of agent  $i$  under truthful response is given by :

$$R = (1 - \varepsilon)R_1 + \varepsilon R_2$$

Here,  $R_1$  is the expected score if the clustering algorithm assigns the agent to its correct cluster and  $R_2$  is the expected score if the clustering algorithm makes an error and assigns its to a wrong cluster. We have already seen that  $R_1$  is strictly positive and  $R_2$  is 0. Thus,  $R$  is strictly positive.

On the other hand, the expected attribute score under a non-truthful response is given by :

$$R^N = (1 - \varepsilon')R_1^N + \varepsilon' R_2^N$$

Here,  $1 - \varepsilon'$  is the probably of the clustering algorithm assigning a non-truthful report to the correct cluster of the corresponding true observation (this probability was assumed to vanish in the previous theorem).  $R_1^N$  is the expected score of the agent, if the report gets assigned



to the correct cluster of the corresponding true observation.  $R_2^N$  is the expected score of the agent, if the report doesn't get assigned to the correct cluster of the corresponding true observation. We already know that  $R_1^N$  is strictly less than  $R_1$  and  $R_N^2$  is strictly negative. Hence, only requiring  $(1 - \varepsilon) \geq (1 - \varepsilon')$  is sufficient to prove that truthful response is the best strategy. This is a reasonable requirement for any small error clustering algorithm i.e. it should assign a true observation to its correct cluster with higher or equal probability than that of assigning a mapping of the true observation. Note that this is not a tight requirement but a sufficient one. Even if such a requirement is not satisfied by the clustering algorithm, the agent has to find a mapping for which this requirement is not satisfied, which is almost impossible given only the true observation.  $\square$

### A.3 Missing Proofs for Chapter 4

#### A.3.1 Proof of Proposition 2

*Proof.* We know that relative saving is given by

$$\text{relative saving: } \frac{n\mathcal{R} - \mathcal{P}}{n\mathcal{R}},$$

where  $\mathcal{P}$  is the total expected payment (side-payment + outcome dependent payment) under the scheme to the agents. If the agents report honestly without getting any side-payment (i.e., the side-payment is 0), then the total expected payment  $\mathcal{P}$  in this setting will be  $n\mathcal{R}(1 - P(1))$ . This is because the only payments made in this setting are the outcome dependent payments and the expected value of the outcome is  $1 - P(1)$ , when everyone reports honestly. Thus,

$$\text{relative saving: } \frac{n\mathcal{R} - n\mathcal{R}(1 - P(1))}{n\mathcal{R}} = P(1)$$

This completes the proof.  $\square$

#### A.3.2 Proof of Theorem 9

*Proof.* By a standard convexity argument, it is sufficient to only consider pure strategies in this analysis, i.e. strategies where all the probability is placed on a single report. By the statement of the theorem, we know that  $\alpha > \mathcal{R}/(n \cdot \delta)$ , where  $\delta = \delta^* + \beta$ .

Let us first consider the case when  $\beta \leq 0$ . Note that if  $\beta \leq 0$ , then  $\alpha > \mathcal{R}/(n \cdot \delta)$  implies that  $\alpha > \mathcal{R}/(n \cdot \delta^*)$ . Let  $x$  be the signal observed and  $y$  be the signal reported by an agent in question. There are two cases :

**Case 1.**  $x = 0$ : We need to prove that for any agent  $i$ , given that other agents are truthful, the

total expected payment received by reporting 0 is more than the expected payment received by reporting 1 ,i.e., that

$$\begin{aligned} & \left( \frac{P_i(X_p = 1|X_i = 0)}{P_i(X_p = 1)} - 1 \right) + \mathcal{R} \left( \frac{(n-1)P_i(X_p = 0|X_i = 0)}{n} \right) \\ & < \alpha \left( \frac{P_i(X_p = 0|X_i = 0)}{P_i(X_p = 0)} - 1 \right) + \mathcal{R} \left( \frac{(n-1)P_i(X_p = 0|X_i = 0) + 1}{n} \right) \end{aligned}$$

This inequality is always true whenever  $\delta^* > 0$ , regardless of the value of  $\alpha$ .

**Case 2  $x = 1$ :** We need to prove that for any agent  $i$ , given that other agents are truthful, the total expected payment received by reporting 1 is more than the expected payment received by reporting 0, i.e., that

$$\begin{aligned} & \alpha \left( \frac{P_i(X_p = 0|X_i = 1)}{P_i(X_p = 0)} - 1 \right) + \mathcal{R} \left( \frac{(n-1)P_i(X_p = 0|X_i = 1) + 1}{n} \right) \\ & < \alpha \left( \frac{P_i(X_p = 1|X_i = 1)}{P_i(X_p = 1)} - 1 \right) + \mathcal{R} \left( \frac{(n-1)P_i(X_p = 0|X_i = 1)}{n} \right) \end{aligned}$$

Simplifying the above expression , we get the following requirement on the scaling constant  $\alpha$ :

$$\alpha > \frac{\mathcal{R}/n}{\frac{P_i(X_p=1|X_i=1)}{P_i(X_p=1)} - \frac{P_i(X_p=0|X_i=1)}{P_i(X_p=0)}} \Rightarrow \alpha > \frac{\mathcal{R}}{n\delta^*},$$

which is true since

$$\delta^* = \min_i \left( \frac{P_i(X_p = 1|X_i = 1)}{P_i(X_p = 1)} - \frac{P_i(X_p = 0|X_i = 1)}{P_i(X_p = 0)} \right).$$

This condition on  $\alpha$  is satisfied when  $\beta \leq 0$  as noted earlier. This completes the proof for the exact truth-telling equilibrium, when  $\beta \leq 0$ .

We will now prove the second part of the theorem which is approximate truth-telling equilibrium, when  $\beta > 0$ . As proved earlier, when  $x = 0$ , reporting truthfully is the best strategy regardless of the value of  $\alpha$  and therefore regardless also of the value of  $\beta$ . Hence, it suffices to argue about the case when  $x = 1$  and note that we only need to prove that  $\varepsilon = (\frac{\beta \cdot \mathcal{R}}{n \cdot \delta})$ , where  $\varepsilon$  is the maximum difference between the expected payment that the agent receives by reporting 1 and the expected payment that she receives by reporting 0; let  $\mathcal{D}$  denote this difference. We have that:

$$\begin{aligned}
\mathcal{D} &= \alpha \left( \frac{P_i(X_p = 0|X_i = 1)}{P_i(X_p = 0)} - 1 \right) + \mathcal{R} \left( \frac{(n-1)P_i(X_p = 0|X_i = 1) + 1}{n} \right) \\
&\quad - \alpha \left( \frac{P_i(X_p = 1|X_i = 1)}{P_i(X_p = 1)} - 1 \right) - \mathcal{R} \left( \frac{(n-1)P_i(X_p = 0|X_i = 1)}{n} \right) \\
&= -\alpha \left( \frac{P_i(X_p = 1|X_i = 1)}{P_i(X_p = 1)} - \frac{P_i(X_p = 0|X_i = 1)}{P_i(X_p = 0)} \right) + \frac{\mathcal{R}}{n} \\
&\leq -\alpha\delta^* + \frac{\mathcal{R}}{n}
\end{aligned}$$

We also know that  $\alpha > \mathcal{R}/(n\delta)$  and therefore,

$$\mathcal{D} \leq -\frac{\mathcal{R}}{n\delta}\delta^* + \frac{\mathcal{R}}{n} = \frac{\mathcal{R}}{n} \left( 1 - \frac{\delta^*}{\delta} \right) = \frac{\mathcal{R}\beta}{n\delta}$$

Since by definition,  $\varepsilon$  is the maximum value of  $\mathcal{D}$ , we get  $\varepsilon = (\mathcal{R}\beta)/(n\delta)$ . This completes the proof.  $\square$

### A.3.3 Proof of Theorem 10

*Proof.* We know that relative saving is given by

$$\text{relative saving: } \frac{n\mathcal{R} - \mathcal{P}}{n\mathcal{R}},$$

where  $\mathcal{P}$  is the total expected payment (side-payment + outcome dependent payment) under the scheme to the agents. Since we are interested in finding the relative saving in the truth-telling equilibrium of PTSC, let  $\mathcal{P}$  be the total expected payment (side-payment + outcome dependent payment) in the truth-telling equilibrium. We have that:

$$\mathcal{P} = n\alpha \left[ P(0) \left( \frac{P(0|0)}{P(0)} - 1 \right) + P(1) \left( \frac{P(1|1)}{P(1)} - 1 \right) \right] + n\mathcal{R}(1 - P(1)) \quad (\text{A.6})$$

The first term is due to expected side-payment in the truth-telling equilibrium and the second term is the outcome-dependent expected compensation payment in the truth-telling equilibrium. Here,  $P(0)$  and  $P(1)$  are the real probabilities that an agent observes 0 and 1 respectively for any random question. Moreover,  $P(0|0)$  is the real probability that for any question, given that one agent reported 0, the other also reports 0 and similarly,  $P(1|1)$  is the real probability that for any question, given that one agent reported 1, the other also reports 1. Note that the expectation in this case is not taken with respect to any agent's beliefs. Using this in the

expression for the relative saving, we have that

$$\begin{aligned}
 \text{relative saving} &= 1 - \frac{n\alpha \left[ P(0) \left( \frac{P(0|0)}{P(0)} - 1 \right) + P(1) \left( \frac{P(1|1)}{P(1)} - 1 \right) \right] + n\mathcal{R}(1 - P(1))}{n\mathcal{R}} \\
 &= P(1) - \frac{\alpha \left[ P(0) \left( \frac{P(0|0)}{P(0)} - 1 \right) + P(1) \left( \frac{P(1|1)}{P(1)} - 1 \right) \right]}{\mathcal{R}} \\
 &= P(1) - \frac{\alpha (P(0|0) + P(1|1) - 1)}{\mathcal{R}}
 \end{aligned}$$

Let  $\alpha = \mathcal{R}/(n\delta)$ . Then, we have that

$$\begin{aligned}
 \text{relative saving} &= P(1) - \frac{\frac{\mathcal{R}}{n\delta} (P(0|0) + P(1|1) - 1)}{\mathcal{R}} \\
 &\geq P(1) - \frac{1}{n\delta}
 \end{aligned}$$

The inequality holds because  $P(0|0) + P(1|1) - 1 \leq 1$ . This completes the proof.  $\square$

### A.3.4 Proof of Proposition 3

*Proof.* We first prove that the denial strategy remains an equilibrium strategy for the agents even after implementing PTSC as a side-payment mechanism. For this, we need to prove that, given that the other agents play the denial strategy, the best response for some agent  $i$  is to also play the denial strategy. In this case, the expected payment of agent  $i$  from PTSC is:

$$\begin{aligned}
 &\sigma_i(Y_i = 0|X_i = x) \cdot [\alpha(1/1 - 1)] + \sigma_i(Y_i = 1|X_i = x) \cdot 0 \\
 &\quad + \mathcal{R} \left[ \frac{\sigma_i(Y_i = 0|X_i = x) + (n-1)}{n} \right]
 \end{aligned}$$

where the first two terms are due to the expected side-payment given by PTSC and the third term is due to the expected outcome dependent payment, i.e., the expected compensation payment. Clearly, since all the remaining agents adopt the denial strategy, the payment due to PTSC is 0 regardless of the strategy of agent  $i$ , and therefore, her expected payment is simply

$$\mathcal{R} \left[ \frac{\sigma_i(Y_i = 0|X_i = x) + (n-1)}{n} \right]$$

This quantity is maximized for  $\sigma_i(Y_i = 0|X_i = x) = 1$  and the expected payment for  $\sigma_i(Y_i = 0|X_i = x) = 1$  is equal to  $\mathcal{R}$ .

It is easy to see that the expected payment that agents receive in the truthful equilibrium with  $\alpha = \mathcal{R}/(n\delta^*)$  is given by  $(\mathcal{R} \cdot P(1))/n$ . Therefore, the expected payment that agents get in the denial strategy equilibrium (i.e.,  $\mathcal{R}$ ) is clearly greater than the expected payment that the agents receive in the truth-telling equilibrium, if we assume that  $\alpha = \mathcal{R}/(n\delta^*)$ . One can

obviously make  $\mathcal{R}$  be smaller than the expected payment that agents receive in the truthful equilibrium by making  $\alpha$  larger.

Now, we prove the general result, i.e., that it is not possible for any side-payment mechanism to make the truth-telling equilibrium more profitable than the denial strategy equilibrium without incurring net loss; by net loss, we mean here that the relative saving is negative. This can be easily proved by contradiction: Assume that there exists a side-payment mechanism that makes the truth-telling equilibrium more profitable than the denial strategy equilibrium and yet achieves a positive relative saving. Since the truth-telling equilibrium is more profitable than the denial strategy equilibrium in this mechanism, the expected total payment  $\mathcal{P}$  in the truth-telling equilibrium must be strictly greater than  $n\mathcal{R}$ . This is because  $n\mathcal{R}$  is just the expected outcome-dependent payment that agents receive in the denial strategy equilibrium. Now looking at the expression for the relative saving, i.e.,

$$\text{relative saving: } \frac{n\mathcal{R} - \mathcal{P}}{n\mathcal{R}},$$

we can see that this is negative because, as reasoned above, it holds that  $\mathcal{P} > n\mathcal{R}$ , and we obtain a contradiction.  $\square$

### A.3.5 Proof of Lemma 2

*Proof.* It is sufficient to prove that  $(\delta_c^*)_i > 0$ , where

$$(\delta_c^*)_i = \left( \frac{P_i(X_p = 1|X_i = 1)}{P_i(X_p = 1)} - \frac{Q_i(X_p = 0|X_i = 1)}{Q_i(X_p = 0)} \right).$$

Replacing  $Q_i(\cdot|\cdot)$  and  $Q_i(\cdot)$  with their respective expressions, we obtain

$$\begin{aligned} (\delta_c^*)_i &= \left( \frac{P_i(X_p = 1|X_i = 1)}{P_i(X_p = 1)} - \frac{(1-f) + f \cdot P_i(X_p = 0|X_i = 1)}{(1-f) + f \cdot P_i(X_p = 0)} \right) \\ &= \left( \frac{f \cdot P_i(X_p = 1|X_i = 1)}{f \cdot P_i(X_p = 1)} - \frac{(1-f)}{(1-f) + f \cdot P_i(X_p = 0)} + \frac{f \cdot P_i(X_p = 0|X_i = 1)}{(1-f) + f \cdot P_i(X_p = 0)} \right) \end{aligned}$$

For brevity, we will use the following notation:

$$\begin{aligned} P_i(X_p = 1|X_i = 1) &= P \\ P_i(X_p = 0|X_i = 1) &= 1 - P \\ P_i(X_p = 1) &= P_1 \\ P_i(X_p = 0) &= 1 - P_1 \end{aligned}$$

We have that

$$\begin{aligned} (\delta_c^*)_i &= \left( \frac{f \cdot P}{f \cdot P_1} - \frac{(1-f)}{(1-f) + f \cdot (1-P_1)} + \frac{f \cdot (1-P)}{(1-f) + f \cdot (1-P_1)} \right) \\ &= f \left[ \frac{P}{f \cdot P_1} + \frac{(1-P)}{(1-f) + f \cdot (1-P_1)} \right] - \frac{(1-f)}{(1-f) + f \cdot (1-P_1)} \\ &= \frac{f}{(1-f) + f(1-P_1)} \left[ \frac{P-P_1}{P_1} \right] \end{aligned}$$

This term is always positive whenever  $\delta^* > 0$ . This completes the proof.  $\square$

### A.3.6 Proof of Theorem 11

*Proof.* Let us first consider the case when  $\beta_c \leq 0$ . Note that if  $\beta_c \leq 0$ , then  $\alpha > \mathcal{R}/(n \cdot \delta_c)$  implies that  $\alpha > \mathcal{R}/(n \cdot \delta_c^*)$ . Let  $x$  be the signal observed and  $y$  be the signal reported by an agent in question. There are two cases :

**Case 1.**  $x = 0$ : We need to prove that for any agent  $i$ , given that other agents are truthful, the total expected payment received by reporting 0 is more than the expected payment received by reporting 1 ,i.e.

$$\begin{aligned} &\alpha \cdot f \cdot \left( \frac{P_i(X_p = 1|X_i = 0)}{f \cdot P_i(X_p = 1)} - 1 \right) + \alpha \cdot (1-f)(0-1) \\ &< \alpha \cdot f \cdot \left( \frac{P_i(X_p = 0|X_i = 0)}{f \cdot P_i(X_p = 0) + (1-f)} - 1 \right) + \alpha \cdot (1-f) \left[ \frac{1}{f \cdot P_i(X_p = 0) + (1-f)} - 1 \right] + \frac{\mathcal{R}}{n} \end{aligned}$$

This inequality is always true whenever  $\delta_c^* > 0$ , regardless of the value of  $\alpha$ .

**Case 2**  $x = 1$ : We need to prove that for any agent  $i$ , given that other agents are truthful, the total expected payment received by reporting 1 is more than the expected payment received

by reporting 0, i.e., that

$$\begin{aligned} \alpha \cdot f \cdot \left( \frac{P_i(X_p = 0|X_i = 1)}{f \cdot P_i(X_p = 0) + (1-f)} - 1 \right) + \alpha \cdot (1-f) \cdot \left( \frac{1}{f \cdot P_i(X_p = 0) + (1-f)} - 1 \right) + \frac{\mathcal{R}}{n} \\ < \alpha \cdot f \cdot \left( \frac{P_i(X_p = 1|X_i = 1)}{f \cdot P(X_p = 1)} - 1 \right) + \alpha \cdot (1-f) \cdot (0-1) \end{aligned}$$

Simplifying the above expression, we get the following requirement on the scaling constant  $\alpha$ :

$$\alpha > \frac{\mathcal{R}/n}{\frac{P_i(X_p=1|X_i=1)}{P_i(X_p=1)} - \frac{Q_i(X_p=0|X_i=1)}{Q_i(X_p=0)}}$$

This is because  $(\delta_c^*)_i > 0$  as proved in Lemma 1. Moreover, since  $\delta_c^* = \min_i \left( \frac{P_i(X_p=1|X_i=1)}{P_i(X_p=1)} - \frac{Q_i(X_p=0|X_i=1)}{Q_i(X_p=0)} \right)$ , we get,

$$\alpha > \frac{\mathcal{R}}{n\delta_c^*},$$

This condition on  $\alpha$  is satisfied when  $\beta_c \leq 0$  as noted earlier. This completes the proof for the exact truth-telling equilibrium, when  $\beta_c \leq 0$ .

We will now prove the second part of the theorem which is approximate truth-telling equilibrium, when  $\beta_c > 0$ . As proved earlier, when  $x = 0$ , reporting truthfully is the best strategy regardless of the value of  $\alpha$  and therefore regardless also of the value of  $\beta_c$ . Hence, it suffices to argue about the case when  $x = 1$  and note that we only need to prove that  $\varepsilon = (\frac{\beta_c \cdot \mathcal{R}}{n \cdot \delta_c^*})$ , where  $\varepsilon$  is the maximum difference between the expected payment that the agent receives by reporting 1 and the expected payment that she receives by reporting 0; let  $\mathcal{D}$  denote this difference. We have that:

$$\begin{aligned} \mathcal{D} &= \alpha \cdot f \cdot \left( \frac{P_i(X_p = 0|X_i = 1)}{f \cdot P_i(X_p = 0) + (1-f)} - 1 \right) + \alpha \cdot (1-f) \cdot \left( \frac{1}{f \cdot P_i(X_p = 0) + (1-f)} - 1 \right) \\ &\quad + \frac{\mathcal{R}}{n} - \alpha \cdot f \cdot \left( \frac{P_i(X_p = 1|X_i = 1)}{f \cdot P(X_p = 1)} - 1 \right) - \alpha \cdot (1-f) \cdot (0-1) \\ &= -\alpha \left( \frac{P_i(X_p = 1|X_i = 1)}{P_i(X_p = 1)} - \frac{Q_i(X_p = 0|X_i = 1)}{Q_i(X_p = 0)} \right) + \frac{\mathcal{R}}{n} \\ &\leq -\alpha\delta_c^* + \frac{\mathcal{R}}{n} \end{aligned}$$

We also know that  $\alpha > \mathcal{R}/(n\delta_c)$  and therefore,

$$\mathcal{D} \leq -\frac{\mathcal{R}}{n\delta_c}\delta_c^* + \frac{\mathcal{R}}{n} = \frac{\mathcal{R}}{n} \left(1 - \frac{\delta_c^*}{\delta_c}\right) = \frac{\mathcal{R}\beta_c}{n\delta_c}$$

Since by definition,  $\varepsilon$  is the maximum value of  $\mathcal{D}$ , we get  $\varepsilon = (\mathcal{R}\beta_c)/(n\delta_c)$ . This completes the proof.  $\square$

### Relative Savings with Honest Agents

The baseline for computing relative saving now naturally becomes the rational outcome in which the honest agents report the truth and the remaining agents play according to their denial strategies. Thus, the saving of a side-payment scheme, under which a total payment of  $\mathcal{P}$  is made to *all* the agents (including the honest ones), now becomes:

$$\text{relative saving: } \frac{n\mathcal{R}' - \mathcal{P}}{n\mathcal{R}'},$$

where  $\mathcal{R}' = \mathcal{R} \cdot \left[(1-f) + f \cdot (1-P(1))\right]$ .

#### A.3.7 Proof of Theorem 12

*Proof.* We know that in this case, the relative saving is given by

$$\text{relative saving: } \frac{n\mathcal{R}' - \mathcal{P}}{n\mathcal{R}'},$$

where  $\mathcal{R}' = \mathcal{R} \cdot \left[(1-f) + f \cdot (1-P(1))\right]$  and  $\mathcal{P}$  is the total expected payment (side-payment + outcome dependent payment) under the scheme to the agents. Since we are interested in finding the relative saving in the truth-telling equilibrium of PTSC, let  $\mathcal{P}$  be the total expected payment (side-payment + outcome dependent payment) in the truth-telling equilibrium. We have that:

$$\mathcal{P} = n\alpha \left[ P(0) \left( \frac{P(0|0)}{P(0)} - 1 \right) + P(1) \left( \frac{P(1|1)}{P(1)} - 1 \right) \right] + n\mathcal{R}(1-P(1)) \quad (\text{A.7})$$

The first term is due to expected side-payment in the truth-telling equilibrium and the second term is the outcome-dependent expected compensation payment in the truth-telling equilibrium. Here,  $P(0)$  and  $P(1)$  are the real probabilities that an agent observes 0 and 1 respectively for any random question. Moreover,  $P(0|0)$  is the real probability that for any question, given that one agent reported 0, the other also reports 0 and similarly,  $P(1|1)$  is the real probability that for any question, given that one agent reported 1, the other also reports 1. Note that the expectation in this case is not taken with respect to any agent's beliefs. Using this in the



expression for the relative saving, we have that

$$\begin{aligned}
 \text{relative saving} &= 1 - \frac{n\alpha \left[ P(0) \left( \frac{P(0|0)}{P(0)} - 1 \right) + P(1) \left( \frac{P(1|1)}{P(1)} - 1 \right) \right] + n\mathcal{R}(1 - P(1))}{n\mathcal{R} \cdot \left[ (1 - f) + f \cdot (1 - P(1)) \right]} \\
 &= 1 - \frac{\alpha \left[ P(0) \left( \frac{P(0|0)}{P(0)} - 1 \right) + P(1) \left( \frac{P(1|1)}{P(1)} - 1 \right) \right]}{\mathcal{R} \cdot \left[ 1 - f \cdot P(1) \right]} - \frac{1 - P(1)}{1 - f \cdot P(1)} \\
 &= 1 - \frac{\alpha \left( P(0|0) + P(1|1) - 1 \right)}{\mathcal{R} \cdot \left[ 1 - f \cdot P(1) \right]} - \frac{1 - P(1)}{1 - f \cdot P(1)}
 \end{aligned}$$

Let  $\alpha = \mathcal{R}/(n\delta_c)$ . Then, we have that

$$\begin{aligned}
 \text{relative saving} &= 1 - \frac{\frac{\mathcal{R}}{n\delta_c} (P(0|0) + P(1|1) - 1)}{\mathcal{R} \cdot \left[ 1 - f \cdot P(1) \right]} - \frac{1 - P(1)}{1 - f \cdot P(1)} \\
 &\geq 1 - \frac{1}{n\delta_c \cdot (1 - f \cdot P(1))} - \frac{1 - P(1)}{1 - f \cdot P(1)}
 \end{aligned}$$

The inequality holds because  $P(0|0) + P(1|1) - 1 \leq 1$ . Simplifying the above, we get:

$$\text{relative saving} \geq \left[ (1 - f)P(1) - \frac{1}{n\delta_c} \right] \cdot \frac{1}{(1 - fP(1))}$$

This completes the proof.  $\square$

### Optimal Relative Saving

The maximum achievable relative saving is obtained when the agents turn truthful without any incentive mechanism (or a mechanism that pays 0 for truthful elicitation). In this case,  $\mathcal{P} = n\mathcal{R} \cdot P(0)$ . We can now calculate optimal relative saving by putting this  $\mathcal{P}$  in the above expression of relative saving:

$$\text{relative saving: } \frac{n\mathcal{R}' - n\mathcal{R} \cdot P(0)}{n\mathcal{R}'},$$

which equals to  $\frac{(1-f)P(1)}{1-fP(1)}$

Note that the relative saving derived in Theorem 4 approaches this value as  $n \rightarrow \infty$

## A.4 Missing Proofs for Chapter 6

We will use the Hoeffding's inequality [48] to prove our theorems. Let  $X_1, X_2, \dots, X_p$  be independent random variables bounded by the interval  $[0, 1] : 0 \leq X_j \leq 1$ . We define the empirical mean of these variables by

$$M_p = \frac{1}{p}(X_1 + X_2 + \dots + X_p)$$

then,

$$\begin{aligned} P(M_p \geq \mathbb{E}[M_p] + t) &\leq e^{-2nt^2} \\ P(M_p \leq \mathbb{E}[M_p] - t) &\leq e^{-2nt^2} \\ \text{and } P(|M_p - \mathbb{E}[M_p]| \geq t) &\leq 2e^{-2nt^2} \end{aligned}$$

### A.4.1 Proof of Theorem 13

*Proof.* Let the solution obtained from linear program with estimated worker matrices be  $\hat{S}$ . Then, we have

$$\mathcal{A}_0[0, 1] - \mathcal{A}_1[0, 1] = \sum_{i=1}^n \hat{S}[i] \mathcal{A}_{i0}[0, 1] - \sum_{i=1}^n \hat{S}[i] \mathcal{A}_{i1}[0, 1] = \sum_{i=1}^n \hat{S}[i] \left[ \mathcal{A}_{i0}[0, 1] - \mathcal{A}_{i1}[0, 1] \right] \quad (\text{A.8})$$

Now, using Hoeffding's inequality, we can say that with probability at least  $\gamma$ ,

$$\begin{aligned} \hat{\mathcal{A}}_{i0}[0, 1] &\leq \mathcal{A}_{i0}[0, 1] + \sqrt{\frac{-\ln \frac{1-\gamma}{2}}{2N_g}} \\ \text{and } \hat{\mathcal{A}}_{i0}[0, 1] &\geq \mathcal{A}_{i0}[0, 1] - \sqrt{\frac{-\ln \frac{1-\gamma}{2}}{2N_g}} \end{aligned}$$

We want to bound both  $\hat{\mathcal{A}}_{i0}[0, 1]$  and  $\hat{\mathcal{A}}_{i1}[0, 1]$  for all workers  $i \in \{1, 2, \dots, n\}$  in the same way. So, we can apply union bound and say that with probability at least  $\gamma$ , for every worker  $i$ ,

$$\begin{aligned}
 \hat{\mathcal{A}}_{i0}[0, 1] &\leq \mathcal{A}_{i0}[0, 1] + \sqrt{\frac{-\ln \frac{1-2\sqrt[q]{\gamma}}{2}}{2N_g}}, \\
 \hat{\mathcal{A}}_{i0}[0, 1] &\geq \mathcal{A}_{i0}[0, 1] - \sqrt{\frac{-\ln \frac{1-2\sqrt[q]{\gamma}}{2}}{2N_g}}, \\
 \hat{\mathcal{A}}_{i1}[0, 1] &\leq \mathcal{A}_{i1}[0, 1] + \sqrt{\frac{-\ln \frac{1-2\sqrt[q]{\gamma}}{2}}{2N_g}}, \\
 \text{and } \hat{\mathcal{A}}_{i1}[0, 1] &\geq \mathcal{A}_{i1}[0, 1] - \sqrt{\frac{-\ln \frac{1-2\sqrt[q]{\gamma}}{2}}{2N_g}},
 \end{aligned}$$

We now put these bounds in Equation A.8:

$$\begin{aligned}
 \mathcal{A}_0[0, 1] - \mathcal{A}_1[0, 1] &= \sum_{i=1}^n \hat{S}[i] \left[ \mathcal{A}_{i0}[0, 1] - \mathcal{A}_{i1}[0, 1] \right] \\
 &\leq \sum_{i=1}^n \hat{S}[i] \left[ \hat{\mathcal{A}}_{i0}[0, 1] + \sqrt{\frac{-\ln \frac{1-2\sqrt[q]{\gamma}}{2}}{2N_g}} - \left( \hat{\mathcal{A}}_{i1}[0, 1] - \sqrt{\frac{-\ln \frac{1-2\sqrt[q]{\gamma}}{2}}{2N_g}} \right) \right] \\
 &= \sum_{i=1}^n \hat{S}[i] \left[ \hat{\mathcal{A}}_{i0}[0, 1] - \hat{\mathcal{A}}_{i1}[0, 1] + 2\sqrt{\frac{-\ln \frac{1-2\sqrt[q]{\gamma}}{2}}{2N_g}} \right] \\
 &= \sum_{i=1}^n \hat{S}[i] \left[ \hat{\mathcal{A}}_{i0}[0, 1] - \hat{\mathcal{A}}_{i1}[0, 1] \right] + \sum_{i=1}^n \hat{S}[i] 2\sqrt{\frac{-\ln \frac{1-2\sqrt[q]{\gamma}}{2}}{2N_g}} \\
 &= \sum_{i=1}^n \hat{S}[i] \left[ \hat{\mathcal{A}}_{i0}[0, 1] - \hat{\mathcal{A}}_{i1}[0, 1] \right] + 2\sqrt{\frac{-\ln \frac{1-2\sqrt[q]{\gamma}}{2}}{2N_g}}
 \end{aligned}$$

We know that  $\hat{S}$  is the solution to the optimization problem with estimated accuracy matrices, thus it must satisfy

$$\sum_{i=1}^n \hat{S}[i] \left[ \hat{\mathcal{A}}_{i0}[0, 1] - \hat{\mathcal{A}}_{i1}[0, 1] \right] \leq \alpha$$

Putting this in our bound, we get

$$\mathcal{A}_0[0, 1] - \mathcal{A}_1[0, 1] \leq \alpha + 2\sqrt{\frac{-\ln \frac{1-2\sqrt[q]{\gamma}}{2}}{2N_g}} \tag{A.9}$$

We can similarly prove that

$$\mathcal{A}_0[0, 1] - \mathcal{A}_1[0, 1] \geq -\alpha - 2\sqrt{\frac{-\ln \frac{1-2\eta\sqrt{\gamma}}{2}}{2N_g}} \quad (\text{A.10})$$

Combining inequalities A.9 and A.10, we get

$$|\mathcal{A}_0[0, 1] - \mathcal{A}_1[0, 1]| \leq \alpha + 2\sqrt{\frac{-\ln \frac{1-2\eta\sqrt{\gamma}}{2}}{2N_g}} \quad (\text{A.11})$$

This completes the proof.  $\square$

#### A.4.2 Proof of Theorem 14

The difference in actual expected accuracy should be defined as:

$$\mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(\hat{S})]$$

where

$$\begin{aligned} \mathbb{E}[\mathcal{A}(S)] &= \sum_{z \in \{0,1\}} P(Z = z) \sum_{y \in \{0,1\}} P_z(Y = y) \sum_{i=1}^n S[i] \mathcal{A}_{iz}[y, y], \\ \mathbb{E}[\mathcal{A}(\hat{S})] &= \sum_{z \in \{0,1\}} P(Z = z) \sum_{y \in \{0,1\}} P_z(Y = y) \sum_{i=1}^n \hat{S}[i] \mathcal{A}_{iz}[y, y] \end{aligned}$$

We defined  $\mathbb{E}[\mathcal{A}(S)]$  as the expected accuracy of the optimal solution  $S$ , that would have been obtained if the real accuracy matrices were known and  $\mathbb{E}[\mathcal{A}(\hat{S})]$  as the expected accuracy of the obtained solution  $\hat{S}$ . Note that in both expressions, we use the real accuracy matrices of all workers and not the estimated matrices. This is because, we are reasoning about the difference in actual accuracy of the optimal solution and the actual accuracy of the obtained solution. Moreover, in the statement of the theorem, we made the following assumptions : the optimal solution  $\hat{S}$  of problem 7 satisfies fairness constraints of problem 5 and the optimal solution  $S$  of problem 5 satisfies fairness constraints of problem 7. This implies that the difference in expected accuracy is always positive because if both  $S$  and  $\hat{S}$  are feasible solutions, then  $S$  must have higher actual accuracy because  $S$  is the optimal solution among all feasible solutions. Now, we obtain an upper bound on this difference.

$$\begin{aligned} \mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(\hat{S})] &= \mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(\hat{S})] + 0 \\ &= \mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(\hat{S})] + \mathbb{E}[\hat{\mathcal{A}}(S)] - \mathbb{E}[\hat{\mathcal{A}}(S)] \end{aligned}$$

where we defined  $\mathbb{E}[\mathcal{A}(S)]$  as

$$\mathbb{E}[\mathcal{A}(S)] = \sum_{z \in \{0,1\}} P(Z = z) \sum_{y \in \{0,1\}} P_z(Y = y) \sum_{i=1}^n S[i] \hat{\mathcal{A}}_{iz}[y, y]$$

Let us also define  $\mathbb{E}[\mathcal{A}(\hat{S})]$  as

$$\mathbb{E}[\mathcal{A}(\hat{S})] = \sum_{z \in \{0,1\}} P(Z = z) \sum_{y \in \{0,1\}} P_z(Y = y) \sum_{i=1}^n \hat{S}[i] \hat{\mathcal{A}}_{iz}[y, y]$$

Note that

$$\mathbb{E}[\mathcal{A}(S)] \leq \mathbb{E}[\mathcal{A}(\hat{S})]$$

because  $\hat{S}$  is the optimal solution according to the estimated matrices among all feasible solutions.

Putting this inequality in the difference in expected accuracy,

$$\begin{aligned} \mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(\hat{S})] &= \mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(\hat{S})] + \mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(\hat{S})] \\ &\leq \mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(\hat{S})] + \mathbb{E}[\mathcal{A}(\hat{S})] - \mathbb{E}[\mathcal{A}(\hat{S})] \\ &= \mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(S)] + \mathbb{E}[\mathcal{A}(\hat{S})] - \mathbb{E}[\mathcal{A}(\hat{S})] \text{ (Rearrangement of terms)} \\ &= \sum_{z \in \{0,1\}} P(Z = z) \sum_{y \in \{0,1\}} P_z(Y = y) \left[ \sum_{i=1}^n S[i] (\mathcal{A}_{iz}[y, y] - \hat{\mathcal{A}}_{iz}[y, y]) \right. \\ &\quad \left. + \sum_{i=1}^n \hat{S}[i] (\hat{\mathcal{A}}_{iz}[y, y] - \mathcal{A}_{iz}[y, y]) \right] \end{aligned}$$

Here we use Hoeffding inequality in the same way as we did in the previous proof and use trivial bound of  $\beta$  for  $S[i]$  and  $\hat{S}[i]$ . This gives that with probability at least  $\gamma'$

$$\mathbb{E}[\mathcal{A}(S)] - \mathbb{E}[\mathcal{A}(\hat{S})] \leq 2n\beta \sqrt{\frac{-\ln \frac{(1-4n\sqrt{\gamma'})}{2}}{2N_g}}$$



## Bibliography

- [1] Guidance for regulation of artificial intelligence applications. *MEMORANDUM FOR THE HEADS OF EXECUTIVE DEPARTMENTS AND AGENCIES* (2019).
- [2] ABRAHAO, B., PARIGI, P., GUPTA, A., AND COOK, K. S. Reputation offsets trust judgments based on social biases among airbnb users. *Proceedings of the National Academy of Sciences* (2017), 201604234.
- [3] ACM. Statement on algorithmic transparency and accountability (2017).
- [4] ADLER, J., BERRYHILL, R., VENERIS, A., POULOS, Z., VEIRA, N., AND KASTANIA, A. As-traea: A decentralized blockchain oracle. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (Green-Com) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (2018), IEEE, pp. 1145–1152.
- [5] AGARWAL, A., MANDAL, D., PARKES, D. C., AND SHAH, N. Peer prediction with heterogeneous users. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC-2017)* (2017).
- [6] AN, B., XIAO, M., LIU, A., GAO, G., AND ZHAO, H. Truthful crowdsensed data trading based on reverse auction and blockchain. In *International Conference on Database Systems for Advanced Applications* (2019), Springer, pp. 292–309.
- [7] BAROCAS, S., HARDT, M., AND NARAYANAN, A. Fairness and machine learning : Limitations and opportunities.
- [8] BAROCAS, S., AND SELBST, A. D. Big data's disparate impact. *California Law Review* 671 (2016).
- [9] BOLUKBASI, T., CHANG, K.-W., ZOU, J. Y., SALIGRAMA, V., AND KALAI, A. T. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems* (2016).
- [10] BRAGG, J., WELD, D. S., ET AL. Optimal testing for crowd workers. In *International Conference on Autonomous Agents & Multiagent Systems* (2016).

## Bibliography

---

- [11] CALMON, F., WEI, D., VINZAMURI, B., RAMAMURTHY, K. N., AND VARSHNEY, K. R. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems* (2017).
- [12] CHAKRABORTY, M., AND DAS, S. Trading on a rigged game: Outcome manipulation in prediction markets. In *IJCAI* (2016).
- [13] CHAKRABORTY, M., DAS, S., LAVOIE, A., MAGDON-ISMAIL, M., AND NAAMAD, Y. Instructor rating markets. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence* (2013), AAAI Press, pp. 159–165.
- [14] CHARYTANOWICZ, M., NIEWCZAS, J., KULCZYCKI, P., KOWALSKI, P. A., ŁUKASIK, S., AND ŻAK, S. Complete gradient clustering algorithm for features analysis of x-ray images. In *Information technologies in biomedicine*. Springer, 2010, pp. 15–24.
- [15] CHECCO, A., BATES, J., AND DEMARTINI, G. All that glitters is gold—an attack scheme on gold questions in crowdsourcing. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* (2018).
- [16] CHEN, Y., GAO, X. A., GOLDSTEIN, R., AND KASH, I. A. Market manipulation with outside incentives. In *AAAI* (2011).
- [17] CHVATAL, V. *Linear programming*. Macmillan, 1983.
- [18] CORBETT-DAVIES, S., AND GOEL, S. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023* (2018).
- [19] COVER, T. M., AND THOMAS, J. A. Elements of information theory.
- [20] DASGUPTA, A., AND GHOSH, A. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web* (2013), ACM, pp. 319–330.
- [21] DAWID, A. P., AND SKENE, A. M. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics* (1979), 20–28.
- [22] DE ALFARO, L., FAELLA, M., POLYCHRONOPOULOS, V., AND SHAVLOVSKY, M. Incentives for truthful evaluations. *arXiv preprint arXiv:1608.07886* (2016).
- [23] DE VITO, S., MASSERA, E., PIGA, M., MARTINOTTO, L., AND DI FRANCIA, G. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical* 129, 2 (2008), 750–757.
- [24] DEPARTMENT OF INDUSTRY, SCIENCE, E., AND RESOURCES. Ai ethics principles. *Department of Industry, Science, Energy and Resources* (Sep 2019).
- [25] DIFALLAH, D. E., DEMARTINI, G., AND CUDRÉ-MAUROUX, P. Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In *WWW, CrowdSearch Workshop* (2012), pp. 26–30.



- 
- [26] DRESSEL, J., AND FARID, H. The accuracy, fairness, and limits of predicting recidivism. *Science advances* (2018).
  - [27] DWORK, C., HARDT, M., PITASSI, T., REINGOLD, O., AND ZEMEL, R. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* (2012), ACM, pp. 214–226.
  - [28] EDELMAN, B., LUCA, M., AND SVIRSKY, D. Racial discrimination in the sharing economy: Evidence from a field experiment. *American Economic Journal: Applied Economics* 9, 2 (April 2017), 1–22.
  - [29] EDELMAN, B. G., AND LUCA, M. Digital discrimination: The case of airbnb. com.
  - [30] FALTINGS, B., AND RADANOVIC, G. Game theory for data science: Eliciting truthful information. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 11, 2 (2017), 1–151.
  - [31] FELDMAN, M., FRIEDLER, S. A., MOELLER, J., SCHEIDEGGER, C., AND VENKATASUBRAMANIAN, S. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), ACM, pp. 259–268.
  - [32] FREEMAN, R., LAHAIE, S., AND PENNOCK, D. M. Crowdsourced outcome determination in prediction markets. In *AAAI* (2017).
  - [33] GAO, A., WRIGHT, J. R., AND LEYTON-BROWN, K. Incentivizing evaluation via limited access to ground truth: Peer-prediction makes things worse. In *2nd Workshop on Algorithmic Game Theory and Data Science at EC 2016*. (2016).
  - [34] GE, Y., KNITTEL, C. R., MACKENZIE, D., AND ZOEPE, S. Racial and gender discrimination in transportation network companies. Working Paper 22776, National Bureau of Economic Research, October 2016.
  - [35] GEBBIA, J. How airbnb designs for trust. *TED. com* (2016).
  - [36] GNEITING, T., AND RAFTERY, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* 102, 477 (2007), 359–378.
  - [37] GOEL, N., AND FALTINGS, B. Deep Bayesian Trust: A dominant and fair incentive mechanism for crowd. In *AAAI Conference on Artificial Intelligence* (2019).
  - [38] GOEL, N., VAN SCHREVEN, C., FILOS-RATSIKAS, A., AND FALTINGS, B. Infochain: A decentralized, trustless and transparent oracle on blockchain. *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)* (2020).
  - [39] GOEL, N., YAGHINI, M., AND FALTINGS, B. Non-discriminatory machine learning through convex fairness criteria. In *Proceedings of AAAI Conference on Artificial Intelligence* (2018).

## Bibliography

---

- [40] GREEN, B., AND CHEN, Y. The principles and limits of algorithm-in-the-loop decision making. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–24.
- [41] HANNÁK, A., WAGNER, C., GARCIA, D., MISLOVE, A., STROHMAIER, M., AND WILSON, C. Bias in online freelance marketplaces: Evidence from taskrabbit and fiverr. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (2017), ACM, pp. 1914–1933.
- [42] HARDT, M., PRICE, E., SREBRO, N., ET AL. Equality of opportunity in supervised learning. In *Advances in neural information processing systems* (2016).
- [43] HEINZ, G., PETERSON, L. J., JOHNSON, R. W., AND KERK, C. J. Exploring relationships in body dimensions. *Journal of Statistics Education* 11, 2 (2003).
- [44] HIRNSCHALL, C., SINGLA, A., TSCHIATSCHEK, S., AND KRAUSE, A. Learning user preferences to incentivize exploration in the sharing economy. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).
- [45] HO, C.-J., JABBARI, S., AND VAUGHAN, J. W. Adaptive task assignment for crowdsourced classification. In *Proceedings of ICML* (2013).
- [46] HO, C.-J., SLIVKINS, A., SURI, S., AND VAUGHAN, J. W. Incentivizing high quality crowdwork. In *Proceedings of the 24th International Conference on World Wide Web* (2015), ACM, pp. 419–429.
- [47] HO, C.-J., AND VAUGHAN, J. W. Online task assignment in crowdsourcing markets. In *Proceedings of AAAI Conference* (2012).
- [48] Hoeffding, W. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*. Springer, 1994, pp. 409–426.
- [49] HU, L., AND CHEN, Y. A short-term intervention for long-term fairness in the labor market. In *Proceedings of the 2018 World Wide Web Conference* (2018), International World Wide Web Conferences Steering Committee, pp. 1389–1398.
- [50] HU, N., PAVLOU, P. A., AND ZHANG, J. Can online reviews reveal a product’s true quality?: empirical findings and analytical modeling of online word-of-mouth communication. In *Proceedings of the 7th ACM conference on Electronic commerce* (2006), ACM, pp. 324–330.
- [51] IBM. Everyday ethics for artificial intelligence.
- [52] IPEIROTIS, P. G., PROVOST, F., AND WANG, J. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation* (2010), ACM, pp. 64–67.
- [53] JOHN PODESTA, PENNY PRITZKER, E. J. M. J. H., AND ZIENTS, J. Seizing opportunities and preserving values. *Executive Office of the President* (2014).

- 
- [54] JURCA, R., AND FALTINGS, B. Enforcing truthful strategies in incentive compatible reputation mechanisms. In *International Workshop on Internet and Network Economics* (2005), Springer, pp. 268–277.
  - [55] JURCA, R., AND FALTINGS, B. Mechanisms for making crowds truthful. *Journal of Artificial Intelligence Research* 34 (2009), 209–253.
  - [56] JURCA, R., AND FALTINGS, B. Incentives for answering hypothetical questions. Tech. rep., 2011.
  - [57] JURCA, R., FALTINGS, B., AND BINDER, W. Reliable qos monitoring based on client feedback. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 1003–1012.
  - [58] KALAI, A. T., MOITRA, A., AND VALIANT, G. Efficiently learning mixtures of two gaussians. In *Proceedings of the forty-second ACM symposium on Theory of computing* (2010), ACM, pp. 553–562.
  - [59] KAMAR, E., AND HORVITZ, E. Incentives for truthful reporting in crowdsourcing. In *Proceedings of the 11th international conference on autonomous agents and multiagent systems-volume 3* (2012), International Foundation for Autonomous Agents and Multiagent Systems, pp. 1329–1330.
  - [60] KAMBLE, V., SHAH, N., MARN, D., PAREKH, A., AND RAMACHANDRAN, K. Truth serums for massively crowdsourced evaluation tasks. *arXiv preprint arXiv:1507.07045* (2015).
  - [61] KARGER, D. R., OH, S., AND SHAH, D. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems* (2011), pp. 1953–1961.
  - [62] KARGER, D. R., OH, S., AND SHAH, D. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research* (2014), 1–24.
  - [63] KAY, M., MATUSZEK, C., AND MUNSON, S. A. Unequal representation and gender stereotypes in image search results for occupations. In *33rd Annual ACM Conference on Human Factors in Computing Systems* (2015).
  - [64] KITTUR, A., CHI, E. H., AND SUH, B. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems* (2008), ACM, pp. 453–456.
  - [65] KLEINBERG, J., MULLAINATHAN, S., AND RAGHAVAN, M. Inherent trade-offs in the fair determination of risk scores. *8th Innovations in Theoretical Computer Science Conference (ITCS)* (2017).
  - [66] KLEINBERG, J. M., MULLAINATHAN, S., AND RAGHAVAN, M. Inherent trade-offs in the fair determination of risk scores. In *Proceedings of ITCS* (2017).

## Bibliography

---

- [67] KOHAVI, R. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD* (1996), vol. 96, pp. 202–207.
- [68] KONG, Y., MA, Y., AND WU, Y. Securely trading unverifiable information without trust. *arXiv preprint arXiv:1903.07379* (2019).
- [69] KONG, Y., AND SCHOENEBECK, G. Equilibrium selection in information elicitation without verification via information monotonicity. *Proceedings of the 9th Innovations in Theoretical Computer Science Conference (ITCS)* (2018).
- [70] KONG, Y., AND SCHOENEBECK, G. An information theoretic framework for designing information elicitation mechanisms that reward truth-telling. *ACM Transactions on Economics and Computation* 7, 1 (Jan. 2019), 2:1–2:33.
- [71] KULSHRESTHA, J., ESLAMI, M., MESSIAS, J., ZAFAR, M. B., GHOSH, S., GUMMADI, K. P., AND KARAHALIOS, K. Quantifying search bias: Investigating sources of bias for political searches in social media. In *ACM Conference on Computer Supported Cooperative Work and Social Computing* (2017).
- [72] LI, M., WENG, J., YANG, A., LU, W., ZHANG, Y., HOU, L., LIU, J.-N., XIANG, Y., AND DENG, R. H. Crowdbc: A blockchain-based decentralized framework for crowdsourcing. *IEEE Transactions on Parallel and Distributed Systems* 30, 6 (2018), 1251–1266.
- [73] LI, Y., GAO, J., MENG, C., LI, Q., SU, L., ZHAO, B., FAN, W., AND HAN, J. A survey on truth discovery. *SIGKDD Explor. Newsl.* 17, 2 (2016), 1–16.
- [74] LIU, F. T., TING, K. M., AND ZHOU, Z.-H. Isolation forest. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on* (2008), IEEE, pp. 413–422.
- [75] LIU, Q., PENG, J., AND IHLER, A. T. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems* (2012), pp. 692–700.
- [76] LIU, Y., AND CHEN, Y. Machine-learning aided peer prediction. In *Proceedings of the 2017 ACM Conference on Economics and Computation* (2017), ACM, pp. 63–80.
- [77] LIU, Y., AND CHEN, Y. Surrogate scoring rules and a dominant truth serum for information elicitation. *arXiv preprint arXiv:1802.09158* (2018).
- [78] LU, Y., TANG, Q., AND WANG, G. Zebralancer: Private and anonymous crowdsourcing system atop open blockchain. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)* (2018), IEEE, pp. 853–865.
- [79] LYON, A. Why are normal distributions normal? *The British Journal for the Philosophy of Science* 65, 3 (2014), 621–649.
- [80] MAVRIDIS, P., GROSS-AMBLARD, D., AND MIKLÓS, Z. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *Proceedings of the 25th International Conference on World Wide Web* (2016), pp. 843–853.

- 
- [81] MICROSOFT. Responsible ai principles from microsoft. *Microsoft*.
- [82] MILLER, N., RESNICK, P., AND ZECKHAUSER, R. Eliciting informative feedback: The peer-prediction method. *Management Science* 51, 9 (2005), 1359–1373.
- [83] MOITRA, A., AND VALIANT, G. Settling the polynomial learnability of mixtures of gaussians. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on* (2010), IEEE, pp. 93–102.
- [84] NEEL, S., AND ROTH, A. Mitigating bias in adaptive data gathering via differential privacy. In *35th International Conference on Machine Learning* (2018).
- [85] NORTHPOINTE. Compas risk and need assessment system, url : [http://www.northpointeinc.com/files/downloads/faq\\_document.pdf](http://www.northpointeinc.com/files/downloads/faq_document.pdf).
- [86] OLESON, D., SOROKIN, A., LAUGHLIN, G. P., HESTER, V., LE, J., AND BIEWALD, L. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Human computation* 11, 11 (2011).
- [87] ON AI, H.-L. E. G. Ethics guidelines for trustworthy ai. *Shaping Europe's digital future - European Commission* (Apr 2019).
- [88] OTTERBACHER, J. Crowdsourcing stereotypes: Linguistic bias in metadata generated via gwap. In *Proceedings of ACM CHI* (2015).
- [89] OTTERBACHER, J. Linguistic bias in collaboratively produced biographies: Crowdsourcing social stereotypes? In *Proceedings of AAAI ICWSM* (2015).
- [90] OTTERBACHER, J., BATES, J., AND CLOUGH, P. Competent men and warm women: Gender stereotypes and backlash in image search results. In *ACM CHI* (2017).
- [91] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., ET AL. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12, Oct (2011), 2825–2830.
- [92] PODESTA, J., PRITZKER, P., MONIZ, E. J., HOLDREN, J., AND ZIENTS, J. Seizing opportunities and preserving values. *Executive Office of the President* (2014).
- [93] PRELEC, D. A bayesian truth serum for subjective data. *science* 306, 5695 (2004), 462–466.
- [94] PROPUBLICA. <https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis>.
- [95] RADANOVIC, G., AND FALTINGS, B. A robust bayesian truth serum for non-binary signals. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI' 13)* (2013), no. EPFL-CONF-197486, pp. 833–839.

## Bibliography


---


- [96] RADANOVIC, G., AND FALTINGS, B. Incentive schemes for participatory sensing. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems* (2015), International Foundation for Autonomous Agents and Multiagent Systems, pp. 1081–1089.
- [97] RADANOVIC, G., FALTINGS, B., AND JURCA, R. Incentives for effort in crowdsourcing using the peer truth serum. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7, 4 (2016), 48.
- [98] RAYKAR, V. C., YU, S., ZHAO, L. H., VALADEZ, G. H., FLORIN, C., BOGONI, L., AND MOY, L. Learning from crowds. *Journal of Machine Learning Research* 11, Apr (2010), 1297–1322.
- [99] ROUSSEEUW, P. J., AND DRIESSEN, K. V. A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41, 3 (1999), 212–223.
- [100] RUBINSTEIN, A. *Modeling bounded rationality*. MIT press, 1998.
- [101] RYAN, D. Calculating costs in ethereum contracts, 2017.
- [102] SCHMIDT, F. A. The good, the bad and the ugly: Why crowdsourcing needs ethics. In *Cloud and Green Computing (CGC), International Conference on* (2013), IEEE, pp. 531–535.
- [103] SCHÖLKOPF, B., PLATT, J. C., SHAWE-TAYLOR, J., SMOLA, A. J., AND WILLIAMSON, R. C. Estimating the support of a high-dimensional distribution. *Neural computation* 13, 7 (2001), 1443–1471.
- [104] SHNAYDER, V., AGARWAL, A., FRONGILLO, R., AND PARKES, D. C. Informed truthfulness in multi-task peer prediction. In *Proceedings of the 2016 ACM Conference on Economics and Computation* (2016), ACM, pp. 179–196.
- [105] SUROWIECKI, J. *The wisdom of crowds*. Anchor, 2005.
- [106] TRAN-THANH, L., STEIN, S., ROGERS, A., AND JENNINGS, N. R. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence* (2014).
- [107] VALERA, I., SINGLA, A., AND RODRIGUEZ, M. G. Enhancing the accuracy and fairness of human decision making. In *Advances in Neural Information Processing Systems* (2018), pp. 1774–1783.
- [108] VEJMEKKA, M., AND PALUŠ, M. Inferring the directionality of coupling with conditional mutual information. *Physical Review E* 77, 2 (2008), 026214.
- [109] VER STEEG, G. Non-parametric entropy estimation toolbox (npeet).
- [110] WAGGONER, B., AND CHEN, Y. Output agreement mechanisms and common knowledge. In *Second AAAI Conference on Human Computation and Crowdsourcing* (2014).

- [111] WAGUIH, D. A., GOEL, N., HAMMADY, H. M., AND BERTI-EQUILLE, L. Allegatortrack: Combining and reporting results of truth discovery from multi-source data. In *2015 IEEE 31st International Conference on Data Engineering* (2015), IEEE, pp. 1440–1443.
- [112] WAN, M., CHEN, X., KAPLAN, L. M., HAN, J., GAO, J., AND ZHAO, B. From truth discovery to trustworthy opinion discovery: An uncertainty-aware quantitative modeling approach. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016* (2016), pp. 1885–1894.
- [113] WITKOWSKI, J., AND PARKES, D. C. Peer prediction without a common prior. In *Proceedings of the 13th ACM Conference on Electronic Commerce* (2012), ACM, pp. 964–981.
- [114] WITKOWSKI, J., AND PARKES, D. C. A robust bayesian truth serum for small populations. In *AAAI Conference on Artificial Intelligence* (2012).
- [115] WOLFERS, J., AND ZITZEWITZ, E. Prediction markets. *Journal of economic perspectives* 18, 2 (2004), 107–126.
- [116] XIONG, W., AND XIONG, L. Smart contract based data trading mode using blockchain and machine learning. *IEEE Access* (2019).
- [117] ZAFAR, M. B., VALERA, I., RODRIGUEZ, M. G., AND GUMMADI, K. P. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *26th International World Wide Web Conference (WWW)* (2017).
- [118] ZAFAR, M. B., VALERA, I., RODRIGUEZ, M. G., AND GUMMADI, K. P. Fairness constraints: Mechanisms for fair classification. In *20th International Conference on Artificial Intelligence and Statistics (AISTATS)* (2017).
- [119] ZAFAR, M. B., VALERA, I., RODRIGUEZ, M. G., GUMMADI, K. P., AND WELLER, A. From parity to preference-based notions of fairness in classification. *Neural information processing systems* (2017).
- [120] ZHANG, J., SHENG, V. S., WU, J., FU, X., AND WU, X. Improving label quality in crowdsourcing using noise correction. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (2015), ACM, pp. 1931–1934.
- [121] ZHANG, Y., CHEN, X., ZHOU, D., AND JORDAN, M. I. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Advances in Neural Information Processing Systems* (2014), pp. 1260–1268.
- [122] ZHENG, Y., WANG, J., LI, G., CHENG, R., AND FENG, J. Qasca: a quality-aware task assignment system for crowdsourcing applications. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (2015), ACM, pp. 1031–1046.
- [123] ZHENG, Z., ZHANG, Y., AND LYU, M. R. Investigating qos of real-world web services. *IEEE transactions on services computing* 7, 1 (2014), 32–39.





CONTACT INFORMATION	EPFL IC IINFCOM LIA, INR 234 (Bâtiment INR) Station 14, CH-1015 Lausanne, Switzerland	<i>E-mail:</i> naman.goel@epfl.ch <a href="https://lia.epfl.ch/~goel/">https://lia.epfl.ch/~goel/</a>
RESEARCH INTERESTS	Artificial Intelligence, Algorithmic Fairness, Data Quality, Crowdsourcing, Game Theory	
EDUCATION	<b>École Polytechnique Fédérale de Lausanne (EPFL)</b> , Lausanne, Switzerland PhD, Computer and Communication Sciences      September 2015 - August 2020 (expected)	
	<b>Indian Institute of Technology (IIT)</b> , BHU, Varanasi, India Integrated Dual Degree, Computer Science and Engineering      2009 - 2014 <ul style="list-style-type: none"> <li>• CGPA: 9.10/10 (First Class with Honours)      Class Rank: 1</li> </ul>	
AWARDS 	12th ACM Web Science Conference, 2020 – <b>Best Paper Award</b> IARPA Geopolitical Forecasting Challenge 2019 – <i>Academic Award</i> , USD 7,500 IARPA Geopolitical Forecasting Challenge 2018 – <i>Multiple Awards</i> , USD 9,750 EPFL, <i>EDIC Fellowship</i> , 2015-2016 – CHF 51,000 IIT (BHU): <ul style="list-style-type: none"> <li>• <b>Institute Medal</b> – 2014</li> <li>• <i>Merit Scholarship</i> – 2010, 2011</li> <li>• <i>Ministry of HRD Scholarship</i> – 2013-14</li> </ul> IIT Kharagpur, <i>Senior Research Fellowship</i> , 2014	
RESEARCH EXPERIENCE	<b>École Polytechnique Fédérale de Lausanne (EPFL)</b> , Lausanne, Switzerland <i>Doctoral Assistant (PhD Candidate)</i> Sep 2016 - Present Artificial Intelligence Lab (Director: <i>Prof Boi Faltings</i> ) Areas of research: <ul style="list-style-type: none"> <li>• Game theoretical incentive schemes to elicit high quality information (crowdsourcing);</li> <li>• Fairness in algorithmic decision making.</li> </ul>	
	<b>Microsoft Research</b> , Bangalore, India <i>Research Intern</i> Sep 2019 - Nov 2019 Area of research: Fairness in algorithmic decision making.	
	<b>QCRI (Qatar Computing Research Institute)</b> , Doha, Qatar <i>Research Associate</i> Oct 2014 - Aug 2015 Data Analytics Group (Director: <i>Prof Divyakant Agrawal</i> ) Areas of research: <ul style="list-style-type: none"> <li>• Truth discovery from multiple conflicting sources of information;</li> <li>• Design of a data-centric platform for distributed machine learning.</li> </ul>	
PUBLICATIONS	<ul style="list-style-type: none"> <li>• <b>N Goel</b>, A Filos-Ratsikas, B Faltings. Peer-Prediction in the Presence of Outcome Dependent Lying Incentives. In Proceedings of the International Joint Conference on Artificial Intelligence (<b>IJCAI</b>), 2020.</li> <li>• <b>N Goel*</b>, C van Schreven*, A Filos-Ratsikas, B Faltings. Infochain: A Decentralized, Trustless and Transparent Oracle on Blockchain. In Proceedings of the International Joint Conference on Artificial Intelligence (<b>IJCAI</b>), 2020. Special Track on AI in FinTech.</li> </ul>	

- **N Goel**, R Maxime, B. Faltings. Tackling Peer-to-Peer Discrimination in the Sharing Economy.  **Best Paper Award**. In Proceedings of the ACM Web Science Conference (**WebSci**), 2020.
- **N Goel**, B Faltings. Personalized Peer Truth Serum for Eliciting Multi-Attribute Personal Data. In Proceedings of the Conference on Uncertainty in Artificial Intelligence (**UAI**), 2019.
- **N Goel**, B Faltings. Deep Bayesian Trust: A Dominant and Fair Incentive Mechanism for Crowd. In Proceedings of the AAAI Conference on Artificial Intelligence (**AAAI**), 2019.
- **N Goel**, B Faltings. Crowdsourcing with Fairness, Diversity and Budget Constraints. In Proceedings of the AAAI/ACM Conference on AI, Ethics and Society (**AAAI/ACM AIES**), 2019.
- **N Goel**, M Yaghini, B Faltings. Non-Discriminatory Machine Learning through Convex Fairness Criteria. In Proceedings of the AAAI Conference on Artificial Intelligence (**AAAI**), 2018.
- DA Waguih, **N Goel**, HM Hammady, L Berti-Équille. AllegatorTrack: Visualizing and Explaining Truth Discovery Results from Multi-source Data. In Proceedings of the IEEE International Conference on Data Engineering (**ICDE**), 2015.
- **N Goel**, D Agrawal, S Chawla, A Elmagarmid. Parameter Database: Data-centric Synchronization for Scalable Machine Learning. QCRI Technical Report, CoRR abs/1508.00703 (2015).

OTHER  
RESEARCH  
EXPERIENCE

**INRIA**, Rennes, Bretagne Atlantique, France

*Research Intern*

Feb 2014 - Apr 2014

ASAP Research Group (Director: *Prof Anne-Marie Kermarrec*)

Area of research: Distributed recommender systems.

**Center of Informatics, UFPE**, Recife, Brazil

*Summer Intern*

May 15, 2012 - July 16, 2012

Software Productivity Group (Director: *Prof Paulo Borba*)

TEACHING (TA)

CS-330, Intelligence Artificielle, EPFL

Spring 2019

CS-430, Intelligent Agents, EPFL

Fall 2018, Fall 2017

PREPA-031(a), Mathematics 1A, EPFL

Spring 2018

CS-251, Theory of Computation, EPFL

Spring 2017, Spring 2016

STUDENT  
SUPERVISION

7 M.S. and 2 B.S., EPFL

Spring 2017 - Spring 2020

SERVICE

PC Member (and Reviewer): AAAI 2020, IJCAI 2020, AAAI HCOMP 2020

Reviewer: NeurIPS 2020, PLOS ONE

Web co-chair: AAAI HCOMP 2019

Workshop PC Member (and Reviewer): Incentives in Machine Learning at ICML 2020