Agent Reputation and Reward Fairness in Peer-Based Crowdsourcing Mechanisms

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in ***Computer Science and Engineering*** *by Research*

by

Samhita Kanaparthy
2018121005
s.v.samhita@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
January, 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Agent Reputation and Reward Fairness in Peer-Based Crowdsourcing Mechanisms**" by **Samhita Kanaparthy**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Dr. Sujit P Gujar

To, my Mom and Dad,

who always assured me of their everlasting love and support throughout the challenging game called life.

Acknowledgements

I am thankful for the decision I made a few years back to join Machine Learning Lab (MLL). I am grateful to my advisor Prof. Sujit Gujar for providing me with this opportunity. He always closely mentored and gave valuable inputs, making research easily manageable. Despite having numerous ups and downs during my work, he was constantly cheerful, optimistic, and reassuring. I am incredibly grateful for his support and concern, not only in academic activities but also in other aspects. It was only because of him that I got to work on various problems with the best colleagues. I am grateful to have the opportunity to collaborate with Prof. Ravi Kiran and get the best exposure under him.

I owe my gratitude to my mentor Sankarshan Damle who has been there since the beginning and without whom my work would not have been possible. It was a pleasure to work with Sankarshan Damle and Manisha Padala, who were always helpful, enthusiastic, and up for discussions. They inspired me with their vibrant work culture. I sincerely thank Vijayraj Shanmugaraj, Anurag Jain, and Debojit Das for constantly reviewing all my submissions. I had the pleasure of mentoring and co-authoring with Sambhav Solanki. Special thanks to other labmates, Shaily Mishra, Sanjay Chandlekar, and Shantanu Das, for making MLL journey more enjoyable. I am thankful to Moin H Moti, Kritika Prakash, Kumar Abhishek and Shoeb Siddiqui for introducing me to MLL and Catan!

I am deeply indebted to my parents for providing me with their love and positivity in every situation without hesitation. I cannot thank them enough for constantly having my back and pumping me with confidence.

I would take an opportunity to sincerely thank all my teachers that had a role in shaping me into the person I am today. I specially thank Prof. Sadagopan Narasimhan and my friends Jayitha C, Harini R and Nitesh Narayana for encouraging me to get into IIIT Hyderabad.

I am also thankful for my friends Vishnu Teja, Pavithra Kondavarthi and my cousin Sharath Kanaparthy for always being a call away. I would be remiss in not thanking my college friends, Shivani Chepuri, Gowri Lekshmy, Sanjana Sunil, Vani Sancheti, Ananya Arun, Thejasvi Konduru, Aishwarya Srivastava, Ishita Vohra, Yash Chaurasia, Deeksha Devendra, Anurag Sahu for majorly supporting me and making college life absolute fun.

Lastly, I thank all the staff and administration for promptly helping me throughout the college.

Abstract

Crowdsourcing effectively solves a large variety of tasks by employing a distributed human population. Information aggregation from multiple reports provided by potentially unreliable or malicious agents is a primary challenge in crowdsourcing systems. As a result, research in this area has focused on incentivising agents to exert efforts and report truthfully. In particular, Peer Based Mechanisms (PBMs) appropriately reward agents for reporting accurately and truthfully. However, we observe that with PBMs, crowdsourcing systems may not be fair. As PBMs evaluate agents' reports based on their consistency with their peers, agents may not receive deserved rewards despite investing efforts and reporting truthfully. Unfair rewards for the agents may discourage participation. Motivated by this, we aim to build a general framework that assures fairness in PBMs. Towards this, we propose the idea of providing trustworthy agents with additional chances of pairing while evaluating their reports. Providing additional chances will help to reduce the penalty obtained by trustworthy agents from unfair pairings, improving their expected reward. To decide which agents to give additional chances we adopt a reputation model that quantifies agents' trustworthiness in the system. Based on this approach, we build a general *iterative* framework, REFORM, which adopts the reward scheme of any existing PBM and uses a suitable reputation model. To quantify fairness in PBMs, we introduce two general notions of fairness for PBMs, namely γ*-fairness* and *qualitative fairness*. γ-fairness is based on the proximity of the expected rewards a PBM assures to a truthful agent with the optimal reward it can provide. Qualitative fairness prioritises agents who consistently report accurate over other agents. In this work, we also consider that the tasks in the setting are time-sensitive. The task's requester expects agents to submit the task reports at the earliest. We refer to such a setting as *temporal settings*. In a temporal setting, the reputation model needs to consider both accuracy of reports and the time taken to report. However, no existing reputation models consider the time taken to report. Towards this, we introduce Temporal Reputation Model (TERM) to quantify an agent's trustworthiness in

a temporal setting. TERM assigns scores to the agents based on their reporting behaviour and the time taken to report. Later, we demonstrate REFORM's significance by deploying the framework with RPTSC's reward scheme and TERM. Specifically, we prove that REFORM considerably improves fairness; while incentivising truthful and early reports. Furthermore, we conduct synthetic simulations to validate our results.

Research Papers from the Thesis Work

Conference Papers

1. Samhita Kanaparthy, Sankarshan Damle, and Sujit Gujar. "REFORM: Reputation Based Fair and Temporal Reward Framework for Crowdsourcing." In Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems **(AAMAS 2022)**

Other Publications

1. Samhita Kanaparthy, Manisha Padala, Sankarshan Damle, and Sujit Gujar. "Fair Federated Learning for Heterogeneous Data." In 5th Joint International Conference on Data Science & Management of Data **(CODS-COMAD 2022)**

2. Sambhav Solanki, Samhita Kanaparthy, Sankarshan Damle, Sujit Gujar. "Differentially Private Federated Combinatorial Bandits with Constraints." In the proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases **(ECML-PKDD 2022)**

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Good ideas can come from anywhere, making openness is an imperative in the times of crisis."

– Prof. Henry Chesbrough, Open Innovation

As good ideas can come from anywhere, many useful systems often rely on crowd wisdom, popularly known as crowdsourcing.

1.1 Motivation

1.1.1 Power of the Crowd

The recent COVID-19 pandemic has caused unprecedented health and economic crises. In response to the emergency, we witnessed many initiatives that prompted interactions among different sectors, including health care, industry, government, educational institutions and individuals, to devise innovative ideas and collaborative infrastructures that support the crisis. Many opportunities encouraged the crowd's involvement to provide feasible solutions for various challenges like contact tracing, emergency planning, public health monitoring, crisis management, and food delivery to the needy [2]. This pandemic practically proved that adopting a *collaborative approach* and bringing together a diverse population to solve a common goal is more *efficient*. Recently, Kevin Boudreau and Karim Lakhani said, *"to answer the most vexing innovation and research questions, the crowd is* *becoming the partner of choice*". The impact of COVID-19 has exacerbated an inclination towards a *decentralised world*, where central powers are transferred to the crowd. Mahatma Gandhi has succinctly put forth the essence of decentralisation, "*A true democracy cannot be worked out by twenty men sitting at the centre. It has to be worked from below by the people*". The most recent rise of decentralised and distributed social systems like distributed ledgers, specifically Bitcoin [3, 4], and decentralised learning techniques like Federated Learning [5, 6] has genuinely revolutionised fundamental social and economic practices leveraging the ***power of the crowd***. The power of the crowd describes the crowd's ability to exert influence. Similarly, *crowdsourcing* systems demonstrated their impact on innovations by utilising various individuals' unique and diverse skills for achieving a common goal. In this work, we mainly discuss crowdsourcing systems. We describe crowdsourcing systems and their applications and primarily focus on the *incentive* mechanisms used to implement them.

1.2 Crowdsourcing Systems

Crowdsourcing is an effective method to solve a large variety of tasks by employing the combined efforts of a distributed human population. Jeff Howe first coined crowdsourcing in 2006 to represent organisations *outsourcing* their tasks to a large group of people [7]. Individuals and organisations often face the issue of a lack of resources and expertise for executing tasks. Outsourcing tasks to experts at a cost enables them to perform them without procuring extra resources. With the advent of networking, outsourcing has become even more convenient. Online social platforms have given access to a vast crowd with plenty of diverse expertise, making it easy to find a group of people to solve a problem collectively or generate web content by *aggregating* collective knowledge. Crowdsourcing has been part of our ecosystem since the 20th century. Still, it also finds its roots in the past. Initially, people used the term crowdsourcing to refer to online outsourcing. However, later, the concept found a broader definition.

> *"Understanding diversity is imperative to understanding collective intelligence, and collective intelligence is an essential ingredient in one of the primary categories of crowdsourcing: the attempt to harness many people's*

knowledge in order to solve problems or predict future outcomes or help direct corporate strategy."

– Jeff Howe, Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business

Crowdsourcing allows *requesters* of the systems to get work done, obtain information, or collect opinions from a large group of people who input their data through the internet, sensors, and other data streams. It enables the crowd with plenty of diverse expertise to contribute to any outsourced task. Such tasks could be rating products online, testing applications for a company, or collecting real-world data [8, 9, 10]. Crowdsourcing helps in achieving greater diversity and quicker solutions to the tasks. However, can we rely on the crowd that participates in these systems? To build a reliable crowdsourcing system, one must offer appropriate incentives to the crowd to maintain the quality of the contributions. In the coming sections, we discuss the extensive use of crowdsourcing applications followed by incentive mechanisms.

1.2.1 Applications of Crowdsourcing in Everyday Life

Some well-known categories of crowdsourcing that are being used effectively in the commercial world include crowd-sensing, crowdfunding, and crowd voting. *Crowdsensing*, also called mobile crowdsensing, is a technique where a large group of individuals with mobile devices capable of sensing or computing collectively share data and extract information to achieve any tasks of common interest [11]. In short, crowdsourcing sensor data from mobile devices is crowdsensing. *Crowdfunding* is the practice of funding a project or venture by raising money from many people, typically via the Internet [12]. *Crowd voting* is an open, dynamic process that replaces expert rating to gather a large group's opinions and judgements on a specific topic [13]. We now explain a few famous crowdsourcing applications we see daily, including crowdsensing, crowdfunding and crowd voting.

Wikipedia

Wikipedia, a classic example of crowdsourcing, often said to be 'the father of internet crowdsourcing'. Wikipedia is the largest and most-read free reference content available in

Figure 1.1: Image courtesy: `https://www.wikipedia.org/`

more than 300 languages. It was officially launched on January 15 2001, by Jimmy Wales and Larry Sanger. Content in Wikipedia is contributed by individual contributors aiming to be *"The sum of all human knowledge in one place"*. As of March 2022, Wikipedia has around 55 million articles, with more than 6.4 million in English [14].

Air Quality Egg (AQE)

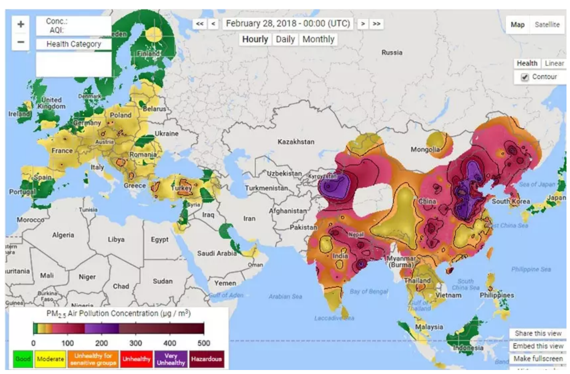

Figure 1.2: Real-time World Air Quality Index Visual Map

Air pollution is a global problem. World Health Organization (WHO) reports that 92% of the worldwide population lives in places where air quality levels exceed WHO limits. Air pollution is causing at least 4.5 million premature deaths each year worldwide.

Air Quality Egg (AQE) is an IoT device to monitor airborne pollutants through mobile crowdsensing. AQE facilitates the crowd to be aware of real-time and accurate air quality data. The AQE project team has covered over 70 countries with 9000 stations in 600 major cities [15]. With the data collected, the non-profit organisation Berkeley Earth developed a real-time visual map of air pollution (Fig 1.2).

Amazon Mechanical Turk (AMT)

Amazon Mechanical Turk (AMT) is a crowdsourcing service operated under Amazon Web Services (AWS). The platform allows businesses or individuals to hire remotely located crowd workers to perform discrete tasks that computers cannot do. Requesters post Human Intelligence Tasks (HITs) on the platform, including identifying objects in an image or video, writing product reviews, or answering queries. 'Turkers' or crowd workers on the platform browse existing tasks and complete them in exchange for a rate set by the requester. As of April 2019, requesters on AMT could register from 49 approved countries [16].

DARPA Red Balloon Challenge

The defence research organisation of the United States launched a challenge to explore the roles the internet and social network play in communications to solve a broader scope of time-critical problems. DARPA Red Balloon Challenge was to locate ten red weather balloons in 10 different undisclosed regions of the United States [1]. DARPA challenge's purpose was to elicit authentic information from the potentially fraudulent information. MIT team won the challenge by locating the balloons within 9 hours. The team used a recursive incentive scheme similar to multi-level marketing to recruit and reward the participants for helping them find the balloons.

Figure 1.3: 10 Red balloons spotted in 9 hours

We next give the intuition behind the incentive scheme used by MIT [1]. Suppose Rahul gets an invitation link `xyz.rahul` from MIT and joins the team. Then Rahul sends his link to Banu, who uses it to join the team and receives a unique link `xyz.banu`. She shares it on her social media, which excites her friend Kavya to sign up. Kavya encourages her brother Karthik to participate. Enthusiastic Karthik spots the balloon and reports to the MIT team. Winning the challenge team rewards Karthik with ₹2000, and Kavya gets ₹1000 for inviting Karthik. Banu gets ₹500 for inviting Kavya and so on, as shown in Fig 1.4. This incentive scheme not only incentivises the participants to locate the balloons but also to invite more people to the challenge to join the team.

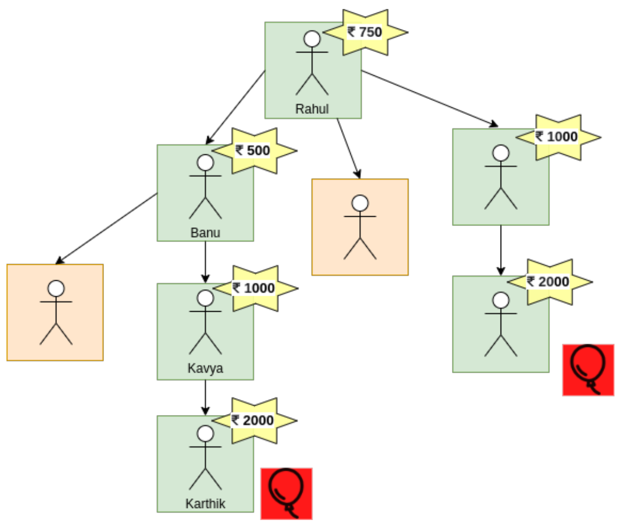

Figure 1.4: Illustration: Incentive Scheme used by Team MIT for DARPA Red Balloon Challenge [1]

Waze Navigation Application

Waze is a free, real-time navigation app operated by over 100 million users. Waze paves the way for dealing with information about city traffic and transportation worldwide. It depends on user data to monitor and forward traffic information for its maps for more than 180 countries.

Waze collects data from the crowd in three different ways: 1) users actively report on live events happening on the road, as seen in Figure 1.5; 2) users' data about driving speed and traffic conditions is passively gathered when the app is active, and 3) volunteers that constantly edit the maps utilised in the app [17].

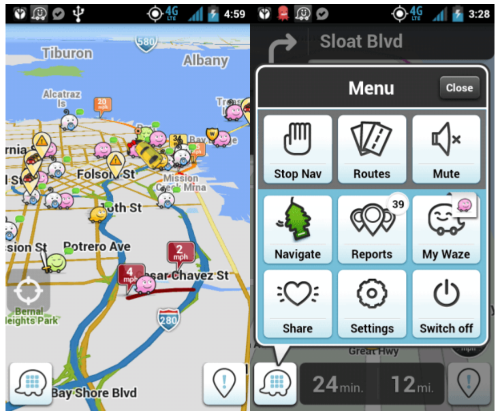

Figure 1.5: Waze Application: Crowdsourcing Maps and Traffic Information

Lego

Lego! Inspired by the building blocks game we all played in our childhood. Lego ideas and building blocks is an online community with around 1 million members that utilises crowdsourcing to select new product ideas and support them by providing an outlet to share their ideas. The platform accepts submissions from any user and conducts crowd voting by other community members to vote for them. The idea with majority votes is forwarded to production. While the designer gets an incentive of 1% royalty, the company enjoys increased customer engagement [18].

Mondelez India's Madbury Campaign

Mondelez India, the makers and bakers of famous food brand Cadbury, have initiated a campaign inviting the nation to 'go Madbury for Cadbury'. This consumer-led, digital pan-India campaign encouraged Indian consumers to experiment with their unique flavours and develop creative suggestions for the next Cadbury Dairy Milk edition. Anil Viswanathan, Marketing Director, Mondelez India, said, *"This initiative is meant to empower consumers by providing more choices, and newer eat experiences"* [19].

Figure 1.6: Madbury Campaign

Even though the concept appears appealing and innovative, there are numerous challenges in designing crowdsourcing mechanisms to ensure the reliability of the contributions. Typically, the crowd in these systems participate to maximise their gains. For instance, in Wikipedia, an individual can fool the system by editing the facts. As there are no penalties for contributions, it can attract extremists. Hence, no fact on Wikipedia is ever final [20]. As a user-generated source, any information it contains at a particular time could be vandalism, a work in progress, or simply incorrect. Information aggregation from multiple reports provided by potentially unreliable or strategic individuals – referred hereafter as *agents* is a primary challenge in crowdsourcing systems. In the case of Wikipedia, its core mission is to create encyclopedic content. Thus, any appropriate restriction against an editor can impede its purpose [21]. However, in general, the crowdsourcing systems must maintain the quality of the reports submitted. Additionally, if the information gathered through crowdsourcing is used for enterprises (e.g. AMT HITs), the agents need their fair share of the reward. Many crowdsourcing systems focus on *incentivising* the agents for reliable contributions. More specifically, Team MIT's incentive scheme in DARPA Red balloon challenge emphasises that having a suitable, well-planned incentive scheme enhances the competence of a crowdsourcing system. In the next section, we provide a high-level overview of the existing incentive mechanisms for crowdsourcing.

1.2.2 Incentive Mechanisms for Crowdsourcing

In general, for most of the crowdsourcing tasks, the requesters do not have answers, i.e., they do not know the *ground truth*. Without the proper information about the ground truth, the requesters of the system cannot verify the correctness/reliability of the agents' reports. Due to this fact, strategic agents may have incentives to manipulate the system.

In general, a straightforward strategy for an agent could be to answer randomly, expecting some reward without actually performing the assigned tasks. To overcome such manipulations, researchers have devoted a large body of work in this field to problems comprising (P1) *selection of agents* for particular tasks [22, 23, 24, 25]; and (P2) *incentivising* them for truthful data elicitation. E.g., providing right incentives for the agents to report AQE data truthfully. In this thesis, we focus on P2. Several incentive mechanisms with specific reward schemes exist in literature [26, 27, 28, 29, 30].

To design such reward schemes, researchers consider the two approaches in crowdsourcing. Both the approaches aim to motivate strategic agents to exert efforts and report their answers truthfully. The first approach assumes access to *gold-standard tasks*, i.e., the tasks for which ground truth is available. In this scenario, researchers use these tasks to design a proper scoring rule to promote an incentive mechanism [31, 32]. In the second approach, the requester can access no such gold-standard tasks. For this, researchers rely on mechanisms with reward schemes that consider the accuracy of the answers provided by the agents to overcome random reporting. *Peer Based Mechanisms* (PBMs) achieve this by rewarding agents based on their consistency with the reports submitted by other agents, referred to as *peers*. For instance, *Robust Peer Truth Serum for Crowdsourcing* (RPTSC) [29] evaluates an agent's report against a randomly paired peer's report and rewards it if they match. With this, RPTSC incentivises agents to exert efforts and report truthfully. However, do these mechanisms provide fair rewards to the agents?

1.2.3 Fairness

Despite their advantages, PBMs are inherently unfair as the reward for an agent in them depends on the consistency of its report with others, i.e., not solely on the agent's effort. While these mechanisms incentivise efforts and truthful reporting, if still, some agents report randomly or are malicious, it may discourage reliable agents' participation. Thus, ensuring a fair evaluation is crucial to motivate agents to participate in a crowdsourcing platform. As a result, researchers are actively looking to achieve fairness in crowdsourcing through mechanism design [31, 33, 34, 35]. Goel and Faltings. [31] propose a fair mechanism assuming few gold-standard tasks. Moti et al. [33] propose a mechanism only for localised settings. However, these mechanisms address fairness in PBMs by assuming access to ground truth or working with a constrained setting. We aim to design a fair and truthful incentive mechanism for crowdsourcing systems without such assumptions. We explain our precise problem statement in the next section.

1.3 Problem Addressed

On average, more than 1,200 tornadoes in the United States occur annually, resulting in over 80 deaths and 1,500 injuries. Disasters like this, which happen without warning, strain the emergency response teams [36]. Here, we observe that the data about the occurrence of a disaster is dynamic. The response team requires *real-time* awareness about the situation for faster access and proper mitigation of damage during such disasters. Emergency response teams can utilise crowdsourcing platforms to collect real-time data from the crowd through sensors, GPS, and other data streams. The data reported early is valuable during a crisis for better damage mitigation. Unless the agents are reliable, it is not beneficial for the agencies to rely upon the crowd's data.

Motivated by this, in this work, we consider a crowdsourcing setting in which the task's requester desires real-time data. That is, the requester requires the agents to submit their reports at the earliest. We refer to such a setting as *temporal setting*. Such tasks are time-sensitive and may comprise real-time data collection (for e.g., passenger train timetable [37], emergent safety incidents information [38], real-time COVID-19 data [39]). In all such tasks, the data reported early is *valuable*.

In such a setting, it is natural to assume that the reward should reduce with time for any mechanism to incentivise early reporting. However, such reward decay may encourage the late agents to report randomly rather than exert efforts, further aggravating the fairness challenges. Thus, a right level of decay in incentives is required. In summary, we address the following challenge.

> **Challenge.** *To design a PBM that ensures* fairness *and* truthful *reporting in temporal setting*

Towards this, we propose two notions of fairness and key interesting ideas to achieve them while ensuring truthful reporting. We list our contributions in the next section.

1.4 Contributions

- To improve fairness in PBMs, we propose the idea of allowing reliable (or trustworthy) agents with additional pairing chance(s) to evaluate their reports for obtaining a reward. Intuitively, such additional chances will nullify the penalty a trustworthy agent would have incurred from the unfair pairings.

- To decide which agent will receive additional chances, we deploy reputation model to quantify trustworthiness of agents in the crowdsourcing system. Towards this, we design an iterative framework ***REFORM:*** *REputation based Fair and tempOral Reward fraMework for crowdsourcing*. We build REFORMas an abstract framework that employs the reward scheme of any existing PBM and a suitable reputation model.

- To compare fairness among different PBMs, we introduce new notions of fairness: (i) *γ-fairness*, which captures the proximity of the expected reward a PBM guarantees with its optimal reward. The greater is the value of γ, the fairer is the PBM. (ii) *Qualitative Fairness*, which ensures that the reward an agent obtains is proportional to its reputation, i.e., the trust the system places on its report.

- To quantify the reliability – the trustworthiness of agents in temporal setting, we propose a novel *Temporal Reputation Model (***TERM***)*, which deploys *Gompertz* function [10] to output reputation scores. TERM assigns scores to the agents based on their reports and the time taken to submit them. We prove that TERM scores are high for early and truthful reports. We also show that TERM is resistant to single report strategy, where all the agents collude to report the same.

- Having TERM as reputation model and RPTSC as PBM whose reward scheme is adopted, we prove that REFORM framework with RPTSC mechanism and TERM is *strict Nash incentive-compatible*, i.e., exerting efforts and reporting truthfully and early is a strict Nash equilibrium.

- With γ-fairness and qualitative fairness, we prove that REFORM with RPTSC is significantly fairer than RPTSC.

- We validate our claims empirically on well-thought simulations.

1.5 Outline of the Work

In this section, we provide a brief outline for each chapter of this thesis.

- In Chapter 2, first, we give background for Mechanism Design and introduce the formulation for a game in Crowdsourcing setting. We formally provide different fundamental game-theoretic properties. Secondly, we broadly classify different incentive mechanisms proposed for different crowdsourcing settings and discuss them in detail. Next, we provide important literature related to existing fair reward mechanisms. Lately, we have discussed how reputation models can be incorporated into crowdsourcing mechanisms to improve the effectiveness of the systems.

- In Chapter 3, we give our approach of improving fairness in PBMs. We show that our approach is better than other naive approach. Based on our approach, we propose an iterative framework REFORM which adopts reward scheme of any existing PBM and a suitable reputation mode. We then introduce two new notions of fairness for PBMs, namely, γ-fairness and Qualitative fairness.

- Chapter 4 introduces a temporal reputation model, TERM. TERM provides reputation scores to agents considering their reports along with the reporting time. We present the detailed procedure to compute TERM scores and discuss their properties.

- In Chapter 5, we demonstrate the significance of the REFORM framework keeping RPTSC as the base PBM and TERM as the reputation model. We theoretically prove that REFORM with RPTSC and TERM is Nash Incentive Compatible. We also show that with framework REFORM, the fairness is significantly improved. Specifically, we demonstrate that REFORM with RPTSC is fairer than RPTSC and is qualitatively fair. Finally, we validate our results through synthetic simulations.

Chapter 2

Foundations of Strategic Crowdsourcing

This chapter focuses on the engineering side of game theory. It speaks about how crowdsourcing systems can be set up as a game allowing agents to explore their freedom of choices in its restrictive environment. The analysis here is about understanding the consequences of freedom in this environment. It also reviews most relevant literature for the same.

This chapter introduces the existing literature on incentive mechanisms and provides the necessary preliminaries for the coming chapters. Section 2.1 presents our crowdsourcing setting formulated as a game. We also give a brief overview of incentive properties that are used in mechanism design. Section 2.2 provides a broad classification of incentive mechanisms, followed by Section 2.3, discussing peer based mechanisms, which is the main focus of our thesis. Since we analyse our framework based on the RPTSC mechanism, we detailly provide its setting and properties in Section 2.4. In Section 2.5, we speak about a few works on fairness in PBMs. Lastly, Section 2.6 briefly reviews existing reputation models in crowdsourcing systems.

2.1 Mechanism Design

To maintain the accuracy of crowdsourcing systems, it is essential for the requesters of the system to acquire an appropriate set of agents to work with and also motivate them to exert *efforts* in obtaining *good quality* reports. Researchers have used different methods to improve the quality of reports by (i) *Filtering the data*: eliminating outliers and statistically inconsistent data [40, 41]. (ii) *Gold-standard tasks*: using the gold-standard tasks, the requester can assign reputation scores to the workers. With the reputation scores of the workers, their reported data is relatively valued [24, 42]. (iii) *Providing game-theoretically designed incentives*: requesters incentivise agents to exert efforts and provide accurate information to the system using game theory [27, 29].

Data is often not easily verifiable, or ground truth may not be available. Thus, methods (i) and (ii) cannot be applied in such settings. Towards this, enforcing game-theoretic properties to reward the agents often comes as a rescue in most of the practical settings. This chapter focuses on the discussion of game theoretically sound incentive mechanisms.

Agents in these systems are typically *strategic*; they can fool the system by misreporting the data. An obvious strategy for agents to maximise their profit is to provide random reports. To overcome random reporting, the challenge is to design novel incentive schemes that only reward agents who exert efforts instead of paying fixed rewards per report or unit time. *Mechanism Design* solves this problem elegantly by setting up a *game* among agents to achieve desired objectives. In *Game Theory*, it is usually assumed that agents in the system are *rational* and *intelligent*; every agent chooses its strategy to maximise its reward. Thus, providing agents with a reward that covers their cost of effort is necessary. We now present the game formulation, related notations and agent strategies required for our analytical discussion. For more details about mechanism design, the interested readers are referred to the tutorials [43, 44].

2.1.1 Crowdsourcing Game: Formulation and Agent Strategies

The following is a general crowdsourcing setting for formulating, analysing, and solving its game-theoretic properties.

- The requester of the system publishes a set of tasks $\mathcal{T}$ on the crowdsourcing platform. Each task $\tau \in \mathcal{T}$ has a discrete and finite answer space $\mathcal{X}$.
- There are a set of agents $\mathcal{N} = \{1, 2, \ldots, n\}$ that are rational and intelligent to solve these tasks.
- Each agent i can choose a strategy s_i such that $s_i \in S$, where S is the strategy space, common for all the agents.
- According to the chosen strategy s_i, agent i exerts its efforts e_i and observes a evaluation x_i.

- We assume every agent i has a *prior belief* $P_i(x)$ regarding its evaluation (or value) x. After observing the evaluation x_i, agent i obtains a *posterior belief* $P_i(x|x_i)$.
- Let $s = \{s_1, s_2, \ldots, s_n\}$ denote the strategy profile containing the strategies made by n agents. We use s_{-i} to denote the strategy profile without an agent i.
- The utility gained by agent i when all the other agents adopt the strategy $s = \{s_i, s_{-i}\}$ is $u_i(s_i, s_{-i}) = R(\cdot) - c(\cdot)$. Here $c(\cdot)$ is the cost required for exerting efforts.

For our analysis, we characterise agents' strategies into three types as follows:

Definition 2.1 (***Trustworthy Strategy***)**.** *A reporting strategy is called* trustworthy *if the agent invests efforts to solve the task and truthfully reports the observation.*

Definition 2.2 (***Deceiving Strategy***)**.** *A reporting strategy is called* deceiving *if the agent invests efforts to solve the task but may not report the true observation.*

Definition 2.3 (***Random Strategy***)**.** *A reporting strategy is called* random *if the agent randomly reports the value irrespective of the task.*

Mechanism Design utilises the concept of *incentive compatibility*. Incentive compatibility refers to offering the proper incentives that make agents report truthfully. Researchers design incentive compatible mechanisms such that agents' best response is to reveal the truth. There are broadly two types of incentive compatibility.

1. ***Dominant Strategy Incentive Compatibility (DSIC)***: Reporting the truth is the best response for each agent, irrespective of what other agents report.
2. ***Nash Incentive Compatibility (NIC)***: Reporting the truth is the best response for each agent, given that the rest of the agents report the truth. Depending on the game, it can be *Ex-Post Nash Incentive Compatible (EPIC)* or *Bayesian Nash Incentive Compatible (BIC)*.

In the following subsection, we discuss these incentive properties in detail with a few examples.

2.1.2 Game Theoretic Definitions: Incentive Properties

Before understanding the key incentive properties, let us first look into the standard representation of games called *strategic form games*. Strategic Form Game is a complete information game where every agent has the entire game as common knowledge.

2.1.2.1 Strategic Form Game

A strategic form game is defined as $\Gamma = \langle \mathcal{N}, (S_i)_{i \in \mathcal{N}}, (u_i)_{i \in \mathcal{N}} \rangle$, where

- $\mathcal{N} = \{1, 2, \ldots, n\}$ is the set of agents
- S_i is the set of pure strategies of agent i where $i = 1, 2, \ldots$
- $u_i : \Theta_1 \times \ldots \times \Theta_n \times S_1 \times \ldots \times S_n \to \mathbb{R}$ is the reward function which assigns the reward a agent i would get.

Having defined the strategic form game, we now formally give definitions of different equilibriums for a complete information game.

Definition 2.4 (***Dominant Strategy Equilibrium (DSE)***)**.** *A strategy profile* $s^* = \{s_1^*, s_2^*, \ldots, s_n^*\}$ *is said to be DSE if for any agent* $i \in \mathcal{N}$

$$u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}^*),\ \forall s_i \in S,\ \forall s_{-i},\ \forall i \in \mathcal{N}$$

Definition 2.5 (***Pure Strategy Nash Equilibrium (PSNE)***)**.** *A strategy profile* $s^* = \{s_1^*, s_2^*, \ldots, s_n^*\}$ *is said to be PSNE if for any agent* $i \in \mathcal{N}$

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*), \forall s_i \in S, \forall i \in \mathcal{N}$$

So far, we have seen game representation and incentive properties for a complete information game. We will now look at games with incomplete information. Here, at least one agent has *private information* about the game, which other agents may not know. We represent incomplete information games using a form called *Bayesian Games*. We formally give the representation of Bayesian games and discuss their incentive notion, BIC.

2.1.2.2 Bayesian Games

Bayesian game is defined as $\Gamma = \langle N, (\Theta_i), (S_i), (b_i), (u_i) \rangle$, where

- $N = \{1, 2, \ldots, n\}$ is the set of agents
- Θ_i is the set of types of agent i where $i = 1, 2, \ldots$
- S_i is the set of pure strategies of agent i where $i = 1, 2, \ldots$
- b_i is the belief function that maps from Θ_i to $\Delta(\Theta_i)$; i.e., the probability distribution representing the agent's beliefs about the types of other agents
- $u_i : \Theta_1 \times \ldots \times \Theta_n \times S_1 \times \ldots \times S_n \rightarrow \mathbb{R}$ is the reward function which assigns the reward a agent i would get.

Having defined the bayesian form game, we now formally give definitions of DSIC, EPIC, BIC. For this, let s_i^{TS} denote the strategy where agent i reports the truth. With this, we present the following definitions.

Definition 2.6 (***Bayesian Nash Equilibrium (BNE)***)**.** *A strategy profile* $(s_1^*, \ldots, s_n^*)$ *is Bayesian Nash Equilibrium if for any agent* $i \in N$*,*

$$u_i((s_i^*, s_{-i}^*)|\theta_i) \geq u_i((s_i, s_{-i}^*)|\theta_i),\ \forall s_i : \Theta_i \rightarrow S_i,\ \forall \theta_i \in \Theta_i$$

Definition 2.7 (***Dominant Strategy Incentive Compatible (DSIC)***)**.** *A mechanism is said to be DSIC if for every agent reporting the truth is dominant strategy, i.e.,*

$$u_i(s_i^{TS}, s_{-i}) > u_i(s_i, s_{-i}), \forall s_i \in S, \forall s_{-i}, \forall i \in \mathcal{N}$$

Example 2.1. *Consider a game with two agents* A, B *who have to report their valuations for a product in an auction. The game is designed such that if the true valuations of two agents are* θ_1 *and* θ_2*, and their reported valuations are* $\overline{\theta}_1$ *and* $\overline{\theta}_2$*, utility for agent* A *is as follows:*

Case 1: $\theta_1 > \overline{\theta_2}$

- if $\overline{\theta_1} > \overline{\theta_2}$: agent A gets $\theta_1 - \overline{\theta_2}$.

- if $\overline{\theta_1} \leq \overline{\theta_2}$: agent A gets 0.

We observe that reporting the true valuation (i.e., $\overline{\theta}_i = \theta_i$) results in maximum reward in the above case.

Case 2: $\theta_1 \leq \overline{\theta_2}$

- if $\overline{\theta_1} > \overline{\theta_2}$: agent A gets $\theta_1 - \overline{\theta_2}$ which is negative.
- if $\overline{\theta_1} \leq \overline{\theta_2}$: agent A gets 0.

Observe that the maximum reward agent A can get in this case is 0. Further, that is after reporting true valuation θ_1. Thus, from both the cases, one can conclude that reporting the truth is the optimal strategy. This game is Dominant Strategy Incentive Compatible. □

Definition 2.8 (***Ex-Post Nash Incentive Compatible (EPIC)***)**.** *In a strategic form game, a mechanism is said to be EPIC if every agent reporting the truth maximises its utility given all the other agents report the truth. i.e.,*

$$u_i(s_i^{TS}, s_{-i}^{TS}) \geq u_i(s_i, s_{-i}^{TS}), \forall s_i \in S, \forall i \in \mathcal{N}$$

In other words, a mechanism is EPIC if it has s^{TS} as its PSNE, where $s^{TS} = \{s_1^{TS}, s_2^{TS}, \ldots, s_n^{TS}\}$.

Example 2.2. *Consider a game of two hunters A, B who have to decide to hunt between Stag and Hare. The game is designed such that (i) Two hunters are required for hunting a Stag and the food is sufficient for 4 days for each hunter; (ii) Hare requires a single hunter and food is sufficient for 2 days; (iii) if both the hunters hunt the Hare, they share their food. The utilities for both the hunters in each case are as follows:*

	Stag	Hare
Stag	(4,4)	(0,2)
Hare	(2,0)	(1,1)

- Say hunter A hunts a Stag. The best response for the hunter B is to hunt Stag.
- If hunter A hunts Hare. Then the best response for the hunter B is to hunt Hare too.

- Hence, (Stag, Stag) and (Hare, Hare) are Pure Strategy Nash Equilibrium. □

Definition 2.9 (***Bayesian Nash Incentive Compatible (BIC)***)**.** *A mechanism is said to be BIC if for each agent reporting the truth maximises its utility in expectation of the types of the rest of the agents. i.e.,*

$$u_i((s_i^{TS}, s_{-i}^{TS})|\theta_i) \geq u_i((s_i, s_{-i}^{TS})|\theta_i) \forall s_i \in S, \forall i \in \mathcal{N}$$

In other words, a mechanism is BIC if it has s^{TS} as its BNE, where $s^{TS} = \{s_1^{TS}, s_2^{TS}, \ldots, s_n^{TS}\}$.

The above notions of truthful information elicitation have been used in many real applications, including crowdsourcing. There are two types of mechanisms, *auction-based* mechanisms [45, 46, 47] or designing incentives for the reports based on the quality of the reports, e.g., [27, 28, 48]. The auction-based mechanisms are elementarily targeted to get the true cost of the workers to collect information. Such mechanisms are useful in tasks requiring significant work, e.g., building a small project. Such crowdsourcing is referred to as *expert-sourcing*. Our primary focus is on collecting truthful reports, where the cost of collecting the information is already public information, not requiring a mechanism design for eliciting the cost.

We have seen different incentive notions mechanisms employ to encourage agents to exert efforts and report truthfully. In the next section, we discuss the prominent crowdsourcing mechanisms existing in the literature.

2.2 Mechanisms for Truthful Information Elicitation

Incentive mechanisms in a crowdsourcing game are designed so that the highest expected reward requires agents to exert efforts to solve the tasks and report honestly. In contrast, random reports will, on average, produce no reward.

Figure 2.1 depicts the broad classification of crowdsourcing mechanisms introduced in the literature [26] and then give a detailed discussion on each of them. We distinguish mechanisms for crowdsourcing settings with verifiable and unverifiable information. In a crowdsourcing setting with verifiable information, where the ground truth will be available, mechanisms adopt reward schemes that provide incentives to agents based on ground truth.

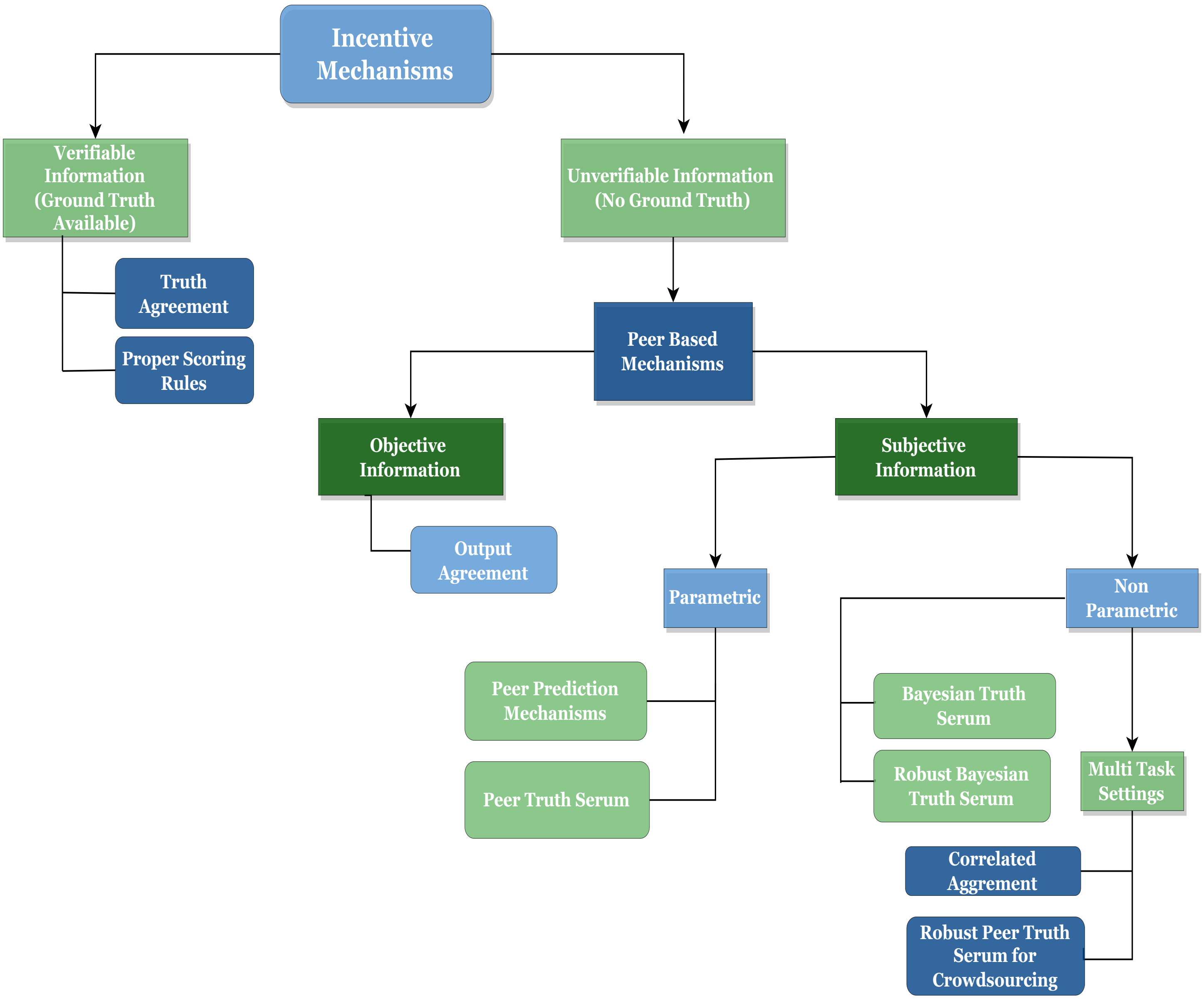

Figure 2.1: Classification of Incentive Mechanisms

We discuss them in Section 2.2.1. In an unverifiable setting where ground truth is never known, researchers have introduced peer-based mechanisms that are discussed in Section 2.3. However, we require assumptions about agent beliefs to achieve incentive properties in unverifiable settings. Some mechanisms take these beliefs as parameters (Section 2.3.2). These parameters are to be set appropriately. Often, setting correct parameters is challenging. Towards this, non-parametric mechanisms obtain the parameters from the data provided by agents (Section 2.3.3). Some of the non-parametric mechanisms require agents to provide additional information during reporting. Others work in a multi-task setting where information about parameters is extracted from statistics of the data provided by a group of agents for different similar tasks.

2.2.1 Mechanisms with Ground Truth

In a setting where reported data is verifiable, the simplest way is to design the rewards based on the accuracy of the data. For instance, consider a crowdsourcing system requesting weather forecasts or outcomes of cricket matches. In such cases, it is practicable to verify the accuracy of agents' reports and reward accordingly once the ground truth becomes available. Two types of incentive mechanisms are used in such verifiable settings (i) Eliciting the value and (ii) Eliciting the probability distribution of values.

Truth Agreement (TA)

Truth Agreement (TA) [49] is the mechanism used for settings that require eliciting verifiable values. The requester of this setting needs agents to submit a discrete value as a report (e.g., the winning team in IPL 2022) and rewards them if their reports match the observed ground truth. Algorithm 1 gives the incentive scheme used in truth agreement. Here, α is a scaling factor compensating the cost of efforts and the indicator variable $\mathbb{I}_{x=g} = 1$ if $x = g$ and 0 otherwise.

Theorem 2.1. *[49] Provided the scaling factor α is large enough, the scaled Truth Agreement Mechanism induces dominant strategies that are trustworthy, i.e., TA is DSIC. With proper offset, random strategies carry no expected reward.*

Algorithm 1 **Truth Agreement** (TA)

1: Agent i reports data y

2: Requester of the system observes the ground truth g

3: Agents receives a reward: $Reward(y,g) = \alpha\,[\mathbb{I}_{y=g} - P_i(x_0)]$, where $x_0 = \text{argmax}\, P_i(x)$, where $P(\cdot)$ is prior.

Proper Scoring Rules

There are settings where the requester would need agents to report their posterior distribution $P(\cdot|x)$ instead of the exact observed value x. For instance, in the IPL example, let us say the requester needs agents to report not the winning team but their confidence by providing the complete probability distribution. Researchers use *proper scoring rules* [50] (Algorithm 2) in such settings to elicit the probability distributions.

Some well-known examples of proper scoring rules are:

- Quadratic scoring rule: $SR(P, g) = 2.P(g) - \sum_{x\in\mathcal{X}} P(x)^2$
- Logarithmic scoring rule: $SR(P, g) = C + \ln P(g)$

Algorithm 2 **Proper Scoring Rule**

1: Agent i reports data (posterior distribution) P

2: Requester of the system observes the ground truth g

3: Agents receives a reward: $R(P, g) = SR(P, g)$, where SR is a proper scoring rule.

Theorem 2.2. *[50] For both the quadratic and logarithmic scoring rules, with proper scaling the scoring rule mechanism induces dominant reporting strategies that are trustworthy, i.e., proper scoring rule mechanism is DSIC. With proper offset, the expected reward for random reporting is equal to zero.*

So far, we have seen the mechanisms whose reward schemes depend on the observed ground truth. Now we look at the mechanisms for the settings where we cannot observe ground truth.

2.3 Peer Based Mechanisms (PBMs)

In most cases, the ground truth is unavailable, so we cannot verify the data that agents report. In such scenarios, data validation is performed using the coherency with data submitted by peers that observed the same phenomenon (or task). We refer to this as *peer consistency*. Towards this, researchers introduce *Peer Based Mechanisms (PBMs)* [27, 28, 29] that reward agents based on peer consistency. That is, PBMs reward an agent if its report matches its peer's report. The following sections discuss different PBMs that are presented in the literature.

Most of these mechanisms depend on *agent beliefs* about the task evaluations and the reward it may receive for different possible strategies. The following assumptions about agent belief updates are required for further discussion.

Definition 2.10 (***Self Dominating Condition***)**.** *An agents' belief update satisfies self dominating condition if and only if the observed value has the highest probability among all possible values:*

$$P(x|x) > P(x'|x)\ \forall x' \neq x$$

Definition 2.11 (***Self Predicting Condition***)**.** *An agents' belief update satisfies self predicting condition if and only if the observed value has the highest relative increase in probability among all possible values:*

$$\frac{P(x|x)}{P(x)} > \frac{P(x'|x)}{P(x')}\ \forall x' \neq x$$

2.3.1 PBMs for Objective Information

Firstly, we discuss a simple peer based mechanism used to elicit objective information.

Output Agreement (OA)

Output Agreement (OA) [51] is used to elicit values that are objective. When the data submitted is objective, one can easily compare the agent's data with its peer to evaluate consistency. In this mechanism, two agents who solve the same task are evaluated against each other. OA provides a constant reward to both of them if and only if their answers match (Algorithm 3).

Theorem 2.3. *[51] For self dominating belief updates (Definition 2.10), the output agreement mechanism has a strict ex-post subjective nash equilibrium (EPIC) in trustworthy strategy.*

Algorithm 3 Output Agreement (OA)

1: Requester assigns tasks to all the agents $i \in \mathcal{N}$

2: Each agent i solves the task and reports its data y_i

3: Requester randomly selects a peer agent p who solved the same task and reported data y_p

4: Reward agent i receives is:

$$R(y_i, y_p) = \begin{cases} 1 & \text{if } y_i = y_p \\ 0 & \text{otherwise} \end{cases}$$

2.3.2 PBMs for Subjective Information

Sometimes, the data reported can be subjective. For example, reviews for dishes in a restaurant: every customer orders a different dish and has different opinions. There is no definite ground truth for individual reports, but there is ground truth for the distribution. We aim to obtain an accurate distribution to validate data in such scenarios. For subjective tasks, there are two types of peer based mechanisms depending on the prior and beliefs updates of agents:

- Mechanisms for homogeneous agent population with identical and known prior and posterior beliefs (belief updates).

- Mechanisms for common and known prior beliefs, but the belief updates can be heterogeneous as they satisfy *self predicting condition* (Definition 2.11).

Peer Prediction Mechanism

In a setting which assumes agents have common beliefs and belief updates, we use *Peer Prediction Mechanism* [52]. In this mechanism, the requester assumes a posterior distribution for every reported answer and calculates a reward using a proper scoring rule. Algorithm 4 formally presents peer prediction mechanism.

Theorem 2.4. *([52]) Peer Prediction Mechanism has a strict bayesian nash equilibrium (BNE) where all agents use trustworthy strategy, provided that all agents have the common beliefs and belief updates assumed in the mechanism.*

Algorithm 4 Peer Prediction Mechanism

1: Requester assigns tasks to all the agents $i \in \mathcal{N}$

2: Each agent i solves the task and reports its data y_i

3: Requester randomly selects a peer agent p with report y_p from the same task

4: Requester selects an assumed posterior distribution $\hat{P}_i$ associated with y_i

5: Agent i's reward is: $R(y_i, y_p) = SR(\hat{P}_i, y_p)$, where SR is a proper scoring rule.

2.3.2.1 Common Prior Mechanisms

In peer prediction mechanism, the requester needs to know the exact posterior distributions of every agent for each observation. Moreover, these distributions have to be the same for every agent, which is a strict condition. Towards this, *Common Prior Mechanisms* were introduced with a more reasonable assumption. Common Prior Mechanisms assume that all the agents have a common prior belief before they evaluate a task. For instance, in the IPL example, prior belief about the winning team could be reasonably guessed from the

IPL seasons that have happened so far. However, the posterior might differ a lot. Next, we discuss one common prior mechanism *Peer truth Serum (PTS)* [27].

Peer Truth Serum (PTS)

In a setting where common prior is available, the requester of the system uses peer truth serum to reward the agents. In this mechanism, requester and agents share a common prior distribution P. Similar to the mechanisms seen so far, peer truth serum rewards an agent if its report matches its peer's report. The formal mechanism is presented in Algorithm 5.

Theorem 2.5. *([27]) While satisfying self predicting condition, peer truth serum mechanism is a strict ex-post nash incentive compatible (EPIC) where all the agents choose trustworthy strategy.*

Algorithm 5 **Peer Truth Serum** (PTS)

1: Requester shares the common prior distribution P that is used in the mechanism
2: Each agent i solves the task and reports its data y_i
3: Requester randomly selects a peer agent p who solved the same task and reported data y_p
4: Reward agent i receives is: $R(y_i, y_p) = \frac{\mathbb{I}_{y_i=y_p}}{P(y_i)} - 1$

2.3.3 Non Parametric PBMs

Till now, the reward schemes of every mechanism we have discussed crucially depend on agent beliefs. This is not reasonable because (i) the requester may not have an estimate of these beliefs correctly, (ii) And the same mechanism is applied to a diverse crowd expecting their beliefs to be uniform. Towards this, we next discuss the PBMs that do not require the requester to know these beliefs. Thus, these PBMs either elicit beliefs from the agents using additional reports or by learning the probability distributions through the agent reported data.

Bayesian Truth Serum (BTS)

Bayesian Truth Serum (BTS) [28] mechanism does not assume any prior. Here, the requester of the system asks agents to provide two reports: an *information report* y_i and a *prediction report* F_i. An information report contains the answer provided by an agent, and a prediction report includes the prediction of an agent about other agents' reports. BTS rewards agents for both the reports submitted. The reward scheme used in BTS is as follows:

$$R(y_i, F_i) = r_{info}(y_i, \ldots) + r_{pred}(F_i, \ldots)$$

The reward for information report is defined as

$$r_{info}(y_i, \ldots) = \log \frac{freq(y_i)}{gm(y_i)}$$

Here, $\log gm(y_i) = \frac{1}{n} \sum_j \log f_j(y_i)$ is geometric mean of agents' predictions.

And the reward for the prediction report is given as

$$r_{pred}(F_i, \ldots) = -KL(freq(y_i)||F_i(y_i))$$

$freq(y) = \frac{num(y)}{n}$.

We next give the formal BTS mechanism in Algorithm 6.

Theorem 2.6. *([28]) Given a large enough population of agents, Bayesian Truth Serum has trustworthy strategy as strict Bayes Nash Equilibrium (BNE).*

As observed in BTS, the distributions submitted in the reported data can be far from the actual distributions. Moreover, this can significantly affect the incentives provided to the agents. To overcome this, researchers have introduced a *robust* version of BTS called *Robust Bayesian Truth Serum (RBTS)* [53]; we next discuss this.

Algorithm 6 **Bayesian Truth Serum** (BTS)

1: Requester assigns same task to a set of agents $\mathcal{A}$.

2: Each agent $i \in \mathcal{A}$ solves the task and submits its information report y_i and prediction report F_i.

3: Requester computes the histogram of information reports $freq(y_i)$ and geometric mean of prediction reports $gm(F_i)$.

4: Requester computes prediction score $r_{pred} = -D_{KL}(freq(y_i)||F_i(y))$ and information score $r_{info} = \ln freq(y_i) - \ln gm(y_i)$.

5: Reward agent i receives is: $R(y_i, F_i) = r_{info} + r_{pred}$.

Robust Bayesian Truth Serum (RBTS)

Robust Bayesian Truth Serum (RBTS)[53] mechanism works even with a smaller number of reports. It keeps the decomposable structure of the score into an information score and a prediction score, where the information score gives an incentive for truthfulness based on another agent's prediction report, and the prediction score uses a proper scoring rule against the information report.

Algorithm 7 **Robust Bayesian Truth Serum** (RBTS)

1: Requester assigns same task to a set of agents $\mathcal{A}$.

2: Each agent $i \in \mathcal{A}$ solves the task and submits its information report y_i and prediction report F_i.

3: Requester picks a random peer $p \in \mathcal{A}$ and computes a reward for agent i as follows:

$$R(y_i, F_i) = \frac{\mathbb{I}_{y_i = y_p}}{f_i(y_i)} + f_i(y_p) - \frac{1}{2}\sum_z f_i(z)^2$$

Both BTS and RBTS elicit beliefs from agents by requesting additional reports called prediction reports. Now we look at the PBMs that learn these beliefs from information reports submitted for *multiple tasks* that are similar within a short interval of time.

2.3.4 Multi Task Settings

First, we present Correlated Agreement [48] mechanism for multi-task settings.

Correlated Agreement (CA)

Correlated Agreement (CA) mechanism learns the beliefs from the data submitted by the agents. It assumes the following:

- Agents answer multiple tasks, and their strategies remain the same
- Agents and requester know and agree on the sign of correlation among each answer pair for different agents/same task.

Algorithm 8 presents the mechanism.

Algorithm 8 Correlated Agreement (CA)

1: Requester gives set of similar tasks $\mathcal{T}$ to a set of agents $\mathcal{A}$.
2: Each agent $i \in \mathcal{A}$ solves multiple tasks and submits its report y_i; and multiple agents can solve a single task.
3: Requester computes the matrix correlations Δ on the evaluation distributions $Pr(s)$ expected for the tasks such that $\Delta(x, y) = Pr(x, y) - Pr(x)Pr(y)$.
4: Requester derives a score matrix $S(x, y) = 1$ if $\Delta(x, y) > 0$ and 0 otherwise.
5: Requester randomly picks a peer p who submitted report y_p for the same task.
6: Let z_i and z_p be two reports submitted by agents i and p for other tasks.
7: Reward to agent i for its report y_i is: $R(y_i, y_p, z_i, z_p) = S(y_i, y_p) - S(z_i, z_p)$

Theorem 2.7. *([48]) Correlated Agreement mechanism is maximally strong truthful among all multi-task mechanisms that only use knowledge of the correlation structure of evaluations.*

Peer Truth Serum for Crowdsourcing (PTSC)

We have discussed Peer Truth Serum (Algorithm 5) where we use prior distribution P as a parameter. Knowing prior distribution is not possible in every setting. Towards this, Randanovic et al. [29] introduced *Peer Truth Serum for Crowdsourcing (PTSC)*. PTSC works in a multi-task setting. Here, one can learn the prior distribution from the data provided by a group of agents. In PTSC, distribution P is considered as the frequency of report $freq(\cdot)$ which is computed from the histogram of reports submitted for a set of multiple similar tasks. An agent's report is rewarded by evaluating against another report submitted for the same task. We formally give the PTSC mechanism in Algorithm 9.

Algorithm 9 Peer Truth Serum for Crowdsourcing (PTSC)

1: Requester gives set of statistically similar tasks $\mathcal{T}$ to a set of agents $\mathcal{A}$.

2: Each agent a solves its task and submits a report y_a.

3: To reward an agent i for submitting the report y_i to the task τ:

4: Requester computes the frequency of reported values within all the tasks. Let $freq(y_i) = \frac{num(y_i)}{\sum_y num(y)}$.

5: Requester randomly picks a peer p who submitted report y_p for the same task.

6: The reward is calculated as: $R(y_i, y_p) = \alpha \left(\frac{\mathbb{I}_{y_i = y_p}}{freq(y_i)} - 1 \right)$

Theorem 2.8. *([29]) While satisfying the self prediction condition and with a sufficient number of tasks, PTSC has trustworthy strategy as strict ex-post nash equilibrium and the reward of this equilibrium is greater than the that of all other equilibria.*

The PTSC mechanism assumes a large number of statistically similar tasks. Randanovic et al. have also proposed a *robust* version of PTSC called *Robust Peer Truth Serum for Crowdsourcing (RPTSC)* [29]. In our framework, we illustrate and prove game-theoretic properties with RPTSC as base PBM. Hence, we discuss RPTSC mechanism and its properties in detail in the next section.

Before that, we will briefly discuss more PBMs proposed in the literature.

2.3.5 Other Mechanisms

Peer Prediction with Heterogeneous Users (PPHU)

Peer Prediction with Heterogeneous Users (PPHU) [54] addresses a setting where users reports are subjective like differ in their taste, judgement and reliability. PPHU solves this problem by clustering agents based on their reporting behaviour. The mechanism works with clusters of agents and adopts algorithms that learn such a clustering.

In PPHU, the tasks are *ex ante* identical, i.e. signal of an agent for different tasks are sampled i.i.d. It is assumed that agent's strategy is uniform across different tasks. Let x_p be random variable for signal of agent p for a task. Use $D_{p,q}(i,j)$ to denote joint probability for a pair of signals (i,j) received by agents p,q respectively on a random task. And $D_p(i)$ and $D_q(j)$ for the marginal probabilities. The delta matrix between agents p and q is defined as:

$$D_{p,q}(i,j) - D_p(i)D_q(j)$$

Strategy of agent p is denoted by F^p, defines distribution for each possible signal i, $F^p_{i,r} = Pr(R_p = r|S_p = i)$. $\{F^p\}_{p\in P}$ is the strategy profile for agent p. A strategy is *informed* if there exist distinct $i,j \in [n]$ and $r \in [n]$ such that $F^p_{i,r} \neq F^p_{j,r}$. Otherwise it is *uninformed*.

For every pair of agents $p,q \in P$, we define *scoring matrix* $S_{p,q} : [n] \times [n] \rightarrow \mathbf{F}$ as a means of scoring agents reports. The set of tasks performed by each agent p are divided into nonempty sets of *bonus tasks* and *penalty tasks*, Denoted by M^p_1 and M^p_2 respectively.

To calculate payment to an agent p for a *bonus task* $t \in M^p_2$

1. Randomly select agent $q \in P\backslash\{p\}$ such that $t \in M^p_1$.
2. Pick penalty tasks $t^{'} \in M^p_2$ and $t^{''} \in M^q_2$ at random such that $t^{'} \neq t^{''}$.
3. Let the reports of agent p be r^t_p and $r^{t^{'}}_p$ and of agent q be r^t_q and $r^{t^{''}}_q$.
4. Payment of agent p for task t is then $S_{p,q}(r^t_p, r^t_q) - S_{p,q}(r^{t^{'}}_p, r^{t^{''}}_q)$.

5. The total payment to the agent is the sum of payments for the agent's bonus tasks.

6. The expected payment to an agent p is given by:

$$u_p(F^p, \{F^q\}_{q \neq p}) = \frac{1}{(l-1)} \sum_{q \neq p} \sum_{i,j} \Delta_{p,q}(i,j).S_{p,q}(F_i^p, F_j^q)$$

we assume agents are clustered into K clusters, denoted by $G_1, .., G_K$. Let $G(p)$ denote the cluster to which the agent p belongs. A clustering is ϵ_1 - accurate, for some $\epsilon_1 \geq 0$, if for every pair of agents $p, q \in P$, $||\Delta_{p,q} - \Delta_{G(p),G(q)}||_1 \leq \epsilon_1$. $\Delta_{G(p),G(q)}$ is cluster Delta matrix between clusters $G(p), G(q)$, defined as

$$\Delta_{G_s,G_t} = \tfrac{1}{|G_s| \times |G_t|} \textstyle\sum_{p \in G_s, q \in G_t} \Delta_{p,q}$$

Having this, lets see the CAHU mechanism (Algorithm 10) proposed for heterogeneous users.

Algorithm 10 CAHU

1: **for** every agent $p \in P$ and **for** every task $b \in M_1^p$ **do**: $q \leftarrow$ uniformly at random conditioned on $b \in M_1^q \cup M_2^q$ ($M_2^p \neq M_2^q$ and $|M_2^p| \geq, 2|M_2^q| \geq 2$)
2: Pick tasks $b' \in M_2^p$ and $b'' \in M_2^q$ randomly such that $b' \neq b''$
3: $S_{G(p),G(q)} \leftarrow \text{Sign}(\overline{\Delta}_{G(p),G(q)})$
4: The reward for agent p for task b is calculated as: $R(y_i, y_p) = S_{G(p),G(q)}(r_b^p, r_b^q) - S_{G(p),G(q)}(r_{b'}{}^p, r_{b''}^q)$

Theorem 2.9. *([54]) With (ϵ_1,ϵ_2)-accurate clustering and learning, mechanism CAHU is ($\epsilon_1 + \epsilon_2$) informed truthful if $min_p u_p{}^*(\mathbf{I}, \{\mathbf{I}\}_{q \neq p}) > \epsilon_1$. In particular,*

1. *For every profile $\{F^q\}_{q \in P}$ and agent $P \in P$, we have $u_p(F^p, \{F^q\}_{q \neq p}) - \epsilon_1 - \epsilon_2$*

2. *For any uninformed strategy $F_0{}^p$, $u_p(F_0^p, \{F^q\}_{q \neq p}) < u_p(\mathbf{I}, \{\mathbf{I}\}_{q \neq p})$*

Peer Prediction with Heterogeneous Tasks (PPHT)

Peer Prediction with Heterogeneous Tasks (PPHT) [55] addresses a setting where each task is associated with different distribution of responses. PPHT mechanism is an extension of CA [48] mechanism, aligning incentives for investing effort without creating opportunities for coordinated manipulations.

Unlike CA mechanism, PPHT tasks need not be *ex ante* identical. Signals for different tasks are drawn independently. Agents are assumed to be exchangeable in their roles in the distribution, with same marginal distributions and joint distributions for any pair of agents. Agents adopts same strategy across all the tasks.

To handle heterogeneous tasks, PPHT modifies the delta matrix for a bonus task to allow for the implied product distribution on signals on penalty tasks and proposes CAH mechanism. CAH mechanism is informed truthful if for each task delta matrix is symmetric and every entry is non-zero.

A Geometric Method to Construct Minimal Peer Prediction Mechanisms

The minimal peer prediction mechanisms are equivalent to power diagrams, a type of weighted Voronoi diagram using geometric perspective. Using computational geometry, Frongillo et al. [56] have shown that many mechanisms are unique up to affine transformations and also introduced a general method to construct new truthful mechanisms.

The classical peer prediction mechanism compares the information reported by two participants and compute a payment rule which ensures that truth revelation is a strategic equilibrium. But this requires too much common knowledge. BTS relaxes common knowledge assumptions but require participants to report both information and prediction report. Hence, BTS mechanism is not minimal. The shadowing method and $\frac{1}{p}$ mechanism are minimal with less assumption on common knowledge.

This work proves that without loss of generality, a minimal peer prediction mechanism can be considered as a power diagram. This allowed to prove uniqueness of several well-known mechanisms up to positive-affine transformations, to construct novel peer prediction mechanisms for new conditions, and to compute the maximally-robust mechanism with respect to agents subjective belief models deviating from the mechanism's.

Personalized Peer Truth Serum for Eliciting Multi-Attribute Personal Data

Personalized Peer Truth Serum (PPTS) [57] considers the problem of eliciting the personal attributes of the agents where the tasks cannot be shared between two agents

PPTS aptly defines which agents can act as peers for one another in settings when agents can't share tasks. It shows that even if such peers are estimated from the reports submitted by the agents, the incentive compatibility is not affected. Authors also extend PPTS mechanism to handle continuous data values instead of only discrete answers.

PPTS mechanism rewards 'surprisingly common' reports and is BIC, with strictly positive expected payoffs in the truthful reporting equilibrium. Heuristic reporting equilibria result in zero expected payoff in the mechanism. 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.

Recent works in multi-task settings achieve informed truthfulness using infinite tasks or requiring a batch of agents to solve the same set of tasks in a single round [58]. We believe that these are strong and impractical assumptions. While we focus on RPTSC, our results do not restrict agent participation and hold for agents solving any finite number of tasks. In the next section, we in detail discuss the importance of the PTSC mechanism and robust version of PTSC – RPTSC with its properties; RPTSC being one of they building block in our framework.

2.4 Robust Peer Truth Serum for Crowdsourcing (RPTSC)

Before we get into Robust Peer Truth Serum for Crowdsourcing (RPTSC), let us first look at an example to illustrate the PTSC mechanism to appreciate the importance of its properties.

Example 2.3. *Consider a set of 10 tasks that are published on the crowdsourcing platform, with each having four possible answers a_1, a_2, a_3, a_4. Each of the tasks is solved by 4 agents. Thus, the requester of the system receives 40 reports that are as shown in Table 2.1. Consider another agent i who solves t_4 and reports $y_i = a_1$. Agent i has a choice between three strategies: trustworthy, deceiving and random. Any rational agent will choose a strategy that maximises its reward. We now analyse which strategy is at the better end for agent i.*

Task	Reports	Ground Truth
t_1	a_1, a_2, a_1, a_3	a_1
t_2	a_1, a_2, a_2, a_2	a_2
t_3	a_1, a_2, a_1, a_1	a_1
t_4	a_1, a_1, a_1, a_3	a_1
t_5	a_1, a_2, a_3, a_3	a_3
t_6	a_1, a_4, a_4, a_4	a_4
t_7	a_1, a_4, a_1, a_1	a_1
t_8	a_1, a_2, a_2, a_2	a_2
t_9	a_1, a_1, a_1, a_1	a_1
t_{10}	a_1, a_2, a_2, a_2	a_2

Table 2.1: Reports received for 10 tasks

In PTSC, the expected reward depends on the probability of getting its report matched with its peer's report. We now calculate the expected rewards agent i observed for different strategies.

Assume that the prior beliefs of the agent is equal to frequency of answers from the collected reports $freq(\cdot)$ (as given in Table 2.2). And once the agent observes its evaluation it updates it beliefs to reflect the distribution $P(x|a) = freq(x|a)$ (as given in Table 2.2).

- **The expected reward for trustworthy strategy**: For trustworthy strategy, the agent exerts efforts to observe the evaluation and reports its true evaluation (say, a_1).

$$E[R(a_1)] = \frac{0.75}{0.5} - 1 = \frac{1}{2}$$

- **The expected reward for a deceiving strategy**: For deceiving strategy, the agent exerts efforts and updates its (*posterior*) beliefs about evaluations of other agents. According to its posterior beliefs, it may report the wrong answer (say a_3).

$$E[R(a_3)] = \frac{0.1}{0.1} - 1 = 0$$

Correct answer		Observed Answer			
		a_1	a_2	a_3	$a4$
a_1	$num(a_1)$	15	2	2	1
	$freq(\cdot\|a_1)$	0.75	0.1	0.1	0.05
a_2	$num(a_2)$	3	9	0	0
	$freq(\cdot\|a_2)$	0.25	0.75	0	0
a_3	$num(a_3)$	1	1	2	0
	$freq(\cdot\|a_3)$	0.25	0.25	0.5	0
a_4	$num(a_4)$	1	0	0	3
	$freq(\cdot\|a_4)$	0.25	0	0	0.75
	num	20	12	4	4
	$freq(\cdot)$	0.5	0.3	0.1	0.1

Table 2.2: Probability of observing different answers, differentiated by the true answers of each task

- **The expected reward for a random strategy**: For random strategy, the agent does not exert efforts; hence, its expected reward only depends on prior beliefs.

$$E[R(a_1)] = 0.5 \times \frac{0.75}{0.5} + 0.3 \times \frac{0.1}{0.3} + 0.1 \times \frac{0.1}{0.1} + 0.1 \times \frac{0.05}{0.1} - 1 = 0$$

Observe that the expected reward for choosing trustworthy strategy is higher compared to the expected reward for the other two strategies. One can also calculate it for different answers across all tasks with the same answer and notice that exerting efforts and reporting true evaluation has the highest reward for each task. That is, choosing trustworthy strategy is beneficial. □

The above example shows that PTSC reward incentivises agents to adopt a trustworthy strategy. However, the PTSC mechanism requires a large number of statistically similar tasks to provide appropriate rewards. RPTSC relaxes this requirement and operates with a smaller number of statistically independent tasks.

We first give the crowdsourcing setting in which the RPTSC mechanism is employed and then formally provide the mechanism and its properties.

2.4.1 Crowdsourcing Setting

Consider a crowdsourcing setting with *statistically independent and a-priori similar tasks*. These tasks are differentiated only by their answers. Requester assigns each task to agents, and each task in the task set $\mathcal{T}$ is assigned to at least two agents. Agents in this setting are assumed to be rational. They solve their assigned tasks either by exerting high (e_H) or low efforts (e_L). And, naturally, the cost of exerting high effort is greater than cost of exerting low effort, i.e., $c(e_H) > c(e_L)$. Any effort that is insufficient to solve the task is considered as low. If an agent i does its reasonable best to solve the task, i.e., exerts high effort, it obtains an evaluation x_i.

2.4.1.1 Agent Beliefs

In RPTSC, agents have private beliefs about other agents. Let P, Q denote agents' beliefs about evaluations, reports, respectively, as defined next.

Agent Beliefs about Evaluations (P)

For an agent i, the prior belief $P_i(x_i)$ denotes the probability of its evaluation being x_i. Consider another agent p who solves the same task as agent i. Then, $P_{p|i}(x_p|x_i)$ is agent i's posterior belief about agent p's evaluation being x_p when its evaluation is x_i. We assume that all the agents' beliefs are *fully mixed*. That is, they satisfy:

$$0 < P_p(x_p), P_{p|i}(x_p|x_i) < 1, \ \forall i, p \in \mathcal{A}, \ \forall x_i, x_p \in \mathcal{X}$$

Since the tasks are statistically independent, if agent i has not solved a task, say τ_k, it has no evaluation for that task, i.e., $x_i = \varnothing$. Therefore, agent i's posterior belief about the evaluation of another agent q for task τ_k is the same as its prior.

Agent Beliefs about Reports (Q)

To decide its best strategy, an agent i estimates its expected reward for reporting an answer y_i based on its beliefs about peers' reports. For this, it transforms its beliefs about peers' evaluations into beliefs about reports: $(P_p, P_{p|i}) \rightarrow (Q_p, Q_{p|i})$. The posterior belief $Q_{p|i}(y_p|x_i)$ is the probability that agent p reports y_p when agent a_i's evaluation is x_i.

For random strategy, agents do not exert efforts; thus, their posterior beliefs are the same as its prior. Formally, if all the agents choose random strategy, we have $Q_p(y) = P_p(y)$ and

$Q_{p|i}(y|\varnothing) = P_{p|i}(y|\varnothing) = P_p(y)$. Likewise, when all the agents choose trustworthy strategy, i.e., all of them report their true evaluations, their beliefs about reports are the same as that of evaluations, i.e., $Q_p(y) = Q_q(y) = P_p(y)$; $Q_{p|i}(y) = P_{p|i}(y)$.

2.4.1.2 Self Predicting Condition

If agent i exerts high effort (e_H) for a task to acquire an evaluation x_i, it develops a posterior belief $P_{p|i}(y_i|x_i)$ regarding the evaluations of peers. RPTSC [29] assumes that this posterior belief has a positive correlation with its evaluation x_i. This is natural to assume and is referred to as the *self predicting condition* (Definition 2.11). In RPTSC self predicting condition is as follows:

$$\frac{P_{p|i}(x_i|x_i)}{P_p(x_i)} > \frac{P_{p|i}(y_i|x_i)}{P_p(y_i)}$$

In RPTSC, if agent i's belief satisfy the self predicting condition, we denote a *self predictor* Δ_i as the least value in $[0, 1]$ that satisfies,

$$\left(\frac{P_{p|i}(x_i|x_i)}{P_p(x_i)} - 1\right)\Delta_i > \frac{P_{p|i}(y_i|x_i)}{P_p(y_i)} - 1 \tag{2.1}$$

Intuitively, the self-predictor Δ_i characterises agent a_i's beliefs about the degree of correlation of its evaluation with peers'. Smaller the Δ_i, greater the belief agent has on its evaluation. If $\Delta_i \approx 1$, the agent a_i is more likely to confuse between different answers. And, if $\Delta_i \approx 0$, the answers do not correlate.

2.4.1.3 Reward

RPTSC uses *surprisingly common principle* using *peer consistency*, for providing rewards to the agents. The reward for the agent who reported y_i is proportional to $\frac{1}{freq(y_i)} - 1$ if its report matches with a randomly chosen peer's report y_p for the same task, where $freq(y_i)$ is the frequency of report calculated from the submitted reports. Otherwise, reward is proportion to -1. To incentivize high efforts, RPTSC has a scaling factor (α) to the reward such that it covers the cost of exerting efforts. To summarize, requester provides a reward $R(y_i, y_p)$ to the agent i on reporting y_i, where

$$R_{RPTSC}(y_i, y_p) = \begin{cases} \alpha\left(\frac{\mathbb{I}_{y_i=y_p}}{freq(y_i)} - 1\right) & \text{if } freq(y_i) \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

The utility of the agent i in RPTSC depends on the cost of effort exerted and the reward received for completing a task, i.e., $u_i(y_i) = R(y_i) - c(e_i)$, where $e_i\{e_H, e_L\}$ is the effort exerted by agent i.

Algorithm 11 formally presents the RPTSC mechanism. With this, we next give the game-theoretic properties of RPTSC.

Algorithm 11 Robust Peer Truth Serum for Crowdsourcing (RPTSC)

1: Requester gives set of statistically similar tasks $\mathcal{T}$ to a set of agents $\mathcal{A}$.
2: The reward for agent i who submitted report y_i is calculated as follows:
3: Consider $n-1$ tasks in addition to task τ, where n satisfies the desirable properties
4: Randomly sample n reports other than agent i's report from n different tasks, including the task τ.
5: Compute the frequency of reported values within the sample. Let $freq(y_i) = \frac{num(y_i)}{\sum_y num(y)}$.
6: Agent i is rewarded:

$$R_{RPTSC}(y_i, y_p) = \begin{cases} \alpha\left(\frac{\mathbb{I}_{y_i=y_p}}{freq(y_i)} - 1\right) & \text{if } freq(y_i) \neq 0 \\ 0 & \text{if } freq(y_i) = 0 \end{cases}$$

2.4.2 RPTSC Properties

Proposition 2.1. [29, Lemma 4.1] *In RPTSC, the expected reward of an agent i with evaluation x_i and report y_i is,*

$$E' = \begin{cases} \alpha\left(\frac{Q_{p|i}(y_i|x_i)}{Q_p(y_i)} - 1\right)\left(1 - (1 - Q_p(y_i))^{n-1}\right), & \textit{if } Q_P(y_i) > 0 \\ 0, & \textit{if } Q_P(y_i) = 0 \end{cases} \tag{2.3}$$

Let M' be the optimal reward of an agent i adopting trustworthy strategy in RPTSC (i.e., if its report matches its peer's report, $q'_{y_i} = 1$ in Prop. 2.3). This implies that,

$$M' = \alpha \left(\frac{1}{Q_p(y_i)} - 1 \right) \left(1 - (1 - Q_p(y_i))^{n-1} \right). \tag{2.4}$$

Furthermore, in RPTSC, the expected reward of an agent i before evaluation when all the other agents choose trustworthy strategy is [29, Equation 8],

$$\overline{R}_i(\alpha) = \mathbb{E}_{x_i \in \mathcal{X}} \left[\alpha \left(\frac{P_{p|i}(x_i|x_i)}{P_p(x_i)} - 1 \right) \left(1 - (1 - P_p(x_i))^{n-1} \right) \right]$$

Proposition 2.2. [29, Theorem 4.3] *Exerting efforts and truthful reporting is strict EPIC in RPTSC, if the following conditions are satisfied,*

$$\mathsf{A} : \overline{R}_i(\alpha) > c(e_H) - c(e_L)$$
$$\mathsf{B}(n) : \frac{1 - (1 - P_i(x_i))^{n-1}}{1 - P_i(x_i)^{n-1}} \geq \Delta_i$$

In Proposition 2.2, note that A implies that the expected reward must be greater than the cost of exerting effort and B is merely a way of representing the self predicting condition (Definition 2.11).

In this work, we use the RPTSC reward scheme to analyse the properties of our proposed framework, REFORM. We focus on RPTSC [29] as it (i) does not assume any prior, (ii) incentivises efforts and trustworthy reporting, and (iii) is resistant to single report strategy. However, like any other PBM, RPTSC also suffers from unfair rewards. In the next section, we briefly discuss a few existing works that improved fairness in PBMs, making strong assumptions like access to gold-standard tasks and localised settings. In this work, we aim to improve fairness in PBMs without such assumptions.

2.5 Fair Reward Mechanisms

In recent times, fairness of reward schemes affecting participation of the crowdsourcing systems is becoming a critical issue. Primarily in PBMs, where agents' reports are evaluated against random peer reports, the rewards can be unfair. In PBMs, a truthful agent can get paired with a random agent, leading to an unfair evaluation. Even though most PBMs assure incentive compatibility, they do not ensure fairness. We now discuss a few works that assure fairness in PBMs.

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

Goel and Falting [31] first studied the challenges in peer based mechanisms and other classical mechanisms which use gold tasks and pay agents accordingly. DBT mechanism assigns gold-standard tasks to a few agents and exploits transitivity to derive accuracy of the rest of the agents from their peers' accuracy. DBT mechanism ensures dominant strategy incentive compatibility and fair rewards to the participating agents.

Crowdsourcing Model

DBT considers a crowdsorucing setting where the tasks given to agents have discrete answer space $\mathcal{X}$. let g be ground truth for the task, x_i be the signal obtained by agent i and y_i be the reported answer. $g, x_i, y_i \in [K]\ \forall i$. Here, the effort strategy of the agent is considered to be binary that is, e_i is either low or high.

- **Reporting Strategy:**
  - When $e_i = 1$, reporting strategy S_i of agent i is a $K \times K$ matrix, where $S_i[x, y]$ is a probability of her reported answer on a task being y given that the observed answer is x.
  - When $e_i = 0$, the reporting strategy S_i is K dimensional probabilistic vector where $S_i[x]$ is the probability of her reported answer on a task being x.
- **Proficiency Matrix:** A $K \times K$ matrix (A_i) where $A_i[g, x]$ is probability that the obtained answers on a task is x given that the ground truth is g.
- **Trustworthiness Matrix:** A $K \times K$ matrix (T_i) where $T_i[g, y]$ is probability that the reported answers on a task is y given that the ground truth is g.

Lemma 2.1. *As $|Q_i \cap Q_j| \to \infty$, the following holds with high probability*

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

where Q_i is the set of tasks assigned to agent i and p is the empirical distribution of answers reported and ω is prior probability of the ground truth of any randomly selected task g.

Algorithm 12 **Deep Bayesian Trust** (DBT)

1: Assign task set to oracle o and obtains its answers.

2: Initialize an *Informative answer pool* (IAP) with the answers given by the oracle.

3: Select some tasks from IAP.

4: Prepare a set of batches of tasks such that each contains tasks selected in previous step and some fresh tasks.

5: Publish the batches on the platform and let agents select a batch they solve.

6: For any agent i who submits her batch, find T_i according to lemma 2.1. Reward agent i with an amount equal to $\beta.\left(\sum_{g\in[K]} T_i[g,g]\right) - 1$, where β is a scaling constant.

7: If the answers of the agent i satisfy informative criteria, add the answers to IAP and assign trustworthiness T_i as obtained in Step 6.

8: Asynchronously repeat steps 3, 4, 5, 6, 7 until desired numbers of answers are collected for all tasks.

Informativeness Criterion

If $p(Y_j = y_j) \neq 0$ and coefficient matrix $\frac{T_j[g,y_j].\omega(g)}{p(Y_j=y_j)}$ is full rank, the informative criterion is said to be satisfied. C^E denote the the cost of effort required to solve batch of tasks.

Theorem 2.10. *([31]) If $\beta > \frac{C^E}{\left(\sum_{g\in[K]} A_i[g,g]\right)-1}$ and $A_i[g,g] > A_i[g',g], \forall g' = g$, then the Deep Bayesian Trust Mechanism*

1. *is dominant strategy incentive compatible (DSIC) for every agent i*

2. *ensures strictly positive expected reward in the truthful strategy.*

Theorem 2.11. *([31]) In the Deep Bayesian Trust Mechanism, a heuristic strategy gives zero expected reward.*

Definition 2.12 (***Fair Incentive Mechanism***)**.** *An incentive mechanism is called* fair *if the expected reward of any agent is directly proportional to the accuracy of the answers reported by her and independent of the strategy and proficiency of her random peer.*

Theorem 2.12. *([31]) DBT is a fair incentive mechanism.*

DBT ensures DSIC and fairness by assuming access to ground truth. We will next look at the fair reward mechanism, FARM, that considers localized settings.

2.5.2 FaRM: Fair Reward Mechanism for Information Aggregation in Spontaneous Localised Settings

FaRM [33] focuses on spontaneous localised settings where tasks are location specific and are required to be answered within a short time. In such settings, information aggregation is challenging as the task answers can only be collected by agents in that area. Moreover, prior knowledge about the answer might not be readily available. Thus, eliciting the reported data is non-trivial. Moti et al. [33] propose a fair reward mechanism for such spontaneous localised settings.

FaRM reward scheme consists of three sub-utility functions, (i) report strength, (ii) consistency score and, (iii) reliability score.

Report Strength ($\Phi(y_i)$)

Report strenth is the count of agents who have reported the same signal as agent i

$$\Phi(y_i) = \sum_{j \in \mathcal{A}} \mathbb{I}_{y_i = y_j}$$

- The report strength of the report of an agent is always positive
- If all other agents report truthfully, then the best response for agent i in order to maximise its sub-utility (report strength) is to report truthfully.

Consistency Score (α)

Consistency score is the reputation an agent receives so far in the mechanism. The score increases for accurate reporting and gets penalised for inaccurate reporting. Here the highest reported answer is used as proxy for ground truth.

$$\alpha_i^t = \begin{cases} \alpha_i^{t-1} - \frac{\alpha_i^{t-1}}{k} \times \frac{(\psi_1 - \Phi(y_i))}{|\mathcal{A}|} & \text{if } \Phi(y_i) < \psi_1 \\ \alpha_i^{t-1} + \frac{1-\alpha_i^{t-1}}{k} \times \frac{(\psi_1 - \psi_2)}{|\mathcal{A}|} & \text{if } \Phi(y_i) = \psi_1 \end{cases}$$

where $k \geq 1$ and ψ is defined as follows:

$$\psi_1 = \max_{s \in \mathcal{X}}(\Phi(s))$$

$$\psi_2 = \begin{cases} \max 2_{s \in \mathcal{X}}(\Phi(s)) & \text{if } \max 2_{s \in \mathcal{X}} > 0 \\ \frac{\psi_1^2 - 1}{\psi_1} & \text{if } \max 2_{s \in \mathcal{X}} = 0 \end{cases}$$

- Consistency Score is bounded by range $[0, 1]$
- If all other agents report truthfully, then the best response for agent i in order to maximise its sub-utility (consistency score) is to report truthfully.

Reliability Score (β)

Reliability score provides incentives to agent i to not collude with her nearby agents $\mathcal{I}_i$. It is the ratio of *external agreement* by *internal agreement.*

$$\beta_i = \frac{\frac{\sum_{j \in \mathcal{E}_i} \mathbb{I}_{y_i = y_j}}{|\mathcal{E}_i|}}{\frac{\sum_{j \in \mathcal{I}_i} \mathbb{I}_{y_i = y_j}}{|\mathcal{I}_i|} + 1}$$

- Reliability Score is bounded by range $[0, 1]$
- If all other agents report truthfully, then the best response for agent i in order to maximise its sub-utility (reliability score) is to report truthfully
- Reliability score prevents agents from colluding with nearby agents

FaRM introduces two notions of fairness, namely selective fairness and cumulative fairness.

Definition 2.13 (***Selective Fairness***)**.** *[33, Definition 4.5] Any two agents $i, j \in \mathcal{A}$ who submit two identical reports y_i and y_j such that $y_i = y_j$. The reward scheme admits selective fairness if, reward is same for both i and j.*

Report strength is selectively fair.

Definition 2.14 (***Cumulative Fairness***)**.** *[33, Definition 4.9] Any two agents $i, j \in \mathcal{A}$ who submit two identical reports y_i and y_j such that $y_i = y_j$. The reward scheme admits cumulative fairness if, reward is more for the agent who is consistently reporting the truth.*

Consistency score is cumulatively fair.

Theorem 2.13. *[33, Theorem 4.15] FaRM is Nash incentive compatible with guaranteed non-negative payoffs and weak budget balanced.*

Proposition 2.3. *[33, Proposition 4.16] FaRM admits selective fairness and cumulative fairness and hence is a fair mechanism.*

Our work focuses on achieving fairness in PBMs without these localised settings or ground truth assumptions. Towards this, we provide trustworthy agents with additional chances to evaluate its report to avoid unfair penalties from the random pairing. That is, a trustworthy agent is provided more chances if its report is evaluated against a less reputed agent. Our framework, REFORM, uses reputation models to quantify the reputation scores of agents in the system. The next section discusses a few existing reputation models used in crowdsourcing systems.

2.6 Reputation Based Reward Mechanisms

In crowdsourcing systems, the quality and reputation scores of the participating agents are considered to be the level of trust the system can place on the services/reports received from them. To improve the quality and effectiveness of a crowdsourcing system, the requesters of the systems value the reports submitted by high reputed agents more than the less reputed agents. We now discuss different reputation models used in the literature.

2.6.1 Are You Contributing Trustworthy Data? The Case for a Reputation System in Participatory Sensing

Participatory sensing is a revolutionary new paradigm in which volunteers collect and share information from their local environment using mobile phones. The inherent openness of this platform makes it easy to contribute corrupted data. Hang et al. [59] proposes a novel reputation system that employs the Gompertz function for computing device reputation score as a reflection of the trustworthiness of the contributed data.

Previously, to maintain the quality of the systems, Ganeriwal et al. [60] proposed a reputation framework referred to as RFSN, to counter faulty and misbehaving nodes in traditional embedded wireless sensor networks. Beta distribution has been employed in [61], where the authors address the problem of selecting suitable participants for participatory sensing applications. The problem of verifying data received from user devices in participatory sensing was also studied and their solutions rely on auxiliary trusted platform module (TPM)

In [59], the architecture of the system primarily consists of: *watchdog module* and *reputation module*, both are implemented at the application server. The system can readily work with any typical participatory sensing applications. Let us assume n devices contributing data within a particular grid. The watchdog module processes sensor values from these n devices in epochs of duration T. For every epoch k, the sensor values from device i is denoted by a vector $X_{i,k} = [x_{i,t}, \ldots, x_{i,t+T-1}]\ \forall i$ with $t = (k-1) \times T + 1$. The watchdog module executes an outlier detection algorithm on the vector $X_{i,k}$ and produce a set of cooperative ratings, $p_{i,k}$ for each device i in epoch k. For each epoch k, the reputation module incorporates past cooperative ratings and computes reputation scores, $R_{i,k}$, for each device i.

The instantaneous average values for epoch k are computed as :

$$r_t = \sum_{i=1}^{n} p_{i,k}.x_{i,t}, \quad (k-1) \times T < t \leq k \times T \tag{2.5}$$

The above Equation 2.5 becomes robust average if $p_{i,k}$ is computed as follows:

$$p_{i,k} = \frac{\frac{1}{\frac{\sum_{t=1}^{T}(x_{i,t}-r_t)^2}{\sum_{i=1}^{n}\sum_{t=1}^{T}(x_{i,t}-r_t)^2}+\epsilon}}{\sum_{j=1}^{n}\frac{\frac{1}{\sum_{t=1}^{T}(x_{i,t}-r_t)^2}}{\sum_{i=1}^{n}\sum_{t=1}^{T}(x_{i,t}-r_t)^2}} + \epsilon \tag{2.6}$$

Reputation module uses the above epoch-based ratings to build a long term view of the trustworthiness of each device. This gradually builds up trust in a person after several instances of trustworthy behaviour and rapidly tear down the reputation for this individual if there is any dishonest behaviour using Gompertz function.

Algorithm 13 Iterated Outlier Detection

Let $p^l_{i,k}$ and r^l_t be the values of $p_{i,k}$ and r_t at the l^{th} iteration, respectively

1. Initialise $l = 0$ and $p^l_{i,k} = \frac{1}{n}$
2. Compute r^{l+1}_t from $p^l_{i,k}$ using Equation 2.5
3. Compute $p^l_{i,k}$ from r^l_t using Equation 2.6
4. $l \leftarrow l + 1$
5. Start from Step 2 if no convergence.

The system proposed is well-suited to quickly adapt to the transitions in user behaviour. The reputation scheme is implemented in real world participatory sensing application for monitoring noise pollution in urban environment and achieved three-fold improvement in comparison with the state-of-the-art Beta reputation scheme.

2.6.2 Quality-Aware and Fine-Grained Incentive Mechanisms for Mobile Crowdsensing

Wang et al. [9] introduces optimal expected expenditure by characterising the quality of recruited crowd and improves flexibility and effectiveness by presenting a reserve auction based incentive mechanism for quality-aware and fine-grained mobile crowdsensing (MCS).

A service user can make a sensing service request via a web portal. The request is then analysed by the cloud operator, that uses an incentive mechanism to recruit a sensing crowd (a set of mobile users) and distribute the request to them. Their smartphones perform the corresponding sensing activities and report sensor data to the cloud operator. The cloud operator aggregates and analyses sensor data, and sends results back to the service user through the web portal.

To incentivize the mobile user, a reverse auction based incentive mechanism is used which enables fair pricing between cloud operator and mobile users in MCS. A fine-grained

MCS is considered, in which each sensing task consists of multiple sub tasks and a mobile user may make contributions to multiple sub tasks.

Auction Formulation

1. The cloud operator (the buyer) announces a sensing task to mobile users (bidders and sellers).

2. Each mobile user i submits a bid $b_i = (w_i, Z_i)$, where w_i, Z_i are mobile user i's declared cost and quality vector.

3. The cloud operator uses an incentive mechanism to select the winners and determine payments.

4. Winners carry out the sensing task and submit results to the cloud operator.

5. The cloud operator checks the results and makes payments to winners.

Quality Aware Incentive Mechanism

This consists of two sub problems: *Winner Selection* and *Payment Determination* Consider M mobile users, and x be the winner selection vector. q_i quantifies the quality of services/data the sensing crowd is potentially capable of providing for a sub-task j. β_i denotes virtual cost of user i

Winner Selection: Formulated as IP problem

$$\min_{i=1}^{M} \beta_i(w_i)x_i$$

Subjected to

$$q_j = g_j(\mathbf{Z}, \mathbf{X}) \geq r_j \forall j \in \{1, \ldots, N\}$$

$$x_i \in \{0, 1\}$$

Payment Determination Let $\Omega(\mathbf{B})$ denote optimal value of IP-Winner and $\Omega(\mathbf{B}_i)$ denote optimal value of IP-Winner with bid b_i removed.

$$p_i = \begin{cases} \beta_i^{-1}(\Omega(\mathbf{B}_{-i}) - (\Omega(\mathbf{B}) - \beta_i(w_i))), & \text{if } x_i = 1 \\ 0, & \text{otherwise} \end{cases}$$

The authors provide a truthful, individually rational, and computationally efficient algorithm for Winner Selection from the bids submitted in the reverse auction and Payment Determination to determine the payment for winning mobile user in of Quality-aware Incentive Mechanism (QIM).

2.6.3 Quantifying User Reputation Scores, Data Trustworthiness, and User Incentives in Mobile Crowd-Sensing

In mobile crowdsensing correctness and truthfulness of the acquired data must be verified, because the users might provide incorrect or inaccurate data, whether due to malicious intent or malfunctioning devices. So, authors in [62] introduce a new metric, *Collaborative reputation scores* and can provide an effective alternative to the previously proposed metrics.

There are three component of mobile crowd-sourcing:

1. User Recruitment
2. Platform Utility and User Utility
3. True Payments and False Payments

The goal of any successful MCS system is to maximise platform utility by compensating the users sufficiently, which will keep the user utility at an acceptable minimum. The third metric false payments, must be minimised to avoid paying for bad information.

Two primary factors that contribute to user reputation are:

1. The sensory accuracy or the possibility of device malfunction, i.e. Hard reputation.
2. The average probability of inaccurate or outright wrong readings that stem from malicious intelligence (either malicious users manipulating readings or a virus causing incorrect reporting),i.e. Soft Reputation.

Thus, data trustworthiness of a user i ($\mathfrak{T}_i$) is function of hard and soft reputation. It is defined as follows:

$$\mathfrak{T}_i = \begin{cases} f(R_i^{hard}, R_i^{soft}), & \text{if } Q_i < Q^{TH} \\ R_i^{soft} = R_i, & \text{if } Q_i \geq Q^{TH} \end{cases}$$

where Q_i is the accuracy of hardware sensors of user i and Q^{TH} is accuracy threshold.

Three Reputation score based MCS are discussed in the paper:

- Statistical Reputation
- Voted Reputation
- Anchor-Assisted Decentralised Reputation

Using collaborative reputation scores in user recruitment improves platform utility and data trustworthiness by reducing false payments. When collaborative methods are employed, using statistical reputation in the assessment of the value of a recruited crowd can reduce the user bias in the decentralised vote-based component of the reputation score.

2.6.4 Reputation-based Worker Filtering in Crowdsourcing

The problem of aggregating noisy labels from crowdsourcing reports is not very trivial. To infer true labels of binary tasks, a computationally efficient reputation algorithm [24] to identify and filter out adversarial workers in crowdsourcing systems is proposed.

Model

Consider set of binary tasks $\mathcal{T}$ having true labels in $\{-1, +1\}$ and worker set W. $w_i(t_j)$ denote label provided by worker to task t_j, $\mathcal{L} = w_i(t_j)$ where $\mathcal{L} \in \{-1, 0, +1\}$. $\mathcal{T}_{cs}$ is conflict set which has both '+1' and '-1' labels. d_j^+ and d_j^- denote the number of workers labelling task t_j as 1 and -1 respectively.

In order to overcome over-penalising honest workers, Two techniques of penalty are considered. (i) Soft Penalty (Algorithm 14), (ii) Hard Penalty (Algorithm 15).

Hard Penalty

It addresses the case sophisticated adversaries, the key idea is not to distribute the penalty evenly across all the workers. This uses the concept of semi-optimal matching on a bipartite graph. In a bipartite graph $B = (U, V, E)$, a semi-matching in B is a set of edges $M \in E$ such that each vertex in V is incident to exactly one edge in M. $deg_M(u)$ denote the number of u is incident on in M and cost is defined as

$$cost_M(u) = \sum_{i=1}^{deg_M(u)} i = \frac{deg_M(u)(deg_M(u) + 1)}{2}$$

.

Algorithm 14 **Soft Penalty**

1. Input: $W, \mathcal{T}, \mathcal{L}$

2. For every task $t_j \in \mathcal{T}_{cs}$, assign penalty s_{ij} to each worker $w_i \in W_j$ as follows:

$$s_{ij} = \frac{1}{d_j^+} \quad \text{if } \mathcal{L}_{ij} = 1$$

$$s_{ij} = \frac{1}{d_j^-} \quad \text{if } \mathcal{L}_{ij} = -1$$

3. Output: Penalty of worker w_i

$$pen(w_i) = \frac{\sum_{t_j \in \mathcal{T}_i \cap \mathcal{T}_{cs}} s_{ij}}{|\mathcal{T}_i \cap \mathcal{T}_{cs}|}$$

Algorithm 15 **Hard Penalty**

1. Input: $W, \mathcal{T}, \mathcal{L}$

2. Create a bipartite graph B^{cs} as follows:

   i. Each worker $w_i \in W$ is represented by a node on the left

   ii. Each task $t_j \in \mathcal{T}_{cs}$ is represented by two nodes on the right t_j^+ and t_j^-

   iii. Add the edge (w_i, t_j^+) if $\mathcal{L}_{ij} = 1$ or edge (w_i, t_j^-) if $\mathcal{L}_{ij} = -1$

3. Compute an optimal semi-matching OSM on B^{cs} and let d_i be the degree of w_i in OSM

4. Output: Penalty of worker w_i

$$pen(w_i) = d_i$$

The optimal semi-matching minimises:

$$\sum_{u \in U} cost_M(u)$$

The reputation based worker filtering that uses disagreement-based penalties and optimal semi-matching to identify adversarial workers is proposed. Shows that our reputation scores are consistent and algorithm can be applied to real crowd-sourced datasets.

2.6.5 Identifying Vulnerabilities in Trust and Reputation Systems

To evaluate trust and reputation systems against known attacks, [63] presents a method to automatically identify vulnerabilities in existing trust models. To provide reliable and objective means to assess how these systems are towards different kinds of attacks.

Previously, BRS (Beta Reputation System) with filtering [64], focused on excluding attackers who provide unfair feedback by badmouthing or ballot-stuffing. The TRAVOS [65] discounted outlying ratings in making trust assessments. The HABIT [66] model uses a hierarchical Bayesian model to identify participants with various profiles of reliability, and factor into aggregated ratings.

The contributions made here are three-fold. (i) Model coordinated, strategic attacks with a specific objective as a derivative-free optimisation problem. (ii) Two search methods are proposed for efficiently identifying coordinated attacks in complex attack spaces through sampling-based optimisation. (iii) This method is used to analyse a selection of existing trust models, providing evidence for the kinds of complex attacks they are vulnerable to.

Prediction of the future behaviour of an agent (i.e. a trust assessment) at time t is, $\varepsilon = \{O^{0:t}_{c_i->p_i} | c_i \in C, p_i \in P\}$. [63] investigates the cases in which an attacker is limited by: **Power**, the number of observations that it can add through the attack ($\rho = |\varepsilon^{'}|$) and **Control** over the witnesses ($W^{'} \subseteq W$).

2.6.5.1 Attack Space

- The space of possible attacks is χ,

$$|\chi| = \begin{cases} \rho + k.|\{O^{0:t}_{w_i->p_j} | w_i \in W^{'}, p_i \in P\}| - 1 \\ k.|\{O^{0:t}_{w_i->p_j} | w_i \in W^{'}, p_i \in P\}|. \end{cases}$$

- The space of attacks is defined in terms of:
  - The number of witnesses to be used, s.
  - The distribution of the attack power, ρ across these selected witnesses, considering those they can report on:
    i. All restricted partitions of ρ into s ($D = RP_s(\rho)$) and their permutations without repetition: ${P_s}^D$
    ii. The distribution of these permutations to each witness-provider pair, such that the number of possible distributions is $(|P|.k)^s$
- The number of attacks in reduced space is,
  $|\chi| = \binom{|W^{'}|}{s} D.{P_s}^D.(|P|.k)^s$
- To solve attackers optimisation problem, 'Monte Carlo Sampling' or 'Hierarchical Sampling' based techniques are used.

A novel method for identifying vulnerabilities in trust and reputation systems is introduced. Model when employed to search for effective strategies through derivative-free optimisation methods, output a set of attack profiles and an estimate of the vulnerability of the TRS to an attack of that kind.

In this chapter, we have built foundations for designing truthful crowdsourcing mechanisms, especially peer-based mechanisms. We also brought up the fairness issues in such mechanisms. In the next chapter, we illustrate how we propose to improve and quantify fairness in PBMs.

Chapter 3

REFORM: Reputation Based Fair Reward Framework for Crowdsourcing

"Fairness is giving all people the treatment they earn and deserve. It doesn't mean treating everyone alike."

– John Wooden, Wooden on Leadership: How to Create a Winning Organization

While existing PBMs incentivize efforts and truthful reporting, agents still cannot be perfectly reliable as they may have noisy observations or be malicious. As observed in Chapter 2, PBMs are inherently unfair as an agent's reward depends on its consistency with randomly selected peers' reports. This lack of fair rewards in PBMs was first observed in [67]. In such a case, an agent with trustworthy strategy may not get the reward it deserves from an unfair pairing. We believe fair rewards are necessary to ensure the participation of these trustworthy agents in crowdsourcing. As such, we build the theory for improving fairness in PBMs in this chapter. We compare a naive approach and our creative approach to improve fairness in PBMs and show that our approach performs well in Section 3.1. With this, we build an abstract framework, REFORM, towards fair rewards in crowdsourcing (Section 3.2). Later, in Section 3.3, we propose two novel notions of fairness in PBMs, namely, γ-fairness and Qualitative fairness.

3.1 Fairer Rewards in PBMs

To improve fairness in PBMs, we should reduce the penalty agents receive from unfair pairings. Increasing the expected rewards of the agents is one way to reduce the penalty. For an increase in expected rewards, a naive approach can be to pair an agent with multiple peers and reward the agent with the average reward obtained from each pairing. In this section, we formally state this approach below and compare it with our approach.

3.1.1 Naive Approach

Naive Approach. *Reward a particular agent with the average (or weighted average) reward obtained after evaluating its report across multiple reports, say w.*

To calculate the expected reward of Naive Approach, consider a trustworthy agent who reports y for some task. Let the optimal reward it obtains when its report matches its peer's report y_p be $g = \textit{peer-fac}(y|y_p = y)$ and penalty obtained when reports do not match be $l = \textit{peer-fac}(y|y_p \neq y)$. Clearly, the reward g obtained when reports match must be greater than the reward l obtained when reports do not match (e.g., [27, 29]).

Expected reward with Naive Approach

Let us consider that the probability of an agent's report matching with its random peer's report is $0 \leq z \leq 1$. Now, averaging reward across w such reports, the expected reward of an agent is,

$$\frac{1}{w}\left(z\left(w \times g\right) + \left(1 - z\right)\left(w \times l\right)\right) = z \times g + \left(1 - z\right) \times l$$

This reduces the penalty obtained from unfair pairings for a trustworthy agent to some extent. However, it also assures higher expected rewards for random agents, increasing the overall budget, which is not desirable. Thus, we aim for a reward scheme that guarantees better fairness for trustworthy agents while discouraging random reporting. Towards this, we present our approach.

3.1.2 Our Approach

As discussed, the Naive Approach may increase the expected rewards even for the random reporting agents. These extra rewards will increase the requester's overall budget and is not desirable. Towards this, we propose a new approach that improves fairness by increasing expected rewards only for trustworthy agents.

> *"You may feel that I have double standards, as I certainly will not treat you all the same. However, I will attempt to give each agent the treatment that he earns and deserves according to my judgement. I know I will not be right in all of my decisions, but I will attempt to be both right and fair."*
>
> *– American basketball coach John Wooden.*

Like John Wooden said, even in crowdsourcing systems, every agent should get a reward it deserves. That is, every agent should receive a reward proportionally equivalent to the effort it exerts in solving tasks and reporting its answers. This assures agents fair rewards and encourages them to participate in these systems. Motivated by this, our idea to improve fairness in PBMs is briefly discussed below.

The ingenuity of our approach in is to *only* give trustworthy agents *additional chances* of pairing to evaluate their reports, which reduces the possibility of agents getting penalised for unfair pairings. This decrease in unfair penalty leads to higher expected rewards while simultaneously restricting the increase in the expected rewards of agents employing a random strategy. We use a *reputation model* to decide which agent will receive the additional chance(s).

Ingenious Approach. *Provide agents with additional chances of pairing, say k, to evaluate their reports when paired with a less reputed agent.*

Next, we show that Naive Approach (Section 3.1.1) will provide lower expected rewards than our Ingenious Approach, which provides additional chances only to reputed agents.

Expected reward with Ingenious Approach

For the Ingenious Approach, consider the least number of additional chances of pairing, i.e., $k = 2$. The agent is given another chance when its report does not match, and if its reputation score is higher than its peer's in the first matching. Here, we assume that an agent's reputation is more than its peer with probability $r \in [0, 1]$. Hence, the expected reward is calculated as follows:

1. The report matches with the peer's report with probability z and the reward is g
2. And with probability $(1-z)$ reports do not match. Hence when the agent's reputation is lower than that of its peer's (with probability $1-r$) the reward is l.
3. However, with probability r the reputation score of agent is higher than its peer's and gets one more round of pairing to match its report.
4. By putting all these together we get the following expected reward.

$$
\begin{aligned}
& z \times g + (1-z)\left((1-r)l + r\left(z \times g + (1-z) \times l\right)\right) \\
& = z\left(r + (1-rz)\right)g + (1-z)\left(1-rz\right)l
\end{aligned}
$$

Trivially, the expected reward in the Ingenious Approach is greater and *closer to the optimal reward* g compared to the expected reward in the Naive Approach (Section 3.1.1). Hence, we adopt Ingenious Approach and build a novel framework for fair rewards, namely, ***REFORM*** – **REputation based Fair and tempOral Reward fraMework for Crowdsourcing**. In the next section, we formally present our framework, REFORM.

3.2 REFORM: Framework

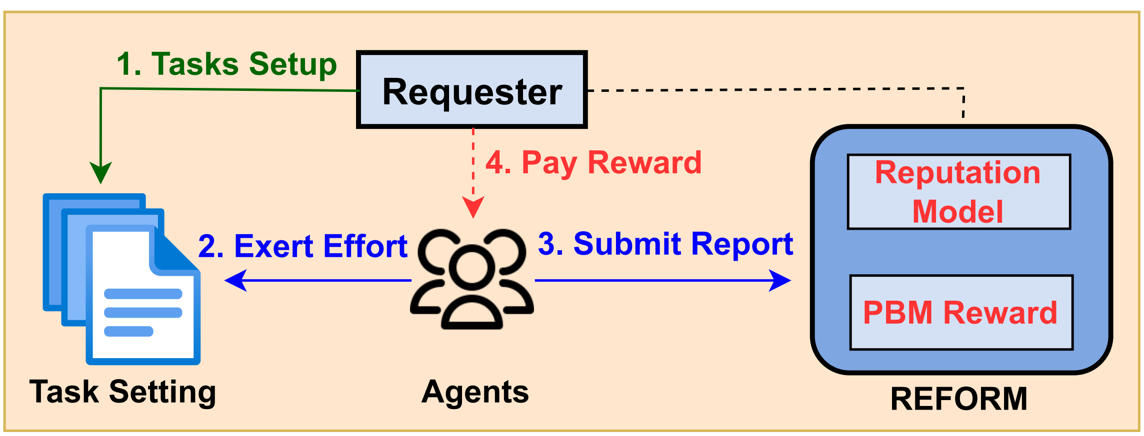

Figure 3.1: REFORM: Overview of the Framework

Using Ingenious Approach (Section 3.1.2), we design an *iterative framework* for crowdsourcing. REFORM incentivizes an agent to report truthfully by improving the expected reward of reputed agents (i.e., agents with higher reputation scores). We increase the expected reward by offering reputed agents additional chance(s) of pairing.

Algorithm 16 REFORM

Agent a_i submits a report y_i for an assigned task τ in round r_j after time t_i.

Input: *peer-fac*$(\cdot)$, *Rep-score*$(\cdot)$, $k \in \mathbb{Z}^+$, y_i

Output: $R_i(y_i, y_p)$

Initialisation: $l = 0$

```
while l < k do
    Randomly choose peer report y_p from the same task τ.
    if l = 1 then
        Ω_{i,j} = Rep-score(y_i, t_i)                      ▷ update reputation score
    end if
    l = l + 1
    if y_i = y_p then
        R_i(y_i, y_p) = peer-fac(y_i | y_i = y_p)     ▷ If reports match agent gets optimal reward
    else
        if Ω_{i,j} ≤ Ω_{p,j} ∨ l = k then
            R_i(y_i, y_p) = peer-fac(y_i | y_i ≠ y_p)     ▷ reputation score is less or maximum chances reached, no more pairing
        end if
    end if
end while
```

Algorithm 16 formally presents REFORM framework. In Algorithm 16, *peer-fac*$(\cdot)$ may be the reward scheme of any PBM and *Rep-score*$(\cdot)$ any relevant reputation model (e.g., [10, 59]). We schematically depict our framework, REFORM in Figure 3.1. We use

$\Omega_{i,j}$ to denote the reputation score of agent i in round r_j. Based on the reward scheme adopted, we evaluate an agent's report against a randomly chosen peer's report from the same task and reward the agent if the reports match (Line 8). However, if the reports do not match, we offer an additional chance for an agent that consistently behaves trustworthy. To determine consistent trustworthiness, we use reputation scores. That is, if an agent's submitted report does not match with its peer's, and if its reputation score is *higher* than that of its peer without having reached the maximum number of chances k, we match the agent with another peer to evaluate its answer. However, if the reputation score is low, we penalise the agent according to the reward scheme adopted (Lines 10, 11).

We now introduce two novel notions of fairness in PBMs to quantitatively assert the fairness achieved with our framework compared to other PBMs.

3.3 Quantifying Fairness in PBMs

3.3.1 γ- Fairness

As we have discussed, the unfairness in PBMs is due to the trustworthy agents getting penalized from the pairings where their report is evaluated against a random/malicious agent. Our framework improves the expected rewards of a trustworthy agent by minimizing the effect of unfair penalties through additional chances. To compare fairness across different PBMs, we present a notion of fairness which depends on the difference in optimal and expected rewards of a trustworthy agent in a PBM. We refer to it as *γ-Fairness*. We believe that this is the first general notion of quantifying fairness in PBMs.

More formally, let M^* be the optimal reward obtained by an agent choosing trustworthy when its report matches with a peer's report in a given PBM. Also, let E^* be the agent's expected reward. having this we define γ-fairness as follows,

Definition 3.1 (***γ-Fairness***)**.** *A PBM is γ-Fair if the expected difference in its optimal and the expected reward taken over all possible reports equals γ, that is,* $\mathbb{E}_{x\in\mathcal{X}}\left[\frac{M^*-E^*}{M^*}\right] = \gamma$.

γ-Fairness measures the *proximity* of a PBM's expected reward with the optimal reward. Naturally, smaller values γ imply closer expected and optimal rewards. Thus, lesser the γ, fairer the PBM.

γ - *Fairness for RPTSC*

Proposition 3.1. *For any $a_i \in \mathcal{A}$, RPTSC is γ-fair with $\gamma = \sum_{x_i \in \mathcal{X}} Q_p(x_i) \left(\frac{1-Q_{p|i}(y_i|x_i)}{1-Q_p(x_i)}\right)$.*

Proof. Using $E^* = E'$ (Proposition 2.3) and $M^* = M'$ (Equation 2.4), we have γ value for RPTSC as,

$$
\begin{aligned}
&\mathbb{E}_{x_i \in \mathcal{X}} \left[\frac{M' - E'}{M'}\right] \\
&= \mathbb{E}_{x_i \in \mathcal{X}} \left(\frac{1 - Q_{p|i}(y_i|x_i)}{1 - Q_p(x_i)}\right) \\
&= \sum_{x_i \in \mathcal{X}} Q_p(x_i) \left(\frac{1 - Q_{p|i}(y_i|x_i)}{1 - Q_p(x_i)}\right)
\end{aligned}
$$

From γ-fairness definition, RPTSC is $\sum_{x_i \in \mathcal{X}} Q_p(x_i) \left(\frac{1-Q_{p|i}(y_i|x_i)}{1-Q_p(x_i)}\right)$-fair. □

3.3.2 Qualitative Fairness

In mechanisms that deploy reputation scores, it is desirable to prioritise an agent with a better reputation score over an agent with a lesser score. A report from the agent who promptly submits truth is always valuable compared to the report from an agent with an arbitrary history of reporting.

Example 3.1. *For instance, consider a peer grading scenario where an instructor asks two students, A and B, to grade another student C's answer script. A and B report their grades, which do not match. The instructor observes that student B always misreports to reduce competition. However, student A has been genuine and reports truthfully. Thus, the instructor values student A's report more than that of B's. In such a scenario, it is fair to reward student A more than student B, i.e., the reward for student A must be more proportional to its efforts and reporting behaviour.* □

Similarly, the expected reward of an agent with a higher reputation should be high compared to an agent with less reputation. We capture this desired property with the notion of new fairness, which we refer to as *Qualitative Fairness*. We define qualitative fairness in similar lines of cumulative fairness (Definition 2.14); the key difference here is that cumulative fairness provides a score (factor of the overall reward) in proportion to previous round's score, whereas, we provide expected reward in proportion to reputation scores, which considers the entire history of agent's submissions.

For the formal definition, consider a reputation model *Rep-score*$(y_i, \cdot)$ which outputs an agent's reputation score Ω_i based on its report y_i and other arbitrary inputs. With this,

Definition 3.2 (***Qualitative Fairness***)**.** *For agents a_i, $a_j \in \mathcal{A}$ that report y_i, y_j such that $y_i = y_j = y \in \mathcal{X}$, we say a PBM satisfies qualitative fairness if its rewards satisfy,*

$$\mathbb{E}[R_i(y_i = y)|\Omega_i] \geq \mathbb{E}[R_j(y_j = y)|\Omega_j] \quad \forall\ \Omega_i \geq \Omega_j.$$

Here, $\mathbb{E}[R_i(y_i = y)|\Omega_i]$ is expected reward of agent a_i with reputation score Ω_i for reporting y_i.

Discussion

We have seen that our framework, REFORM, provides agents with higher reputation scores with extra chances of pairing to evaluate their reports; this reduces the agents' chances of getting penalised for unfair matching, which leads to an increase in the expected rewards. Thus, we intuitively observe that the expected rewards of the agents are proportional to their reputation scores, satisfying qualitative fairness. We discuss this formally in later chapters.

As we have seen, Ingenious Approach requires a quantification of the trustworthiness of agents to decide which agent will receive additional chances. Research has shown that reputation scores successfully quantify the trust a crowdsourcing system must place on an individual agent based on its history. However, no reputation model exists in the literature, which factors in the *time taken to submit the report*. Consequently, in the next chapter, we introduce a reputation model, TERM, to quantify trustworthiness in a temporal setting. However, note that we can adopt any suitable reputation model in our framework, REFORM.

Chapter 4

TERM: Temporal Reputation Model

As we have seen in Chapter 3, the crucial idea of this work is to provide an additional chance(s) to reputed agents if their answers do not match with their peers while rewarding. The provision of these additional chances will increase their expected reward and therefore improve fairness. However, we need to quantify the reliability of agents. In crowdsourcing systems, mechanisms introduce *reputation score* of an agent as a parameter of trust the system places in its submitted report. The system gradually builds up this *trust* in the agent after several instances of trustworthy behaviour and diminishes relatively quickly if the system observes adversarial behaviour. This logic applies to any trust-based system such as Amazon Mechanical Turk [68], Crowdflower [69].

Typically, reputation scores require to satisfy the following:

1. Builds trust in the agent gradually with honest behaviour.

2. To incorporate temporal setting, the increase in scores should be inversely proportional to the time taken to report.

3. The score growth should decrease as it reaches the extreme and should not cross the maximum score allowed.

In general, the existing literature for a reputation model in crowdsourcing only factors the report submitted by the agents [9, 24, 59]. Allahbakhsh et al. [70] consider the time taken for evaluation to estimate the 'quality' of an evaluator. However, we cannot directly adopt this metric as the characteristics of a strategic/malicious agent will differ from that of an evaluator. Therefore, there is a need to design a reputation model that considers the time the agent takes to submit its report to work in a temporal setting. Towards this, we propose *Temporal Reputation Model (TERM)*, which assigns TERM scores to agents considering both the accuracy of the report and the time taken to submit.

4.1 Computation of TERM scores

For TERM to satisfy the properties mentioned above, we use *Gompertz function* [59] whose variation is gradual, smooth, and is well suited for the model. Gompertz function is a particular case of sigmoid function, in which the growth at the start and end is slow. Several crowdsourcing mechanisms deploy this function to measure trust [10, 59, 71]. Moreover, In TERM, to calculate term scores we maintain the agents' history of reporting.

History ($\mathcal{H}$)

We maintain a history $\mathcal{H}$ of all the scores for every agent in each round. Let $\mathcal{H}_{i,j} = (\Omega_{i,j}, |\phi|_{i,j}, |\phi|_{i,j-1}, \ldots, |\phi|_{i,1})$ denote the history of agent a_i till the round r_j, where $\Omega_{i,j}$ is the TERM score, $|\phi|_{i,j}$ is the normalised round-score obtained for the report submitted in the round r_j.

With this, let $\Omega_{i,j}$ be the TERM score an agent a_i obtains after round r_j. We define TERM score as,

$$\Omega_{i,j} = TERM(\psi_{i,j}) = a \times \exp(b \times \exp(c \times \psi_{i,j})) \qquad \text{(TERM)}$$

where, $TERM(\cdot)$ is the Gompertz function with parameters $a \in \mathbb{R}$ controls the asymptote, $b \in \mathbb{R}^-$ sets the displacement, and $c \in \mathbb{R}^-$ controls the growth rate of the curve. We set $a = 1$, $b = -1$, $c = -1/2$ for a smooth growth of TERM scores, as shown in Figure 4.1 (blue curve). Here, the input to Gompertz function is cumulative score $\psi_{i,j}$ (defined in Equation 4.2)

Further, $\phi_{i,j}$ is *round-score* obtained by agent a_i for submitting y_i in the round r_i after time t_i. We define it as follows,

$$\phi_{i,j} = \frac{\mathbb{I}_{y_i=y_p}}{freq(y_i)t_i} \qquad (4.1)$$

Where y_p is the random peer's report chosen from the same task. The round-score includes the report submitted and time taken by the agent in its calculation. We *map* all the round-scores to $[-1, 1]$ and use these normalised round-scores for computing TERM score. The mapping is done as given below.

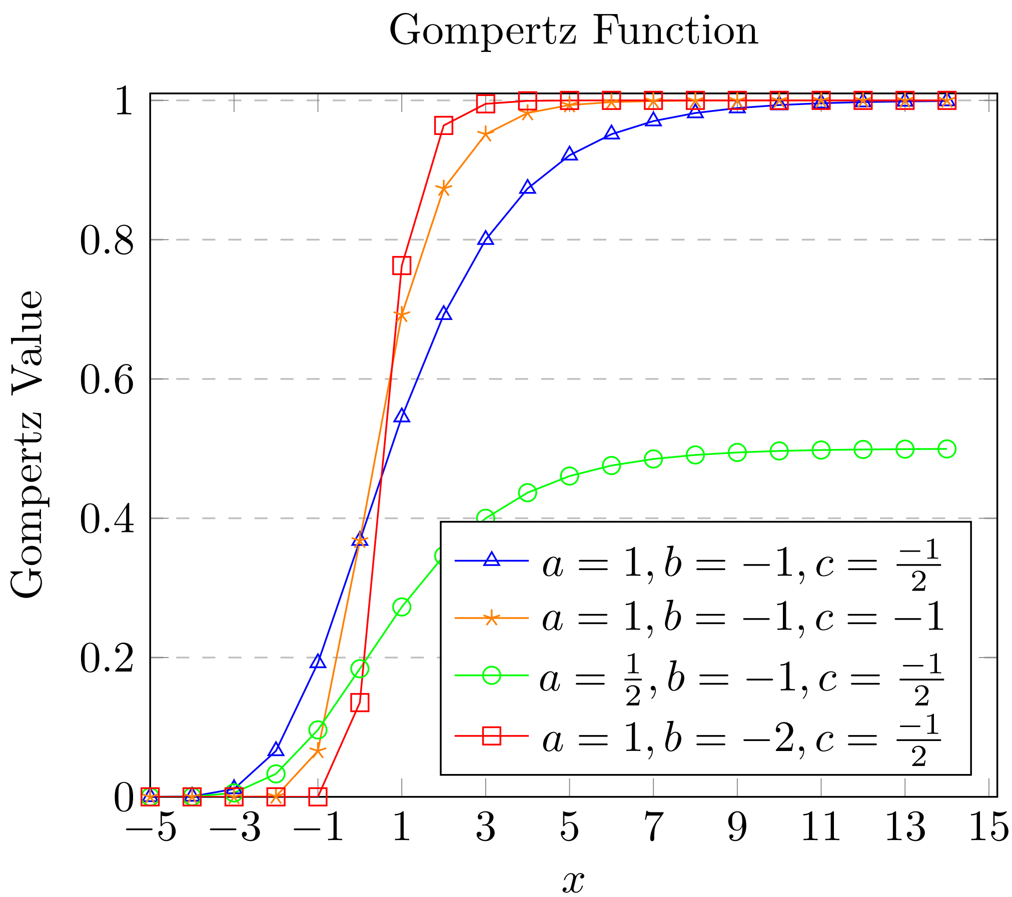

Figure 4.1: Gompertz Function

$$|\phi|_{i,j} = \begin{cases} \frac{\phi_{i,j} - min(\phi^j)}{max(\phi^j) - min(\phi^j)}, & \text{if } max(\phi^j) \neq min(\phi^j) \\ 0, & \text{otherwise} \end{cases}$$

Here, $min(\phi^j)$ and $max(\phi^j)$ denote that minimum and maximum round-scores in the round r_j, respectively.

TERM score is an aggregate of *all* the submissions made by an agent. As desired, in TERM, we account for the fact that recent submissions are more relevant than the previous ones. To do so, we calculate *cumulative-score* $\psi_{i,j}$ of each agent a_i by taking all the normalised round-scores ($|\phi|_{i,j}$) it obtained till the latest round, as follows,

$$\psi_{i,j} = \sum_{k=1}^{j} \lambda^{(j-k)} |\phi|_{i,k} \quad (0 < \lambda < 1) \tag{4.2}$$

Trivially, $\lambda^{(j-k)}$ gradually reduces the impact of previous round-scores.

Algorithm 17 presents TERM, here, the requester maintains frequency $freq(y_i)$ of the report y_i. TERM calculates normalised round-scores of agents from the reports submitted and time taken for reporting (Lines 2-3). The cumulative-score calculation uses all the obtained normalised round scores until the latest round (Line 4). We take cumulative-score as input to the Gompertz function, whose output is the TERM score (Line 5). We now formally give properties of the proposed reputation model TERM.

Algorithm 17 TERM

Agent a_i with history $\mathcal{H}_{i,j-1}$ submits report y_i for a task τ in round r_j at time t_i.

Input: Report y_i, Time taken t_i

Output: Updated TERM score $\Omega_{i,j}$

1: Randomly choose a report y_p of agent a_p from the same task τ

2: $\phi_{i,j} = \frac{\mathbb{I}_{y_i=y_p}}{freq(y_i)t_i}$ ▷ round-scores calculation

3: $|\phi|_{i,j} \leftarrow$ normalised $\phi_{i,j}$

4: $\psi_{i,j} = \sum_{k=1}^{j} \lambda^{(j-k)} |\phi|_{i,k}$ ▷ cumulative-scores calculation

5: **Return**: $\Omega_{i,j} = \exp(-\exp(\frac{-\psi_{i,j}}{2}))$ ▷ TERM score calculation

4.2 TERM Properties

Notice that Equation TERM used in the reputation model gradually increases with early reporting but reduces relatively fast with random reporting when the reports do not match. With this, one can observe that trustworthy reporting benefits the agents over random reporting. Hence, agents cannot manipulate their TERM score.

We consider a *collusive* strategy wherein all agents collude to submit the same report, i.e., single report strategy. We prove that TERM score is resistant to such a strategy. The following lemmas formally present these observations.

Lemma 4.1. *TERM score increases with early reporting.*

Proof. We have seen that the TERM score directly increases with increase in round-scores $\phi_i = \frac{1}{freq(x_i).t_i}$ where x_i, t_i are the report submitted and time taken by agent a_i. It is evident that the round-score increases with early reporting. Let t' be the time taken by the agent a_i to solve the task.

$$\frac{1}{freq(x_i).t'} \geq \frac{1}{freq(x_i).t_i} \quad \forall t_i > t'$$

Hence, TERM score increases with early reporting. □

Lemma 4.2. *TERM is resistant to single report strategy.*

Proof. We observe that in TERM, the round-score $\phi_i^j = \frac{\mathbb{I}_{y_i=y_p}}{f(y_i)t_i}$ directly depends on the report submitted y_i and time taken t_i by the agent a_i.

Consider a single report strategy where all the agents report the same answer $y_i = x$. In this case, $f(y_i) = 1$ and the expected round-score of the agent a_i is,

$$CS : \phi_i^j = \frac{1}{t_i}$$

Suppose, out of m agents in a round, l agents have evaluation x, and others have evaluation different from x. Moreover, all agents are trustworthy.

The expected round-score of agent a_i who report x in a non-colluding (trustworthy) strategy is,

$$TS : \phi_i^j = \frac{l}{m} \times \frac{m}{l \times t_i} = \frac{1}{t_i}$$

We see that the expected round-score in colluding strategy, CS is equal to a trustworthy strategy, TS. Therefore, any rational agent prefers to choose a trustworthy strategy, as it does not benefit from single report strategy. Thus, we claim that TERM is resistant to single report strategy. □

4.2.1 TERM Properties under RPTSC Reward Scheme

Lemma 4.3. TERM *scores are high for truthful reporting when all the other agents choose trustworthy strategy under RPTSC reward scheme.*

Proof. To prove the lemma, we show that TERM produces high scores for reporting truth after exerting efforts, considering that all other agents are trustworthy. We have seen that, TERM score obtained by agent a_i is $\Omega_i^j = G(\psi_i^j)$ in round r_j. Trivially, TERM score increases with an increase in cumulative-score, which aggregates all the normalised round-scores. Hence, an increase in round-score increases the TERM score.

Assuming two agents with identical round-scores in previous rounds, a difference in round-score of the present round will show a difference in their TERM score. From Equation 4.1, we calculate the round-score as $\phi_i^j = \frac{\mathbb{I}_{y_i=y_p}}{freq(y_i)t_i}$. Here, $freq(y_i)$ is the frequency function of y_i, calculated as the ratio of the number of reports $(b+1)$ that match with report y_i to the total number of sampled reports (n). That is, $freq(y_i) = \frac{num(y_i)}{\sum_{y \in \mathcal{X}} num(y)} = \frac{b+1}{n}$. Further, y_p is the random report sampled from the same task. We have seen that a trustworthy agent's strategy is to report true evaluation (x_i) after exerting efforts (e_H). Round-score of an agent who does not report truth (i.e., report $y_i \neq x_i$) is,

$$
\begin{aligned}
&\frac{P_{p|i}(y_i|x_i)}{t_i} \sum_{b=0}^{n-1} \binom{n-1}{b} P_p(y_i)^b (1 - P_p(y_i))^{n-b-1} \frac{n}{b+1} \\
&= \frac{P_{p|i}(y_i|x_i)}{t_i} \sum_{b=0}^{n-1} \binom{n}{b+1} P_p(y_i)^b (1 - P_p(y_i))^{n-b-1} \\
&= \frac{P_{p|i}(y_i|x_i)}{P_p(y_i)t_i} \sum_{b=1}^{n} \binom{n}{b} P_p(y_i)^b (1 - P_p(y_i))^{n-b} \\
&= \frac{P_{p|i}(y_i|x_i)}{P_p(y_i)t_i} (1 - (1 - P_p(y_i))^n) \\
&\leq \frac{P_{p|i}(x_i|x_i)}{P_p(x_i)t_i} (1 - (1 - P_p(x_i))^n) \quad \text{(From, Equation 2.1 and Definition 2.11)}
\end{aligned}
$$

From the first inequality, we observe that the RHS is round-score an agent when it reports truth (i.e., its evaluation x_i at time t_i). From the inequality, we see that round-score is high for truthful reporting. Hence, TERM incentivizes early as well as truthful reporting. $\square$

In this chapter, we built a game theoretically sound reputation model using the Gompertz function. Next, we analyse the properties of our framework, REFORM, having RPTSC's reward scheme and TERM as a reputation model.

Chapter 5

REFORM with RPTSC Reward Scheme and TERM

In this chapter, we demonstrate the significance of our framework REFORM (Algorithm 16) having RPTSC as reward scheme and TERM as reputation model. Algorithm 18 presents REFORM framework with RPTSC and TERM. We select RPTSC as the base PBM in REFORM because it is the state-of-the-art mechanism for crowdsourcing without requiring any ground truth or any prior on the answers. We deploy TERM (Algorithm 17) as the reputation model for temporal setting.

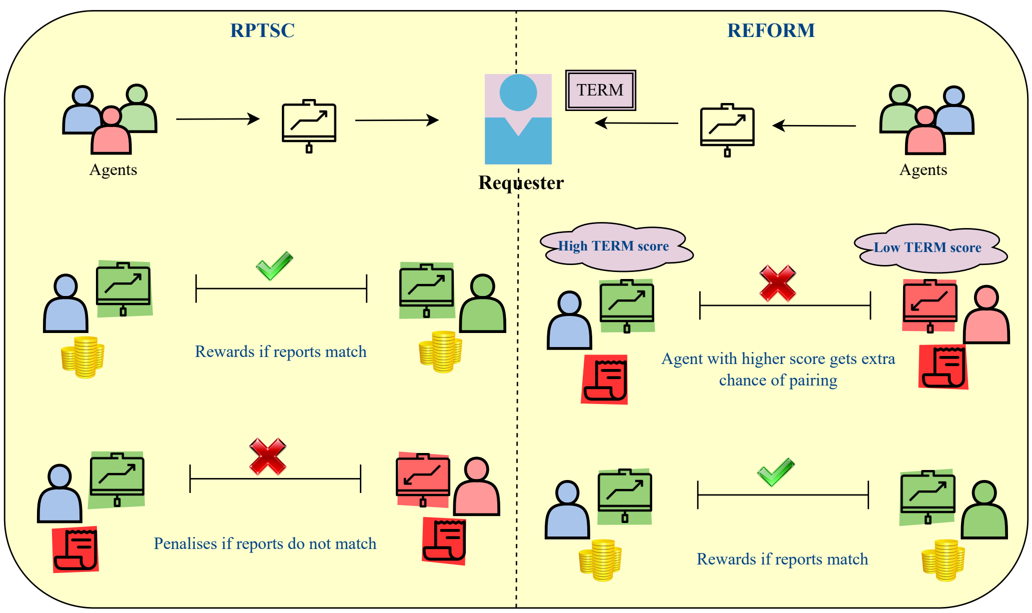

Figure 5.1: REFORM with RPTSC vs RPTSC

5.1 REFORM: An Illustration

We compare the agent rewards in RPTSC (Algorithm 11) with REFORM (Algorithm 18). As shown in Figure 5.1, consider three agents b, g, r (coloured in blue, green and red) that submit their reports y_b, y_g, y_r respectively to the requester of the crowdsourcing system. Let us assume that agents b and g are trustworthy and agent r is malicious. Now, we compute the reward of a trustworthy agent b in RPTSC and REFORM.

RPTSC Reward

In the RPTSC mechanism, to reward agent b, its report is evaluated against a randomly chosen peer. Let us analyse two cases to compute agent b's reward.
Agent b is paired with agent g: Consider the case where agent g is the randomly chosen peer. As both agents, b and g adopt the trustworthy strategy. They exert efforts and report their evaluation truthfully. For the same task, their reports match, and agent b receives the reward it deserves.
Agent b is paired with agent r: Suppose agent b gets agent r as its peer in random matching In this case, b's report does not match its peer. Hence it obtains a penalty.

In RPTSC, it is unfair for agent b as it obtained a penalty due to its report getting evaluated with a malicious agent r.

REFORM Reward

Now, we analyse the reward obtained by agent b in REFORM framework having RPTSC as base PBM. Say, agent b is paired with malicious agent r, and their reports do not match. From Algorithm 18 (Line 11), observe that agent reputation scores are calculated using TERM, and REFORM provides agents with higher reputation additional chances of pairing if their reports do not match. TERM scores are high for trustworthy agents (Lemma 4.3). Thus, TERM score for agent b is higher than TERM score of malicious agent r. Since agent b is paired with a less reputed agent, it gets an additional chance ($k > 1$) to get its report evaluated. In the second chance, say agent b got paired with agent g. As both are trustworthy, their reports match. Agent b is rewarded appropriately. REFORM provides chances to a reputed agent until it exhausts k additional chances. With these extra chances, agents are rescued from unfair penalties in REFORM.

Algorithm 18 REFORM with RPTSC and TERM

Agent i submits a report y_i for an assigned task τ in round r_j after time t_i

Input: $TERM(\cdot)$ TERM, $k \in \mathbb{Z}^+$, y_i

Output: $R_i(y_i, \cdot)$

Initialisation: $l = 0$

Consider $n - 1$ tasks other than τ

Randomly sample $n - 1$ reports from each task

while $l < k$ **do**

    Randomly choose peer report y_p from the same task τ.

    Calculate the frequency function of report y_i from the sampled reports, as $freq(y_i) = \frac{num(y_i)}{\sum_{y \in \mathcal{X}} num(y)}$

    if $l = 1$ **then**

        $\Omega_{i,j} = TERM(y_i, t_i)$

    end if

    $l = l + 1$

    if $y_i = y_p$ **then**

        $R_i(y_i, t_i) = \alpha \left(\frac{1}{freq(y_i)} - 1 \right)$

    else

        if $\Omega_{i,j} \leq \Omega_{p,j}$ **or** $l = k$ **then**

            if $freq(y_i) \neq 0$ **then**

                $R_i(y_i, t_i) = -\alpha$

            else $R_i(y_i, t_i) = 0$

            end if

            end if

    end if

    end while

Observe that using REFORM framework, the chances of an agent with a trustworthy strategy getting penalised are low. Therefore, REFORM assures better fairness to agents compared to any other PBM while preserving the game-theoretic properties of the adopted PBM. We now theoretically prove these observations in the next section.

5.2 REFORM: Theoretical Analysis

In this section, we game-theoretically analyse REFORM by employing RPTSC's reward scheme. The significance of REFORM is highlighted by the fact that despite the theoretical guarantees for *strategy-proofness* and *improvement in fairness*, our analysis does not require any further assumptions than those presented for RPTSC.

However, as we work in temporal settings with reputation scores, we assume that agents have private beliefs about other agents' reputation scores in the system. We denote agent beliefs regarding reputation scores with T.

Agents Beliefs about Reputations (T)

As REFORM comprises a reputation model, we recognise an agent i's belief about the reputation score of an agent p with T_p. We assume that agents' belief about reputation scores do not change with a particular peer as they are randomly paired. That is, we assume $T(\cdot)$ is *identical* and *symmetric*. Mathematically, $T_p = T_q \; \forall p, q \in \mathcal{A}$.

Having, Ω_i, Ω_p be the reputation scores (TERM scores, in this case) of agents i and p. We define the distribution $T_p = Pr(\Omega_i \geq \Omega_p)$ as the probability with which agent p's reputation (Ω_p) is less than i's (Ω_i).

To simplify our notations, we use the following hereafter:

Notation.

$$p_{y_i} = P_p(y_i),\;\; p'_{y_i} = P_{p|i}(y_i|x_i),\;\; p_{x_i} = P_p(x_i),$$
$$p'_{x_i} = P_{p|i}(x_i|x_i),\;\; q_{y_i} = Q_p(y_i),\;\; q'_{y_i} = Q_{p|i}(y_i|x_i),$$
$$q_{x_i} = Q_p(x_i),\;\; q'_{x_i} := Q_{p|i}(x_i|x_i),\;\; r = T_p(\Omega_i, \Omega_p).$$

We begin our analysis by deriving the expected reward for this setting in Lemma 5.1. Observe that the analysis is non-trivial due to the iterative nature of the framework.

Lemma 5.1. *In REFORM with RPTSC, the expected reward of an agent i with evaluation x_i and report y_i, when $k = 2$ is,*

$$\mathbb{E}[R_i(y_i); k = 2] = \begin{cases} \alpha \left[\frac{q'_{y_i}}{q_{y_i}} - 1 + r(1 - q'_{y_i})\frac{q'_{y_i}}{q_{y_i}}\right]\left[1 - (1 - q_{y_i})^{n-1}\right], & \text{if } q_{y_i} > 0 \\ 0, & \text{otherwise} \end{cases}$$

Here, $n \geq 2$ is the number of tasks.

Proof. Observe that when $q_{y_i} = 0$ (i.e., the probability with which agent i's peer reports y_i is 0), the expected reward of agent i is 0. Now, consider agent i with $q_{y_i} > 0$ and evaluation x_i, reports y_i. From Proposition 2.3, the expected reward of an agent i for reporting y_i in RPTSC is E'.

Next we calculate the expected reward $\mathbb{E}[R_i(y_i); k = 2]$ as follows. For this, let the reputation score of an agent i in round r_j be Ω_i. Let agent p with report y_p and reputation score Ω_p be the random peer against whom i is evaluated. As seen before, agent i's belief regarding the reputation scores of the any peer is the same, i.e., $\forall p \in \mathcal{A}, r = T_p(\Omega_i, \Omega_p)$.

From Framework 16, if the reputation score Ω_i of agent i is less than Ω_p, in the first chance of pairing, then agent i does not get an additional chance to pair. In this case, the expected reward is the same as RPTSC expected reward, i.e., E'.

However, if i's reputation score is higher than that of its peer's, then since $k = 2$, it gets another chance to pair. In this case, we have: (i) if $y_i = y_p$ the reward is equal to optimal reward, i.e., M' (Equation 2.4); and (ii) if $y_i \neq y_p$ the expected reward is equal to E', as it receives an additional chance. Formally, we have,

$$\begin{aligned}
\mathbb{E}[R_i(\cdot)] &= Pr(\Omega_i < \Omega_p)E' + Pr(\Omega_i > \Omega_p)\Big(Pr(y_i = y_p|x_i)M' + Pr(y_i \neq y_p|x_i)E'\Big) \\
&= (1 - r)E' + r\big(q'_{y_i}M' + (1 - q'_{y_i})E'\big) \\
&= \Big(E' + rq'_{y_i}\left(M' - E'\right)\Big) \\
&= \alpha\bigg(\left(\frac{q'_{y_i}}{q_{y_i}} - 1\right) + rq'_{y_i}\frac{(1 - q'_{y_i})}{q_{y_i}}\bigg)\left(1 - (1 - q_{y_i})^{n-1}\right).
\end{aligned}$$

This completes the proof of the lemma. $\square$

From Proposition 2.3 and Lemma 5.1, we see that the expected reward in REFORM with RPTSC is greater than RPTSC, highlighting REFORM's efficiency. Naturally, the expected reward also *increases* with an increase in k.

Corollary 5.1. *In REFORM with RPTSC, the expected reward increases with an increase in additional chances, k.*

Proof. Similar to the proof given for Lemma 5.1, the expected reward of an agent i with evaluation x_i and report y_i in REFORM with RPTSC is,

$$
\begin{aligned}
&\mathbb{E}[R_i(y_i); k] \\
&= (1-r)E' + r\Big(q'_{y_i}M' + (1-q'_{y_i})\Big((1-r)E' + \ldots r\big(q'_{y_i}M' + (1-q'_{y_i})E'\big)\Big)\Big) \\
&= rq'_{y_i}M'\Big(1 + (r - rq'_{y_i}) + \ldots + (r - rq'_{y_i})^{k-1}\Big) + E'(r - rq'_{y_i})^{k-1} \\
&\qquad + E'(1-r)\Big(1 + (r - rq'_{y_i}) + \ldots + (r - rq'_{y_i})^{k-2}\Big) \\
&= rq'_{y_i}M'\sum_{i=1}^{k}\left(r - rq'_{y_i}\right)^{i-1} + E'\Big((r - rq'_{y_i})^{k-1} + (1-r)\sum_{i=2}^{k}(r - rq'_{y_i})^{i-2}\Big)
\end{aligned}
$$

From the above, we see that every term is positive, and with an increase in k expected reward increases. This proves the lemma. $\square$

5.2.1 Game Theoretic Guarantees

We now prove that adopting trustworthy strategy is strict NIC in REFORM with RPTSC. Observe that to incentivize high efforts, agents' expected utility must be strictly greater than random reporting (in which agents exert low efforts). Considering all the agents follow trustworthy strategy (i.e., $q'_{y_i} = p'_{y_i}, q_{y_i} = p_{y_i}$ in Lemma 5.1), agent i's expected reward before its evaluation is,

$$
\overline{Ref}_i(\alpha) = \alpha\mathbb{E}_{y_i \in \mathcal{X}}\left[\left(\frac{p'_{y_i}}{p_{y_i}} - 1 + rp'_{y_i}\frac{(1-p'_{y_i})}{p_{y_i}}\right)\left(1 - (1-p_{y_i})^{n-1}\right)\right] \tag{5.1}
$$

With this, in Lemma 5.2, we first give the expected reward for an agent following the random strategy.

Lemma 5.2. *In REFORM with RPTSC, the expected reward (E_{ra}) of random strategy agent with report y_i when all the other agents choose trustworthy strategy is,*

$$E_{ra} = \alpha r(1 - p_{y_i}) \left(1 - (1 - p_{y_i})^{n-1}\right)$$

Proof. Note that a random agent does not exert efforts for a task, i.e., its evaluation for the task is $\varnothing$. Therefore, putting $p'_{y_i} = p_{y_i}$ in Equation 5.1 gives the expected reward of a random agent, when all agents are trustworthy.

$$\begin{aligned} E_{ra} &= \alpha \left(\left(\frac{p_{y_i}}{p_{y_i}} - 1 \right) + r(1 - p_{y_i}) \frac{p_{y_i}}{p_{y_i}} \right) \left(1 - (1 - p_{y_i})^{n-1}\right) \\ &= \alpha r(1 - p_{y_i}) \left(1 - (1 - p_{y_i})^{n-1}\right) \end{aligned}$$

The equality proves the lemma.

□

Next, Lemma 5.3 proves that REFORM with RPTSC incentivizes agents to exert high efforts by showing that the expected utility for exerting efforts is strictly greater than random reporting.

Lemma 5.3. *In REFORM with RPTSC, an agent is incentivized to exert high efforts given all the other agents choose the trustworthy strategy.*

Proof. We have $\overline{Ref}_i(\alpha) - c(e_H)$ (Equation 5.1) as the expected utility of an agent i for exerting efforts before evaluation. And, the expected utility of a random agent is E_{ra} $-c(e_L)$ (Lemma 5.2). From Proposition 2.2, we show that $\overline{Ref}_i(\alpha) - c(e_H) \geq E_{ra} - c(e_L)$ proving the Lemma. We now give the proof in detail.

Before evaluation of the task, agent i's expected utility for investing high efforts is $\overline{Ref}_i(\alpha) - c(e_H)$ and its expected utility when it reports randomly is $E_{ra} - c(e_L)$. We show that $\overline{Ref}_i(\alpha) - c(e_H) > E_{ra} - c(e_L)$, to prove that REFORM incentivizes high efforts.

From Lemma 5.2, we have the expected reward of random agent:

$$\begin{aligned}E_{ra} &= r\alpha\mathbb{E}_{y_i\in\mathcal{X}}\left[(1-p_{y_i})\left(1-(1-p_{y_i})^{n-1}\right)\right]\\ &= \alpha\sum_{y_i\in\mathcal{X}}\left[rp_{y_i}(1-p_{y_i})\left(1-(1-p_{y_i})^{n-1}\right)\right]\\ &< \alpha\end{aligned}$$

And from A (Proposition 2.2), we have $\overline{R}_i(\alpha) > c(e_H) - c(e_L)$. That is,

$$\begin{aligned}\overline{Ref}_i(\alpha) - C(e_H) = \alpha\mathbb{E}_{y_i\in\mathcal{X}}&\left[\left(\frac{p'_{y_i}}{p_{y_i}} - 1 + rp'_{y_i}\frac{(1-p'_{y_i})}{p_{y_i}}\right)\left(1-(1-p_{y_i})^{n-1}\right)\right]\\ &\overline{R}_i(\alpha) + \alpha\mathbb{E}_{y_i\in\mathcal{X}}\left[\left(rp'_{y_i}\frac{(1-p'_{y_i})}{p_{y_i}}\right)\left(1-(1-p_{y_i})^{n-1}\right)\right]\\ &> (C(e_H) + \alpha\mathbb{E}_{y_i\in\mathcal{X}}\left[\left(rp'_{y_i}\frac{(1-p'_{y_i})}{p_{y_i}}\right)\left(1-(1-p_{y_i})^{n-1}\right)\right]) - C(e_L) - C(e_H)\\ &> \alpha - C(e_L)\\ &> E_{ra} - C(e_L)\end{aligned}$$

The expected utility of an agent before evaluation for exerting efforts is strictly greater than the expected utility from random reporting. Thus, a random agent is incentivized to exert high efforts in REFORM. □

From Lemma 5.3, we notice that random strategy is at a disadvantage. TERM scores for agents who report randomly drop significantly, implying they do not get additional chances. Hence, even when an agent i reports randomly at a time $t_i \to 0$, it does not get greater rewards.

Having shown that exerting efforts is incentivized, Lemma 5.4 shows that it is beneficial for an agent to follow trustworthy strategy, given all the other agents follow the same.

Lemma 5.4. *In REFORM with RPTSC, an agent i is incentivized to report truthfully when all the other agents choose trustworthy strategy under Assumption* B(n)*.*

Proof. Consider agent i with evaluation x_i and report y_i submitted. We assume that all other agents are trustworthy. For the strategy profile where all the agents are trustworthy, $q' = p'; q = p$.

From Lemma 5.1, the expected reward of agent i for reporting y_i, in REFORM with RPTSC when $k=2$ is $\mathbb{E}[R_i(y_i); k=2]$.

We prove that the expected reward of a strategic agent, i is less when it reports any value other than its evaluation x_i. For this, we start with the assumption that the reward for reporting the truth is more than the reward for reporting non-truth and arrive at a noticeably obvious result.

$$
\begin{aligned}
&\implies \mathbb{E}[R_i(x_i); k=2] > \mathbb{E}[R_i(y_i); k=2] \\
&\implies \alpha\left[\left(\frac{p'_{x_i}}{p_{x_i}}-1\right)+r(1-p'_{y_i})\frac{p'_{x_i}}{p_{x_i}}\right]\left(1-(1-p_{x_i})^{n-1}\right) > \alpha\left[\left(\frac{p'_{y_i}}{p_{y_i}}-1\right)+r(1-p'_{y_i})\frac{p'_{y_i}}{p_{y_i}}\right]\left(1-(1-p_{y_i})^{n-1}\right) \\
&\implies \left(\frac{p'_{x_i}}{p_{x_i}}-1\right)\left(1-(1-p_{x_i})^{n-1}\right)-\left(\frac{p'_{y_i}}{p_{y_i}}-1\right)\left(1-(1-p_{y_i})^{n-1}\right) \\
&\qquad > r\left[\frac{(1-p'_{y_i})p'_{y_i}}{p_{y_i}}\left(1-(1-p_{y_i})^{n-1}\right)-\frac{p'_{x_i}(1-p'_{x_i})}{p_{x_i}}\left(1-(1-p_{x_i})^{n-1}\right)\right] \\
&\implies \left(\frac{p'_{x_i}}{p_{x_i}}-1\right)\left[\left(1-(1-p_{x_i})^{n-1}\right)-\frac{p_{x_i}}{p_{y_i}}\left(1-(1-p_{y_i})^{n-1}\right)\right] \\
&\qquad > r\left[\frac{(1-p'_{y_i})p'_{y_i}}{p_{y_i}}\left(1-(1-p_{y_i})^{n-1}\right)-\frac{p'_{x_i}(1-p'_{x_i})}{p_{x_i}}\left(1-(1-p_{x_i})^{n-1}\right)\right] \text{(Equation 2.1 \& B(n))} \\
&\implies \left(p'_{x_i}-p_{x_i}\right)\left[p_{y_i}\left(1-(1-p_{x_i})^{n-1}\right)-p_{x_i}\left(1-(1-p_{y_i})^{n-1}\right)\right] \\
&\qquad > r\left[(1-p'_{y_i})p'_{y_i}p_{x_i}\left(1-(1-p_{y_i})^{n-1}\right)-p'_{x_i}p_{y_i}(1-p'_{x_i})\left(1-(1-p_{x_i})^{n-1}\right)\right] \\
&\implies \left(p'_{x_i}-p_{x_i}\right)\left[p_{y_i}\left(1-(1-p_{x_i})^{n-1}\right)-p_{x_i}\left(1-(1-p_{x_i})^{n-1}\right)\right] \\
&\qquad > r\left[p_{x_i}\left(1-(1-p_{y_i})^{n-1}\right)-p'_{x_i}p_{y_i}(1-p'_{x_i})\left(1-(1-p_{x_i})^{n-1}\right)\right] \\
&\implies -\frac{p_{x_i}\left(1-(1-p_{y_i})^{n-1}\right)-p'_{x_i}p_{y_i}(1-p'_{x_i})\left(1-(1-p_{x_i})^{n-1}\right)}{p_{x_i}\left(1-(1-p_{y_i})^{n-1}\right)-p_{y_i}\left(1-(1-p_{x_i})^{n-1}\right)} \\
&\qquad < \frac{(p'_{x_i}-p_{x_i})}{r} \text{(Since, numerator > denominator)}
\end{aligned}
$$

$$
\implies \frac{(p'_{x_i}-p_{x_i})}{r} > -1 \implies \left(p'_{x_i}-p_{x_i}\right) > -r
$$

We know that $p'_{x_i} - p_{x_i} > 0$ (From, self-predicting condition) and $-r \leq 0$. Thus, the above condition is true, implying the assumption made is true. That is, the expected utility for reporting the truth is strictly more than strategic reporting. Hence, proving the lemma. □

From Lemmas 5.3 and 5.4, we see that an agent is incentivized to exert efforts and report truthfully when all the other agents choose the trustworthy strategy. Thus, REFORM with RPTSC incentivizes high efforts and truthful reporting. More formally,

Theorem 5.1. *REFORM with RPTSC is strict Nash Incentive Compatible (NIC).*

Discussion

RPTSC reward is resistant to single report strategy [29, Section 4.4]. That is, all the agents collude to report the same answer is discouraged by the reward structure of RPTSC. One can observe that the reward in Equation 2.2 is zero under this strategy. Hence, for any appropriately chosen scalar constant α, REFORM with RPTSC reward is also resistant to single report strategy.

We next show that REFORM with RPTSC reward guarantees significantly better fairness when compared to RPTSC.

5.2.2 Fairness Guarantees

In addition to the above incentive properties, we now show that REFORM significantly improves fairness compared to RPTSC. Consider the following propositions based on our novel notion of γ-fairness (Definition 3.1).

Proposition 5.1. *For any $i \in \mathcal{A}$, REFORM with RPTSC is γ-fair with $\gamma = \sum_{x_i \in \mathcal{X}} q_{x_i} \left(\frac{(1-q'_{x_i})(1-rq'_{x_i})}{1-q_{x_i}} \right)$*

Proof. From Lemma 5.1, the expected reward of REFORM is $\mathbb{E}[R_i(x_i); k = 2]$. And, the optimal reward in REFORM with RPTSC is same as M'. Now,

$$\mathbb{E}_{x_i \in \mathcal{X}} \left[\frac{M' - \mathbb{E}[R_i(x_i); k = 2]}{M'} \right] = \mathbb{E}_{x_i \in \mathcal{X}} \left[\frac{(1 - q'_{x_i}) - r(1 - q'_{x_i})q'_{x_i}}{1 - q_{x_i}} \right]$$
$$= \alpha \sum_{x_i \in \mathcal{X}} q_{x_i} \left(\frac{(1 - q'_{x_i})(1 - rq'_{x_i})}{1 - q_{x_i}} \right)$$

From Definition 3.1, REFORM is $\alpha \sum_{x_i \in \mathcal{X}} q_{x_i} \left(\frac{(1-q'_{x_i})(1-rq'_{x_i})}{1-q_{x_i}}\right)$-fair. $\square$

Observe that γ in REFORM with RPTSC (Proposition 5.1) is *lesser* than that in RPTSC (Proposition 3.1). This implies that REFORM guarantees an expected reward which is closer to the optimal reward. Thus, REFORM with RPTSC is fairer compared to RPTSC. Moreover, note that RPTSC's expected reward is greater than other PBMs (for the same setting) such as PTS [27]. That is, $\gamma_{PTS} > \gamma_{RPTSC} > \gamma_{REFORM}$, implying REFORM is fairer than RPTSC and PTS, both.

We next show that the iterative nature of REFORM using reputation models is qualitatively fair.

Theorem 5.2. *REFORM with RPTSC is qualitatively fair.*

Proof. REFORM provides additional chances of pairing to the agents with high reputation scores. This reduces their chances of getting penalised from unfair pairing, leading to increased expected rewards. Thus, expected rewards are proportional to reputation scores, satisfying qualitative fairness. We show this mathematically as follows:

Expected reward of an agent i for reporting y_i is

$$
\begin{aligned}
&\mathbb{E}[R_i(x_i); k = 2] \\
&= \alpha \left[\left(\frac{q'_{y_i}}{q_{y_i}} - 1 \right) + r(1 - q'_{y_i}) \frac{q'_{y_i}}{q_{y_i}} \right] \left(1 - (1 - q_{y_i})^{n-1}\right) \\
&= E' + \alpha r (1 - q'_{y_i}) \frac{q'_{y_i}}{q_{y_i}} \left(1 - (1 - q_{y_i})^{n-1}\right)
\end{aligned}
$$

Where r is the probability with which the peer's reputation score is less than that of agent i's score.

Consider two agents a, b who reported same answers, with reputation scores Ω_a, Ω_b (where, $\Omega_a < \Omega_b$) respectively. The beliefs about other agents having lesser reputation scores is given by T_p, where $T_p(\Omega, \Omega_p)$ is the probability with which peer p's score is less than Ω. Since,

$$\Omega_a < \Omega_b \implies T_p(\Omega_a, \Omega_p) < T_p(\Omega_b, \Omega_p)$$

Assuming that both the agents have the same beliefs. The expected reward for the agents a and b are given as,

$$
\begin{aligned}
E_a &= E' + \alpha T_p(\Omega_a, \Omega_p)(1 - q'_{y_i}) \frac{q'_{y_i}}{q_{y_i}} \left(1 - (1 - q_{y_i})^{n-1}\right) \\
E_b &= E' + \alpha T_p(\Omega_b, \Omega_p)(1 - q'_{y_i}) \frac{q'_{y_i}}{q_{y_i}} \left(1 - (1 - q_{y_i})^{n-1}\right) \\
E_a &< E_b
\end{aligned}
$$

We see that the agent's expected reward with a greater reputation is higher than the agent's expected reward with a lower reputation having the same report. Therefore, REFORM with RPTSC is qualitatively fair. □

The above results theoretically prove that REFORM's idea of providing additional chance(s) to agents significantly improves fairness in PBMs. We next validate the same through empirical evaluations.

5.3 REFORM: Experimental Analysis

We now perform experiments to quantitatively validate REFORM's improved fairness and observe the resulting γ values for REFORM and RPTSC.

Setup

For this, we simulate our crowdsourcing setting for 200 rounds. Each round comprises 750 agents that report answers to one of the available 50 tasks. Since REFORM with RPTSC satisfies NIC, we only consider trustworthy and random strategies. As for agents adopting the deceiving strategy, the best possible answer to report is its evaluation since these agents have already exerted effort. We assume that among the 750 agents available for tasks, 60% of them choose a trustworthy strategy, and the remaining 40% choose a random strategy. We believe these numbers reasonably approximate a real-world scenario. However, our theoretical guarantees are not specific to them. To satisfy A (Proposition 2.2), we set α as 10 for both RPTSC and REFORM with RPTSC. Here, our main objective is to analyse the fairness improvement provided by our framework, REFORM, compared to

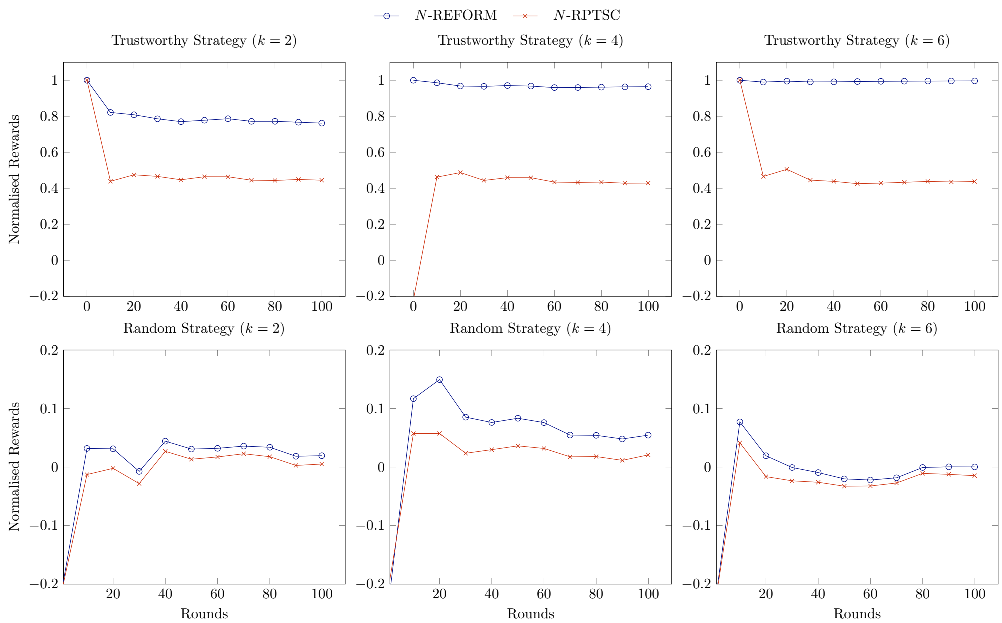

Figure 5.2: Normalised rewards for REFORM with RPTSC vs. RPTSC for distribution 60% and 40%

RPTSC. Thus, we compare the average rewards over 200 rounds for both the strategies in REFORM with RPTSC and standalone RPTSC.

Observations

Figure 5.2 compares the rewards of agents for the two different strategies, (i) trustworthy and (ii) random. In the figure, N-REFORM and N-RPTSC are rewards normalised with their optimal reward for $k = 2, 4$, and 6. Observe that, N-REFORM is sufficiently greater than N-RPTSC when the agents employ trustworthy strategy. Observe that with an increase in k, N-REFORM for trustworthy strategy (Row 1) gets closer to the line $y = 1$, i.e., the rewards tend towards the optimal reward. However, for random strategy, N-REFORM and N-RPTSC are almost the same. Thus, one can note that REFORM guarantees greater rewards than RPTSC for trustworthy strategy; and approximately similar rewards for random. The γ values also quantify the increase in reward; we observe γ to be 0.23 in REFORM with RPTSC and 0.4 in RPTSC, respectively, after 200 rounds. As we have seen in previous chapters, the lesser the γ value greater the fairness. Thus, REFORM with RPTSC is fairer than RPSTC w.r.t. γ-fairness.

5.3.1 Additional Experiments

Varying Fraction of Honest agents

Previously, we have used a setting which assumes that 60% agents choose trustworthy strategy and remaining 40% choose random strategy. Here, we give the plots similar to Figure 5.2 for two other distributions. We consider other distributions with (i) 70% trustworthy behaviour and 30% random behaviour, and (ii) 50% trustworthy behaviour and 50% random behaviour.

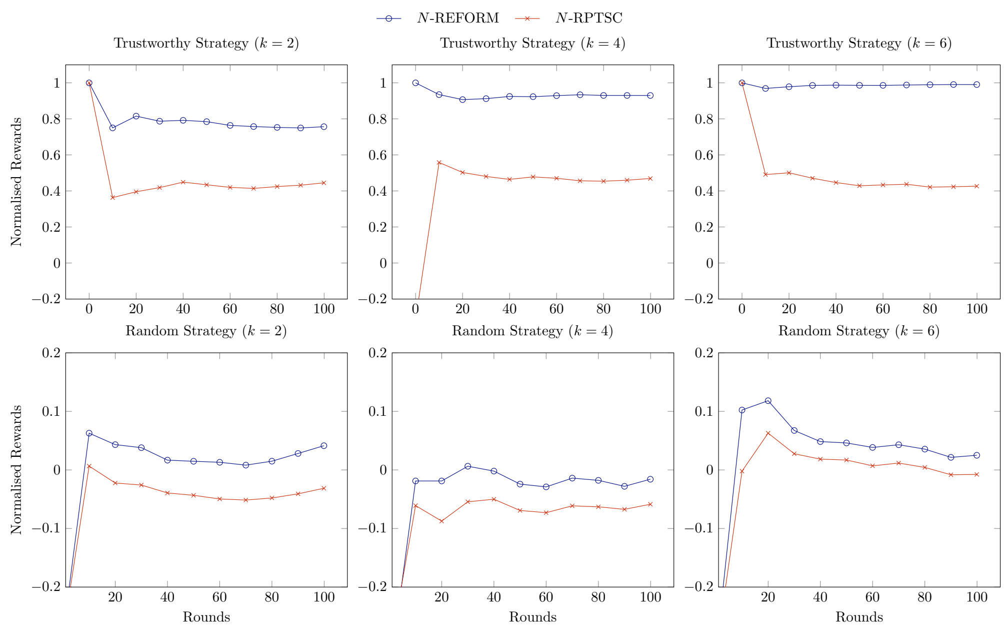

Figure 5.3: Normalised rewards for REFORM with RPTSC vs. RPTSC for distribution 70% and 30%

From Figure 5.3 and 5.4, we observe that irrespective of the distribution of agents adopting different strategies REFORM with RPTSC provided better fairness than RPTSC. That is, the REFORM framework provided rewards closer to the optimal reward for agents

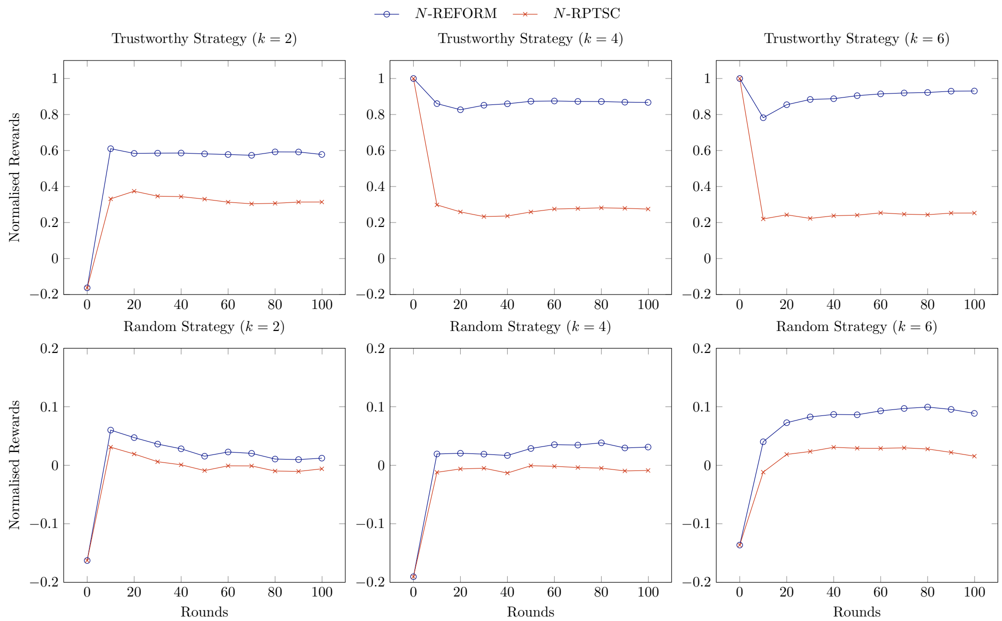

Figure 5.4: Normalised rewards for REFORM with RPTSC vs. RPTSC for distribution 50% and 50%

that choose trustworthy strategy and similar or worse rewards to random strategy agents ensuring fairness.

Effect of k on Expected Reward for Honest Agents and Total Budget For The Requester

We studied the effect of offering more chances to reputed agents in the setting where 60% and 40% agents choose trustworthy and random strategies, respectively. We observed that the reward increases with k (Figure 5.5). As k increases, the REFORM rewards tend towards the optimal reward (green line) and almost saturate at $k = 5$. As k reaches 8, the expected rewards for trustworthy agents in REFORM are more than 0.99 times the optimal reward – 73.4 for our settings which is 2.3 times higher than that of RPTSC. Similarly, we observed that the budget required in REFORM with RPTSC compared to RPTSC increases proportionately – twice than RPTSC for even $k = 18$. Figure. 5.6 depicts the ratio of REFORM and RPTSC budget. However, note that the per agent reward for agents adopting a trustworthy strategy in REFORM with RPTSC is the same as that in RPTSC. REFORM framework only avoids the penalty a reputed agent receives from unfair pairings by increasing its expected rewards. As fewer agents are wrongly penalised through the evaluation process in REFORM the budget increases compared to RPTSC.

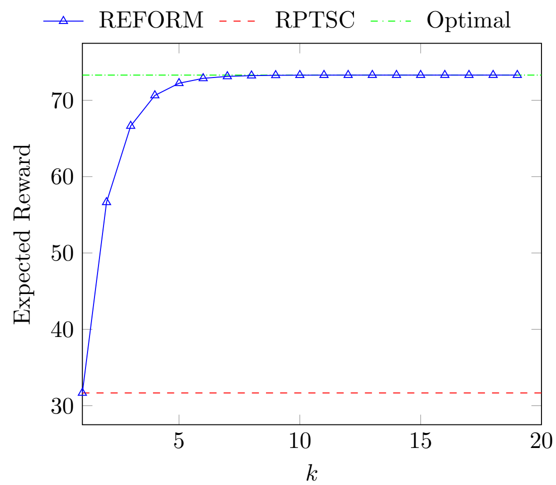

Figure 5.5: Expected reward of a trustworthy agent w.r.t. k increases

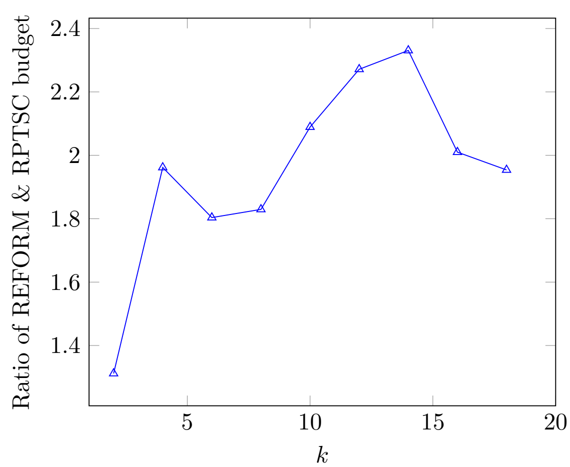

Figure 5.6: Ratio of budget in REFORM and RPTSC w.r.t. k increases

Chapter 6

Conclusion and Future Work

With the advent of technology, relying on the crowd for various tasks is inevitable. Towards this, crowdsourcing systems have gained traction in serving the purpose. These systems need to offer participating agents appropriate incentives for reporting truthful data. The most important class of incentive mechanisms is *Peer Based Mechanisms* where each agent is matched with a random peer to evaluate its report. This thesis focused on data elicitation in crowdsourcing, considering the temporal setting. The existing PBMs have *fairness* issues. Towards this, our primary goal was to design a framework for crowdsourcing that (i) improves fairness in PBMs while incorporating temporal settings and (ii) ensures truthful reporting. We proposed that trustworthy agents (TA) should get *additional chances* of pairing for computing their reward to *minimise the penalties* from unfair pairings. With this approach, we introduced **REFORM** (Algorithm 16), a novel iterative framework that takes the reward scheme of any existing PBMs and reputation model as a plug-in. REFORM uses reputation models to decide which agents get additional chances. We introduced two notions of fairness, (A) *γ-fairness* (Definition 3.1) and (B) *quantitative fairness* (Definition 3.2), to quantify the fairness of a PBM. As we work in temporal settings, it necessitates a manipulation-free reputation model that can incorporate temporal settings. We quantified the trustworthiness of agents in the system by introducing a temporal reputation model, **TERM** (Algorithm 17), and demonstrated that it provides high scores for trustworthy and early reporting (Lemmas 4.1 & 4.3). Using TERM as a reputation model and RPTSC as base PBM (Algorithm 18), we proved that in REFORM, it is a strict Nash equilibrium for trustworthy reporting at the earliest (Theorem 5.1). We have shown that REFORM provides fairer rewards than RPTSC through γ-fairness and is qualitatively fair (Theorems 5.2). Through experiments, we have demonstrated that REFORM's improved fairness comes at a marginal increase in the budget (Section 5.3).

Future Work

As the expected rewards for trustworthy agents rewards are higher in REFORM. In future, from a system designer's perspective, one may further analyse the extra cost a requester incurs due to REFORM's improved fairness. We can also look at providing *dominant strategy incentive compatiability* (DSIC, Definition 2.7) guarantees either by introducing assumptions on agent beliefs or through mechanisms which do not rely on repeated matching. One can also explore the impact of these mechanisms on the fairness of an agent's reward.

Bibliography

[1] J. C. Tang, M. Cebrian, N. A. Giacobe, H.-W. Kim, T. Kim, and D. B. Wickert. Reflecting on the darpa red balloon challenge. *Communications of the ACM*, 54(4):78–85, 2011.

[2] S. Vermicelli, L. Cricelli, and M. Grimaldi. How can crowdsourcing help tackle the covid-19 pandemic? an explorative overview of innovative collaborative practices. *R&D Management*, 51, 11 2020.

[3] D. Chatzopoulos, A. Jain, S. Gujar, B. Faltings, and P. Hui. Towards mobile distributed ledgers. *IEEE Internet of Things Journal*, 2021.

[4] A. Sunyaev. Distributed ledger technology. In *Internet Computing*, pages 265–299. Springer, 2020.

[5] S. Kanaparthy, M. Padala, S. Damle, and S. Gujar. Fair federated learning for heterogeneous data. In *Young Researchers Symposium, CODS-COMAD'22.*, 2022.

[6] M. Padala, S. Damle, and S. Gujar. Federated learning meets fairness and differential privacy. Proceedings of the 28th International Conference on Neural Information Processing of the Asia-Pacific Neural Network Society 2021 (ICONIP '21), 2021.

[7] P. Whitla. Crowdsourcing and its application in marketing activities. *Contemporary Management Research*, 5(1), 2009.

[8] A. Ghezzi, D. Gabelloni, A. Martini, and A. Natalicchio. Crowdsourcing: A review and suggestions for future research. *International Journal of Management Reviews (IJMR'17)*, 20, 01 2017.

[9] J. Wang, J. Tang, D. Yang, E. Wang, and G. Xue. Quality-aware and fine-grained incentive mechanisms for mobile crowdsensing. In *IEEE International Conference on Distributed Computing Systems (ICDCS'16)*, pages 354–363, 2016.

[10] Y. Tang, S. Tasnim, N. Pissinou, S. S. Iyengar, and A. Shahid. Reputation-aware data fusion and malicious participant detection in mobile crowdsensing. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4820–4828, 2018.

[11] Crowd sensing. https://encyclopedia.pub/entry/1215.

[12] Crowd funding. https://en.wikipedia.org/wiki/Crowdfunding.

[13] Crowd voting. https://carlsonschool.umn.edu/sites/carlsonschool.umn.edu/files/2020-07/ChenXuLiu-CrowdVoting.pdf.

[14] Wikipedia:size of wikipedia. https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia.

[15] World's air pollution: Real-time air quality index. https://waqi.info/.

[16] Amazon mechanical turk. https://artsandculture.google.com/entity/amazon-mechanical-turk/m08q6d3?hl=en.

[17] How crowdsourcing is changing the waze we drive. https://digital.hbs.edu/platform-rctom/submission/how-crowdsourcing-is-changing-the-waze-we-drive/.

[18] How lego used crowdsourcing to achieve 21st century success. https://crowdsourcingweek.com/blog/lego-success-through-crowdsourcing/.

[19] Go madbury and invent a new cadbury dairy milk! https://in.mondelezinternational.com/newsroom/gomadburyforcadbury.

[20] Greybeard and K. Martin. The myth of crowdsourcing at wikipedia.

[21] Wikipedia:disruptive sanctions. https://en.wikipedia.org/wiki/Wikipedia:Disruptive_sanctions.

[22] K. Emery, T. Sallee, and Q. Han. Worker selection for reliably crowdsourcing location-dependent tasks. In *International Conference on Mobile Computing, Applications and Services (MobiCASE'15)*, 2015.

[23] B. Ye, Y. Wang, and L. Liu. Crowd trust: A context-aware trust model for worker selection in crowdsourcing environments. In *2015 IEEE International Conference on Web Services (ICWS'15)*, 2015.

[24] S. Jagabathula, L. Subramanian, and A. Venkataraman. Reputation-based worker filtering in crowdsourcing. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14)*, page 2492–2500, 2014.

[25] H. Li and Q. Liu. Cheaper and better: Selecting good workers for crowdsourcing, 2015.

[26] B. Faltings, G. Radanovic, R. Brachman, and P. Stone. *Game Theory for Data Science: Eliciting Truthful Information.* 2017.

[27] B. Faltings, R. Jurca, P. Pu, and B. D. Tran. Incentives to counter bias in human computation. In *AAAI Conference on Human Computation and Crowdsourcing (HCOMP'14)*, 2014.

[28] D. Prelec. A bayesian truth serum for subjective data. *Science (New York, N.Y.)*, 306:462–6, 11 2004.

[29] G. Radanovic, B. Faltings, and R. Jurca. Incentives for effort in crowdsourcing using the peer truth serum. *ACM Trans. Intell. Syst. Technol.*, 7(4), Mar. 2016.

[30] S. Alkoby, D. Sarne, E. Segal-Halevi, and T. Sharbaf. Eliciting truthful unverifiable information. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'18)*, page 1850–1852, 2018.

[31] N. Goel and B. Faltings. Deep bayesian trust: A dominant and fair incentive mechanism for crowd. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:1996–2003, 07 2019.

[32] D. Oleson, A. Sorokin, G. Laughlin, V. Hester, J. Le, and L. Biewald. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. In *Proceedings of the 11th AAAI Conference on Human Computation (AAAIWS'11)*, 2011.

[33] M. H. Moti, D. Chatzopoulos, P. Hui, and S. Gujar. Farm: Fair reward mechanism for information aggregation in spontaneous localized settings. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI'19)*, 7 2019.

[34] C. Qiu, A. Squicciarini, and B. Hanrahan. Incentivizing distributive fairness for crowdsourcing workers. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'19)*, pages 404–412, 2019.

[35] N. Goel and B. Faltings. Crowdsourcing with fairness, diversity and budget constraints. *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society (AIES 2019).*

[36] C. C. Innovating emergency response through crowdsourcing (2017). `https://www.govloop.com/community/blog/innovating-emergency-response-through-crowdsourcing/`.

[37] A. Rehan. Crowdsourcing real-time traveler information systems. 2015.

[38] E. Tan, H. Xia, C. Ji, R. Joshi, and Y. Huang. Designing a mobile crowdsourcing system for campus safety. 2015.

[39] W. Zhang, R. Kim, and S. Subramanian. Crowdsourcing for robust, real-time covid-19 data. *Nature India*, 2020.

[40] C. Li, V. S. Sheng, L. Jiang, and H. Li. Noise filtering to improve data and model quality for crowdsourcing. *Know.-Based Syst.*, 107(C):96–103, sep 2016.

[41] S. Ivanovic, A.-M. Olteanu Raimond, and S. Mustiere. A filtering-based approach for improving crowdsourced gnss traces in a data update context. *ISPRS International Journal of Geo-Information*, 8:380, 08 2019.

[42] G. Kazai and I. Zitouni. Quality management in crowdsourcing using gold judges behavior. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM'16)*, page 267–276, 2016.

[43] D. Garg, Y. Narahari, and S. Gujar. Foundations of mechanism design: A tutorial part 1-key concepts and classical results. *Sadhana*, 33(2):83–130, 2008.

[44] D. Garg, Y. Narahari, and S. Gujar. Foundations of mechanism design: A tutorial part 2-advanced concepts and results. *Sadhana*, 33(2):131–174, 2008.

[45] S. Jain, G. Ghalme, S. Bhat, S. Gujar, and Y. Narahari. A deterministic mab mechanism for crowdsourcing with logarithmic regret and immediate payments. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS'16)*, pages 86–94, 2016.

[46] S. Jain, S. Gujar, S. Bhat, O. Zoeter, and Y. Narahari. A quality assuring, cost optimal multi-armed bandit mechanism for expertsourcing. *Artificial Intelligence*, 254:44–63, 2018.

[47] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui. Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain. In *IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS'18)*, pages 442–450, 2018.

[48] V. Shnayder, A. Agarwal, R. Frongillo, and D. C. Parkes. Informed truthfulness in multi-task peer prediction. In *Proceedings of the 2016 ACM Conference on Economics and Computation (EC'16)*, New York, NY, USA, 2016. Association for Computing Machinery.

[49] N. Lambert and Y. Shoham. Eliciting truthful answers to multiple-choice questions. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC'09)*, page 109–118, New York, NY, USA, 2009. Association for Computing Machinery.

[50] G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

[51] B. Waggoner and Y. Chen. Output agreement mechanisms and common knowledge. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 2(1), Sep. 2014.

[52] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

[53] J. Witkowski and D. C. Parkes. A robust bayesian truth serum for small populations. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI'12)*, page 1492–1498. AAAI Press, 2012.

[54] A. Agarwal, D. Mandal, D. C. Parkes, and N. Shah. Peer prediction with heterogeneous users. In *ACM EC*, page 81–98, 2017.

[55] A. Agarwal, D. Mandal, D. C. Parkes, and N. Shah. Peer prediction with heterogeneous users. *ACM Trans. Econ. Comput.*, 8(1), 2020.

[56] R. Frongillo and J. Witkowski. A geometric method to construct minimal peer prediction mechanisms. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*, page 502–508. AAAI Press, 2016.

[57] N. Goel and B. Faltings. Personalized peer truth serum for eliciting multi-attribute personal data. In *Association for Uncertainty in Artificial Intelligence (UAI'19)*, 2019.

[58] Y. Kong. Dominantly truthful multi-task peer prediction with a constant number of tasks. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 2398–2411. SIAM, 2020.

[59] K. L. Huang, S. S. Kanhere, and W. Hu. Are you contributing trustworthy data? the case for a reputation system in participatory sensing. In *Proceedings of the 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWIM'10)*, page 14–22. Association for Computing Machinery, 2010.

[60] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Trans. Sen. Netw.*, 4(3), jun 2008.

[61] S. Reddy, D. Estrin, and M. Srivastava. Recruitment framework for participatory sensing data collections. *IEEE Pervasive Comput.*, 6030:138–155, 12 2010.

[62] M. Pouryazdan, B. Kantarci, T. Soyata, L. Foschini, and H. Song. Quantifying user reputation scores, data trustworthiness, and user incentives in mobile crowd-sensing. *IEEE Access*, 5:1382–1397, 2017.

[63] T. D. Güneş, L. Tran-Thanh, and T. J. Norman. Identifying vulnerabilities in trust and reputation systems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI'19)*, pages 308–314, 7 2019.

[64] A. Whitby and A. Jøsang. Filtering out unfair ratings in bayesian reputation systems. *The Icfain Journal of Management Research*, 4, 01 2004.

[65] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, 12:183–198, 2006.

[66] W. L. Teacy, M. Luck, A. Rogers, and N. R. Jennings. An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. *Artificial Intelligence*, 193:149–185, 2012.

[67] E. Kamar and E. Horvitz. Incentives for truthful reporting in crowdsourcing. In *Proceedings of the 2012 International Conference on Autonomous Agents & Multiagent Systems (AAMAS'12)*, volume 12, pages 1329–1330. Citeseer, 2012.

[68] Khasraghi, M. Hanieh Javadi, and Shahriar. An innovative crowdsourcing approach for amazon mechanical turk. *International Journal of Computer Applications*, 52:20–13, 08 2012.

[69] J. Le, A. Edmonds, V. Hester, and L. Biewald. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *SIGIR 2010 workshop on crowdsourcing for search evaluation*, volume 2126, pages 22–32, 2010.

[70] M. Allahbakhsh, A. Ignjatovic, B. Benatallah, S.-M.-R. Beheshti, E. Bertino, and N. Foo. Reputation management in crowdsourcing systems. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 664–671, 2012.

[71] R. Yu, R. Liu, X. Wang, and J. Cao. Improving data quality with an accumulated reputation model in participatory sensing systems. *Sensors*, 14(3):5573–5594, 2014.