Empirical Methods in Peer Prediction

Empirical Methods in Peer Prediction

Richard Kim

A Thesis in the Field of Information Technology
for the Degree of Master of Liberal Arts in Extension Studies

Harvard University

May 2016

Abstract

Human computation system, often popularly referred to as crowdsourcing, requires the alignment of the incentives of human participants to report truthfully and an effective mean to deal with noise in the human-generated data. The main objective of this thesis is to introduce a new class of peer prediction mechanisms called *empirical peer prediction mechanisms* that represent an unified approach to resolving the incentive alignment and noisy-data challenges in human computation systems.

In the information elicitation literature, existing peer prediction mechanisms provide theoretical solutions to the incentive alignment problems; however, implementing them in practice has been challenging due to restrictive assumptions. On the other hand, in the machine learning literature, researchers have proposed models and algorithms to estimate the error-rates of workers in human computation systems in an effort to reduce noise in the system; however, these models have largely ignored the incentive problem. While they have developed independently, these two disciplines ultimately share the same goal of improving human computation systems.

In this thesis, I bring together the mechanisms and the algorithms from these two disciplines to introduce three new peer prediction mechanisms - the *Empirical Peer Prediction Method*, the *k-Means Peer Prediction Method*, and the *Empirical Scoring Rule Mechanism*. I empirically demonstrate that these mechanisms align the incentives of the self-interested agents such that their utilities are maximized by reporting their signals truthfully. Moreover, I also show that the three mechanisms are robust against various reporting strategies including collusion.

Acknowledgements

I owe my deepest gratitude to my thesis director, Professor David Parkes. His course, CS 186 Economics and Computation, provided the foundation for many of the core concepts in this thesis. David opened my eye to many new ideas and helped me elevate myself intellectually. I am inspired by his intellect as well as his kindness and humbleness. David is truly an inspirational role model as a scholar and an educator. I could not have asked for a better person to guide me through this process.

I want to thank my thesis advisor, Professor Jeff Parker of the Harvard Extension School. Jeff was tremendously helpful in the early stages of this thesis by helping me hone down my ideas. I am indebted to Jeff and all the Extension School's staff members for providing me with tremendously valuable resources during my tenure at the Extension School.

I also would like to extend my personal gratitude to Pavlos Protopapas, Scientific Program Director of Harvard Institute for Applied Computational Science. Even though I am not a student of Harvard IACS, Pavlos took me under his wing and became one of my most valuable mentors at Harvard. He kindly opened up many resources available only to those affiliated with Harvard IACS. I consider Pavlos my valuable friend and my personal hero.

Finally, I am forever indebted to my wife, Ji Yeon Lee, who has given me unwavering support throughout the last three years while I have been pursuing my master's degree at the Harvard Extension School.

Contents

List of Figures

List of Tables

List of Symbols

δ_{max} . . . . . Parameter for k-Means-Confusion algorithm to determine the outliers

$d : \Theta \times \Theta \to \mathbb{R}$ . . . . . A function that computes distance between two matrices (i.e. Mean Squared Distance) in k-Means-Confusion algorithm to cluster agents in the k-Means Peer Prediction Method

$\mathcal{O}$ . . . . . Set of agents that are determined to be outliers by k-Means-Confusion algorithm, $\{i \in \mathcal{I} | d(\Theta^i | \mathbf{\Theta}_\mu) > \delta_{max}\}$

$\mathcal{T}$ . . . . . Set of M states, $\{1, ..., M\}$

T . . . . . Random Variable that takes on a $t \in \mathcal{T}$

ρ_m . . . . . State prior of $m \in \mathcal{T}$, $P(T = m)$

$\boldsymbol{\rho}$ . . . . . State Priors — probability vector that describes distribution of the states, $\boldsymbol{\rho} = \begin{bmatrix} \rho_1 & \dots & \rho_M \end{bmatrix}$

$\mathcal{I}$ . . . . . Set of I agents, $\{1, ..., I\}$

i . . . . . Agent $i \in \mathcal{I}$

$\{-i\}$ . . . . . Set of agent i's peers, $\{1, ..., i-1, i+1, ..., I\}$

$\mathcal{S}$ . . . . . Set of K signals, {1,...,K}

S_i . . . . . Random Variable that takes on a $s \in \mathcal{S}$ to denote signal observed by Agent i

ϕ_k . . . . . Signal prior of $k \in \mathcal{S}$, $P(S = k)$

$\mathbf{\Phi}$ . . . . . Signal Priors — probability vector that describes distribution of the signals, $\mathbf{\Phi} = \begin{bmatrix} \phi_1 & \dots & \phi_K \end{bmatrix}$

Θ . . . . . Confusion Matrix — $M \times K$ stochastic matrix to denote the conditional signal probabilities. In the CommonBelief model, all agent i in the model share the same matrix

$\Theta_{h,g}$ . . . . . Conditional signal probability of signal g given state h, $P(S = g|T = h)$

$\mathbf{\Theta}$ . . . . . Set of I confusion matrices, $\{\Theta^1, ..., \Theta^I\}$

Θ^i . . . . . $M \times K$ stochastic matrix to denote the unique conditional signal probabilities of agent i

$\mathcal{G}$ . . . . . Set of G agent types, {1,...,G}

γ_i . . . . . Random Variables that takes on value $g \in \mathcal{G}$ that represents group membership of agent i

Γ . . . . . Set of random variables γ_i for every agent i, $\{\gamma_1, ..., \gamma_I\}$

α . . . . . Hyperparameter that defines the distribution of agent type, $\alpha \in [0, 1]^G$. Example: $\gamma_i \sim Multinomial(\alpha)$

Λ^g . . . . . Hyperparameter that defines the distribution of confusion matrices of agents that belongs to group $g \in \mathcal{G}$

$\hat{\mathbf{\Theta}}_\mu$ . . . . . Set of G matrices,$\{\hat{\Theta}^1_\mu, ..., \hat{\Theta}^G_\mu, \}$, that represents estimated centeral confusion matrices from k-Means-Confusion algorithm

$\pi^i_{k,m}$ . . . . . State posterior of state m given agent i's observation of signal k, $P(T = m|S_i = k)$

Π^i . . . . . $K \times M$ stochastic matrix to denote the unique state posteriors for agent i

$\Psi^i_{g,l}$ . . . . . Probability that agent i will report signal l given that her observation is g, $P(r_i = l|S_i = g)$

Ψ^i . . . . . $K \times K$ stochastic matrix to represent mixed-strategy of agent i conditioned on her signal observation

$\boldsymbol{\Psi}$ . . . . . Set of I strategy matrices, $\{\Psi^1, ..., \Psi^I\}$

r_i . . . . . Signal report of agent i, $r_i \in \mathcal{S}$

r_{-i} . . . . . Set of signal reports by agent i's peers

$x : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ . . . . . A function that represents payment rule of peer prediction mechanism

$R : \mathcal{P} \times \Omega \to \mathbb{R} \cup \{\pm\infty\}$ . . Scoring Rule

Ω . . . . . Set of outcomes, $\{1, ..., m\}$

$\mathcal{P}$ . . . . . Probability distribution over Ω

R_L . . . . . Logarithmic Scoring Rule

R_Q . . . . . Quadratic Scoring Rule

R_S . . . . . Simple Linear Scoring Rule

List of Mechanisms

Chapter 1: Introduction

1.1. Background

One of the earliest records of distributing computational tasks to multiple human workers in scientific research dates back as far as 1758 when the French astronomer Alexis Clairaut employed two other astronomers to compute the returning time of the Halley's Comet (Grier, 2005). However, it is in the Twenty-First century with the emergence of online crowdsourcing platforms such as Amazon Mechanical Turk and Galaxy Zoo that scientists in many different disciplines have been able to access a multitude of human workers to analyze data that requires tasks currently out of reach of computers. The ease of distributing computational work to a vast number of human workers has ushered in a new paradigm of computational method called *human computation* (von Ahn & Dabbish, 2004).

Human computation poses new theoretical and practical challenges: (1) workers come in a wide range of levels of expertise, which are unknown *a priori* to the designer of human computation system; (2) some tasks generate multiple versions of reports by different workers that must be somehow combined to arrive at the actual estimate of the *truth*; (3) human workers must be properly incentivized via financial reward or non-financial means such as *fun*, rankings, badges, etc.; and (4) some human workers may participate in the system with the adversarial intent to sabotage the system.

Addressing the first two challenges, researchers in machine learning and statistics have proposed various innovative methods to model and infer the error-rates of the

human workers and to recover truth from noisy reports (Ghahramani & Kim, 2003; Welinder, Brandson, Belongie, & Perona, 2010; Whitehill, Ruvolo, Wu, Bergsma, & Movellan, 2009).

Interestingly, even before the emergence of online crowdsourcing platforms or the Internet, Dawid and Skene in 1979 proposed a model, using *confusion matrices*, to quantify the error-rates of medical clinicians evaluating their patients. Dawid and Skene showed that using the Expectation Maximization algorithm (Dempster, Laird, & Rubin, 1977) they can estimate the confusion matrices of individual clinicians and also predict the true states of the patients. Recently, their model has gained traction among human computation researchers whose goal is to quantify the error-rates of the human workers and also to infer the true labels from the multiple labels submitted by the error-prone workers (Lakkaraju, Leskovec, Kleinberg, & Mullainathan, 2015; Liu & Wang, 2012).

Addressing the third and fourth challenges in human computation, researchers in the information elicitation discipline have proposed a class of innovative mechanisms called *peer prediction mechanisms*. Peer prediction mechanism exploits correlation between the reports of the participants to determine their payments with the goal of implementing a payment rule that incentivizes the participants to truthfully report their signals (i.e. observations).

One of the earliest and most notable peer prediction mechanisms is the *Peer Prediction Method* (Miller, Resnick, & Zeckhauser, 2005). The Peer Prediction Method (PPM) is a *truthful peer prediction mechanism*, which means that the expected payment to an agent is strictly maximized when the agent reports her observed signal truthfully. The *truthful* property of PPM is attractive for human computation systems; however, PPM requires the designer to know *a priori* the *state priors* (i.e. distribution of the true states) and the *conditional signal probabilities* (i.e. the probability of observing a certain signal given a true state). Moreover, the base model of PPM requires homogeneity of the agents, which models all agents to share the

same biases or abilities. In human computation systems, this assumption is akin to believing that all workers share the same error-rates in computational tasks whether they be labeling images or translating texts. In reality, empirical results show that the human workers have varying error-rates (Ipeirotis & Paritosh, 2011; Welinder & Perona, 2010).

Consequently, while large-scale human computation systems have successfully adopted and implemented simpler peer prediction mechanisms such as the *Output Agreement* mechanism as examplified by the ESP game (von Ahn & Dabbish, 2004), they have not implemented a theoretically more robust peer prediction mechanism such as the Peer Prediction Method.

In this thesis, I combine the machine learning-based approach to model and infer the error-rates of the human workers and the properties of the Peer Prediction Method to design a new class of peer prediction mechanisms that does not require *a priori* knowledge of the state priors and the conditional signal probabilities. These new peer prediction mechanisms represent an unified approach to addressing the challenges in human computation.

1.2. Contributions

The main contribution of this thesis can be summarized as introducing an unified approach to designing incentive scheme in human computation system using machine learning algorithms and mechanism design. I introduce three new peer prediction mechanisms : the *Empirical Peer Prediction Method*, the *k-Means Peer Prediction Method*, and the *Empirical Scoring Rule Mechanism*.

The Empirical Peer Prediction Method (EPPM) and the k-Means Peer Prediction Method (kPPM) can be seen as extensions of the classical Peer Prediction Method (CPPM) (Miller et al., 2005). These new mechanisms build on CPPM by addressing two challenges left out by Miller et al.: (1) deriving the probabilistic belief

model used to design payment rules, and (2) modeling heterogeneous set of agents.

On the other hand, the Empirical Scoring Rule Mechanism (ESRM) takes a departure from CPPM. Instead of relying on the peer reports to compute the payment to an agent, ESRM uses recovered true states and the estimated state posteriors of the agent to pay the agent based on a strictly proper scoring rule.

As I introduce each mechanism, I progressively relax the assumption of homogeneity of the agents in their releveant models. Using simulated data, I demonstrate empirically that the three mechanisms are truthful peer prediction mechanisms and are robust against various reporting strategies.

Empirical Peer Prediction Method

The classical Peer Prediction Method is the first minimal-reporting truthful peer prediction mechanism. The minimal-reporting property refers to the mechanism requiring the agent to only report her privately observed signals, and the truthful property describes how the mechanism computes the payment to the agent such that the expected payment is strictly maximized when the agent truthfully reports her observed signals.

However, CPPM requires the critically important assumption that the mechanism designer knows *a priori* the state priors and the conditional signal probabilities of the model that define the private observation of signals by the agents. For example, in order to implement CPPM in an image labeling human computation system, the designer must know in advance the distribution of the different types of images in the data set; moreover, he must also know in advance the probabilities that a human worker will observe certain labels given an image.

Miller et al., in their original paper, introduce simple methods to obtain these probabilities using historical reports submitted by the agents; however, they leave this topic open as a future research opportunity.

In this thesis, I introduce the CommonBelief model and the *Empirical Peer*

Prediction Method (EPPM) as answers to the challenge left open by Miller et al. Similar to the base model of CPPM, the CommonBelief model maintains the assumption of homogeneity of agents. However, EPPM does not require that the designer know *a priori* the state priors or the conditional signal probabilities. EPPM estimates these probabilities from the agents' reports using the Expectation Maximization algorithm.

Given a sufficiently large number of reports, EPPM accurately infers the state priors and the conditional signal probabilities, and with these probabilites, EPPM computes the payment such that an agent's expected payment is maximized when she reports her signals truthfully.

In addition, borrowing the idea from market scoring rule (Hanson, 2007), I introduce a new payment rule for empirical peer prediction mechanisms including EPPM. This new payment rule, together with the EM algorithm, enables empirical peer prediction mechanisms to resolve the uninformative equilibria problem that afflicts many existing peer prediction mechanisms in the academic literature.

I test the mechanism using simulated data in various conditions including different number of world states, different number of agents, and different state priors. I also test EPPM against various reporting strategies including unilateral random reporting strategy, unilateral *Signal-prior* reporting strategy, and collusion strategy where all agents coordinate to report the same signal repeatedly. I demonstrate empirically that EPPM is robust against all these reporting strategies.

k-means Peer Prediction Method

In the paper that introduced CPPM, Miller et al. briefly discuss settings in which the agents' "tastes differ systematically." Without going into an in-depth discussion, they state that in such case the designer must model the different agent types explicitly, but they also leave this topic open as a future research opportunity.

In this thesis, I introduce the GroupBelief model and the *k-Means Peer*

Prediction Method (kPPM) as answers to the challenge left open by Miller et al. The GroupBelief model relaxes the assumption of homogeneity of agents. It models every agent as a member of $\mathcal{G} = \{1, ..., G\}$ groups, and that the agents belonging to the same group share similar error-rates or biases that are pertinent to that group.

I also introduce *k-Means-Confusion* algorithm, a variation of the k-means--algorithm (Chawla & Gionis, 2013) that identifies agents' membership in the agent groups based on their conditional signal probabilities.

Using the k-Means-Confusion algorithm and the EM algorithm, kPPM identifies the group membership of an agent and computes her payment according to her group peer reports.

I test kPPM under varying conditions including different number of world states and different number of agents using simulated data. I test kPPM against various reporting strategies including unilateral random reporting strategy, unilateral Signal-prior reporting strategy, and collusion strategy. I demonstrate empirically that kPPM is also robust against all the strategies.

Moreover, I show that the GroupConfusion model and kPPM provide a natural approach to inducing effort from the agents by rewarding the group of agents with lower error-rates with higher payments.

Empirical Scoring Rule Mechanism

I introduce the PrivateBelief model and the *Empirical Scoring Rule Mechanism* (ESRM). The PrivateBelief model assumes heterogeneous population of agents, which gives the designer the flexibility to assume that every agent is an unique individual with her own error-rates or biases. However, the heterogeneity assumption also makes implementation of peer prediction mechanisms problematic (Radanovic & Faltings, 2015).

As a radical departure from the existing peer prediction mechanisms, ESRM provides a solution to the heterogeneous agent problem by not requiring a reference

peer to determine the payment for an agent. After using the EM algorithm to recover the true states and estimate the state posteriors of the agent, ESRM pays the agent based on the agent's report alone. Nevertheless, ESRM may still be considered a peer prediction mechanism due the mechanism's reliance on the reports of many agents to recover the true states.

I test ESRM in various conditions including different number of world states and different number of agents. I also test ESRM against various reporting strategies including the collusion strategy to show that the mechanism is robust against all strategic reporting.

Although ESRM is introduced in conjunction with the PrivateBelief model, ESRM is also applicable in other models such as the CommonBelief model and the GroupBelief model. In addition to testing the mechanism accuracy and robustness in the PrivateBelief model, I compare the expected payments of ESRM against EPPM and kPPM in the CommonBelief model and the GroupBelief model, respectively, and show that ESRM is also applicable in these models as a truthful peer prediction mechanism.

Summary of the Contributions

- Proposes an unified approach that combines machine learning algorithm and mechanism design to design new peer prediction mechanisms that identify the error-rates and align the incentives of self-interested, error-prone workers in human computation system.
- Introduces a new class of minimal-reporting, truthful peer prediction mechanisms that do not require the mechanism designer to know in advance the probabilistic belief model.
- Introduces a new minimal-reporting peer prediction mechanism that models the systematic differences in the capabilities or biases of the agents.
- Introduces a variant of k-Means algorithm that identifies group membership of the human workers in human computation system based on their error-rates.
- Designs a new minimal-reporting truthful peer prediction mechanism for a model with heterogeneous population of agents.
- Introduces a new payment rule that, combined with the EM algorithm to estimate the model probabilities, helps to resolve the uninformative equilibria problem.
- Designs a new class of peer prediction mechanisms that are robust against collusion of agents thereby resolving the uninformative equilibria problem that is problematic in many previously proposed peer prediction mechanisms.

1.3. Outline

In Chapter 2, we take a brief survey over the information elicitation methods starting with the setting where truth is verifiable. We discuss *strictly proper scoring rules*, which serve as building blocks for the classical Peer Prediction Method and empirical peer prediction mechanisms. Then, in Section 2.2, we shift our discussion to the information elicitation methods where truth is not verifiable. We introduce two notable peer prediction mechanisms — the Output Agreement mechanism and the Peer Prediction Method — and discuss each mechanism's required assumptions and the challenges of its implementation in practice.

In Chapter 3, we turn our attention to machine learning-based models and algorithms to quantify error-rates of workers in human computation systems. We review the Expectation Maximization algorithm and discuss two existing models in the machine learning literature - the CommonConfusion model and the PrivateConfusion model. In Section 3.4, I introduce a new model called the GroupConfusion model and the k-Means-Confusion algorithm.

At the heart of this thesis is Chapter 4, where I introduce the empirical methods in peer prediction. First, in Section 4.1.1, I present the Empirical Peer Prediction Method, an empirical peer prediction mechanism for the CommonBelief model. In Section 4.2.1, I present the k-Means Peer Prediction Method, which is an empirical peer prediction mechanism for the GroupBelief model. Finally, in Section 4.3.1, I introduce the Empirical Scoring Rule Mechanism, which is an empirical peer prediction mechanism for the PrivateBelief Model.

Finally, we conclude this thesis in Chapter 5 with the summary of the lessons learned and a brief discussion on interesting future directions for the empirical methods in peer prediction.

Chapter 2: Information Elicitation

Online recommendation systems, such as those used by popular websites such as Amazon, eBay, and Yelp, elicit reviews and ratings from the visitors. Reviews and ratings inform other visitors and businesses that uses these websites as marketing platforms; moreover, helpful and informative reviews are also beneficial for those websites because they help boost more traffic. As a result, there is a significant financial incentive for these systems to elicit honest reviews from its visitors.

Some visitors share their opinions out of goodwill. However, eliciting informative reviews poses two major challenges in general:

- "Underprovision" — writing a review takes time and effort, and these investments do not directly benefit the reviewer, but only those who receive the information.

- "Honesty" — reviewer does not have an incentive to be honest about his or her opinion. For example, a reviewer who had a negative experience with a business may refrain from sharing her opinion out of fear of retaliation from the business.

Addressing these challenges is the concern for researchers of *information eliciation*, an academic discipline that is focussed on the study and design of methods to elicit honest report of private belief from self-interested human agents (Parkes & Seuken, 2017). In this chapter, we review various incentive schemes that have been proposed in the information eliciation literature. These schemes serve as important building blocks for new incentive schemes that I introduce in Chapter 4.

We start with a brief discussion about information elicitation methods in the setting where underlying truth is verifiable. For instance, consider a setting in which a pollster asks a weather forecaster to predict the probability that the weather tomorrow is sunny. The pollster can wait to observe the weather tomorrow and reward the forecaster for her prediction based on the outcome. In Section 2.1, I present *scoring rules* as solutions in such a setting.

Because the research in scoring rules is a mature field there is an increased interest in the study of information eliciation in settings where underlying truth is not verifiable (Waggoner & Chen, 2013). For instance, consider an online recommendation system such as Yelp that elicits reviews from its visitors about the quality of a restaurant's service. There is no objective standard that one can use to verify the "true" qualify of a restaurant's service. In such a setting, direct implementation of scoring rule is infeasible.

In Section 2.2, I introduce the idea of *peer prediction*, which uses the correlation between the reports of human agents to design reward payments. I introduce the base model of the peer prediction in Section 2.2.1. In Section 2.2.3, I present the simplest peer prediction mechanism, the Output Agreement mechanism. In Section 2.2.4, I present the classical Peer Prediction Method, the first minimal-reporting truthful peer prediction mechanism. For each mechanism, I discuss the required assumptions and describe the challenges in implementing it in a human computation system.

In Section 2.3, we briefly go over other noteworthy peer prediction mechanisms that have been proposed in the information eliciation literature. We conclude this chapter with a short discussion about the shortcomings of implementing the classical Peer Prediction Method and how those shortcomings relate to the empirical peer prediction mechanisms that I introduce in Chapter 4.

2.1. Scoring Rules

Let us consider a scenario in which an investment manager seeks an accurate prediction of the next quarter's earnings of Apple, Inc. from a research analyst. The investment managers wants to know the probability that Apple's next quarterly earning will be up from the last quarter's earning. With a lot of money at stake in this prediction, the investment manager decides to reward the analyst with a bonus payment at the end of the quarter if the analyst correctly predicts the outcome. The investment manager must design a payment scheme that will incentivize the analyst to make an effort to accurately predict the outcome and honestly report her prediction to the manager. What payment scheme should the investment manager adopt to incentivize the analyst to report truthfully?

This example describes a setting that we describe as *information elicitation with verifiable truth*. The outcome of the prediction is verifiable because it is realized after a period of uncertainty, and typically, the payment is withheld until the the outcome is verified.

In this section, we briefly discuss the properties of *strictly proper scoring rule* and introduce two such scoring rules — the *logarithmic scoring rule* and the *quadratic scoring rule*. Strictly proper scoring rules serve as important building blocks in designing truthful peer prediction mechanisms, which we discuss in detail in Section 2.2.

2.1.1 Scoring Rules

Scoring rules provide solutions to the incentive challenge in the information eliciation problem with verifiable truth.

Definition 2.1.1 (Scoring Rule)**.** *Given $\Omega = \{1, ..., m\}$ possible outcomes and a report $p = (p_1, ..., p_m) \in \mathcal{P}$ that defines the probability distribution over Ω, scoring rule is a function $R : \mathcal{P} \times \Omega \to \mathbb{R} \cup \{\pm\infty\}$.*

Let us consider a simple scoring rule, the *linear scoring rule*. The linear scoring rule receives a report of an agent's subjective belief about the probabilities of the outcomes, and it pays the agent a payment equivalent to the probability that the agent assigns to the realized outcome.

Definition 2.1.2 (Linear Scoring Rule)**.** *The linear scoring rule is a function defined as*

$$R_S(p, \omega) = p_\omega \tag{2.1}$$

for reported belief p and outcome ω, and where $R_S(p, \omega) \in [0, 1], \forall p \in \mathcal{P}, \omega \in \Omega$

It is easy to see that the linear scoring rule does not correctly incentivize the agent to truthfully report her subjective beliefs about the probabilities of the outcomes. We demonstrate this property with an example.

Example 1. *It is March 2016, in middle of the US Presidential Election Primaries. In the Democratic party, two candidates competing for the nomination of the party are Hillary Clinton and Bernie Sanders. There are only two possible outcomes, $\Omega =$ { "Hillary wins" , "Bernie wins"}.*

A polling organization requests a political pundit to submit her subjective belief about the probability that Hillary Clinton wins the nomination. At the end of the primaries when an outcome is realized, the polling organization will reward the pundit with a score based on the linear scoring rule.

The pundit believes $P(\omega_1 =$ "Hillary wins"$) = 0.6$ and $P(\omega_2 =$ "Bernie wins"$) = 0.4$. Let $\tilde{p} \in [0, 1]$ represent her reported belief about $P(\omega_1 =$ "Hillary wins"$)$. If her report is scored on the linear scoring rule, her expected score with respect to her report $\tilde{p}$ is

$$\begin{aligned} \mathbb{E}[R_S(\tilde{p}, \omega)] &= P(\omega_1) \cdot \tilde{p} + P(\omega_2) \cdot (1 - \tilde{p}) \\ &= 0.6 \cdot \tilde{p} + 0.4 \cdot (1 - \tilde{p}) \\ &= 0.4 + 0.2 \cdot \tilde{p} \end{aligned} \tag{2.2}$$

From equation 2.2, the pundit realizes that her expected score is optimized if she reports $\tilde{p} = 1$ instead of truthfully reporting her belief, $\tilde{p} = 0.6$. Therefore, the pundit exaggerates her belief about Hillary Clinton's chance of winning the nomination.

In general, the expected payment of the linear scoring rule is optimized when the agent puts all the weight on the outcome that she believes to be most likely (Parkes & Seuken, 2017). Therefore, the linear scoring rule is an example of scoring rule that is not *strictly proper*.

Definition 2.1.3 (Strictly Proper Scoring Rule)**.** *A scoring rule R is* proper *if an agent's expected score is maximized, with respect to an agent's beliefs $p \in \mathcal{P}$, by reporting truthfully, and is* strictly proper *if the truthful report is the only report that maximizes the agent's expected score.*

Strictly proper scoring rule is a type of scoring rule that correctly incentivizes the participating agents to report truthfully. In the following sections, we introduce two strictly proper scoring rules.

2.1.2 Logarithmic Scoring Rule

The *logarithmic scoring rule* is an example of strictly proper scoring rule.

Definition 2.1.4 (Logarithmic Scoring Rule)**.** *The logarithmic scoring rule is a function defined as*

$$R_L(p, \omega) = ln(p_\omega) \tag{2.3}$$

for reported belief p and outcome ω where $R_L(p, \omega) \in \mathbb{R}^- \cup \{-\infty\}$ for all $p \in \mathcal{P}$ and $\omega \in \Omega$.

Theorem 2.1.1. *Logarithmic scoring rule is a strictly proper scoring rule.*

Proof. Consider a set of outcomes $\Omega = \{1, ..., m\}$. An agent holds a subjective belief about the probabilities of the outcomes, which we denote as $p = (p_1, ..., p_m) \in \mathcal{P}$

where $p_\omega \in [0,1],\ \forall \omega \in \Omega$, and $\sum_{i=1}^{m} p_i = 1$. The agent's expected score given her report $\tilde{p}$ is,

$$\mathbb{E}[R_L(\tilde{p}, \omega)] = \sum_{i=1}^{m} p_i \cdot ln(\tilde{p}_i) \tag{2.4}$$

We find the optimal report by solving the first order derivative of the expected score with respect to $\tilde{p}_i$.

$$\begin{aligned} &\frac{\partial}{\partial \tilde{p}_i}(\sum_{j=1}^{m} p_j \cdot ln(\tilde{p}_j)) = 0 \\ &\Leftrightarrow \frac{p_i}{\tilde{p}_i} = 0 \\ &\Leftrightarrow p_i = \tilde{p}_i \end{aligned} \tag{2.5}$$

Checking the second order derivate,

$$\frac{\partial}{\partial \tilde{p}_i}\frac{p_i}{\tilde{p}_i} = -p_i \cdot \tilde{p}_i^2 < 0, \text{ because } p_i > 0 \tag{2.6}$$

Therefore, we prove that agent's score is maximized if she reports $\tilde{p}_i = p_i$ for all outcomes. □

Example 1 (continued)**.** *Let us come back to the example of the political pundit who must report her belief about $P(\omega_1 =$ "Hillary wins"$)$. The scoring rule of the poll has changed to the logarithmic scoring rule. Now the pundit's expected score with respect to her report $\tilde{p}$ is*

$$\begin{aligned} \mathbb{E}[R_L(\tilde{p}, \omega)] &= P(\omega_1) \cdot ln(\tilde{p}) + P(\omega_2) \cdot ln(1 - \tilde{p}) \\ &= 0.6 \cdot ln(\tilde{p}) + 0.4 \cdot ln(1 - \tilde{p}) \end{aligned} \tag{2.7}$$

She finds the report that maximizes her expected score by taking the first order deriva-

tive with respect to $\tilde{p}$,

$$\begin{aligned} & \frac{\partial}{\partial \tilde{p}}(0.6 \cdot ln(\tilde{p}) + 0.4 \cdot ln(1-\tilde{p})) = 0 \\ \Leftrightarrow\ & \frac{0.6}{\tilde{p}} - \frac{0.4}{1-\tilde{p}} = 0 \\ \Leftrightarrow\ & 0.6 \cdot (1-\tilde{p}) = 0.4 \cdot \tilde{p} \\ \Leftrightarrow\ & 0.6 = \tilde{p} \end{aligned} \tag{2.8}$$

Her expected score is maximized when she reports truthfully, $\tilde{p} = P(\omega_1)$.

While the logarithmic scoring rule is a strictly proper scoring rule, there are challenges to its implementation in practice. By the nature of logarithmic functions, the logarithmic scoring rule will yield $-\infty$ score; for example, the expected score for an agent is $-\infty$ if she believes one of the outcome is improbable with $p_i = 0$. One can offset the zero probability with a very small number $\epsilon > 0$ to prevent it from computing $-\infty$; however, this practice distorts the computation of the expected score.

Moreover, the logarithmic scoring rule suffers from *hypersensitivity*, which roughly describes a property that the expected score reacts very strongly to differences in small probabilites (Selten, 1998).

2.1.3 Quadratic Scoring Rule

The *quadratic scoring rule* is another strictly proper scoring rule that is an alternative to the logarithmic scoring rule.

Definition 2.1.5 (Quadratic Scoring Rule)**.** *The quadratic scoring rule is a function defined as*

$$R_Q(p, \omega) = 2 \cdot p_\omega - \sum_{i=1}^{m} p_i^2 \tag{2.9}$$

for reported belief p *and outcome* ω *where* $R_Q(p, \omega) \in \mathbb{R}$ *for all* $p \in \mathcal{P}$ *and* $\omega \in \Omega$.

Theorem 2.1.2. *Quadratic scoring rule is a strictly proper scoring rule.*

Proof. Consider a set of outcomes $\Omega = \{1, ..., m\}$ and an agent's subjective belief about the probabilities of the outcomes, $p = (p_1, ..., p_m) \in \mathcal{P}$ where $p_\omega \in [0, 1],\ \forall \omega \in \Omega$, and $\sum_{i=1}^{m} p_i = 1$. The agent's expected score given her report $\tilde{p}$ is,

$$\begin{aligned} \mathbb{E}[R_Q(\tilde{p}, \omega)] &= \sum_{i=1}^{m} p_i \cdot (2\tilde{p}_i - \sum_{j=1}^{m} \tilde{p}_j^2) \\ &= 2\sum_{i=1} p_i \cdot \tilde{p}_j - \sum_{h=1}^{m} \tilde{p}_m^2 \\ &= \sum_{i=1} p_i^2 - \sum_{j=1} (p_j - \tilde{p}_j)^2 \end{aligned} \tag{2.10}$$

Solving for the first order derivative of the expected score with respect to $\tilde{p}_i$,

$$\begin{aligned} &\frac{\partial}{\partial \tilde{p}_i}(\sum_{i=1} p_i^2 - \sum_{j=1} (p_j - \tilde{p}_j)^2) = 0 \\ \Leftrightarrow &2(p_i - \tilde{p}_i) = 0 \\ \Leftrightarrow &p_i = \tilde{p}_i \end{aligned} \tag{2.11}$$

Checking the second order derivative,

$$\frac{\partial}{\partial \tilde{p}_i} 2(p_i - \tilde{p}_i) = -2 \tag{2.12}$$

This is true for all $\tilde{p}_i$; therefore, we prove that agent's score is maximized if she reports $\tilde{p}_i = p_i$ for all outcomes. □

Example 1 (continued)**.** *Coming back to the political pundit example, the poll will now use the quadratic soring rule. The pundit's expected score with respect to her*

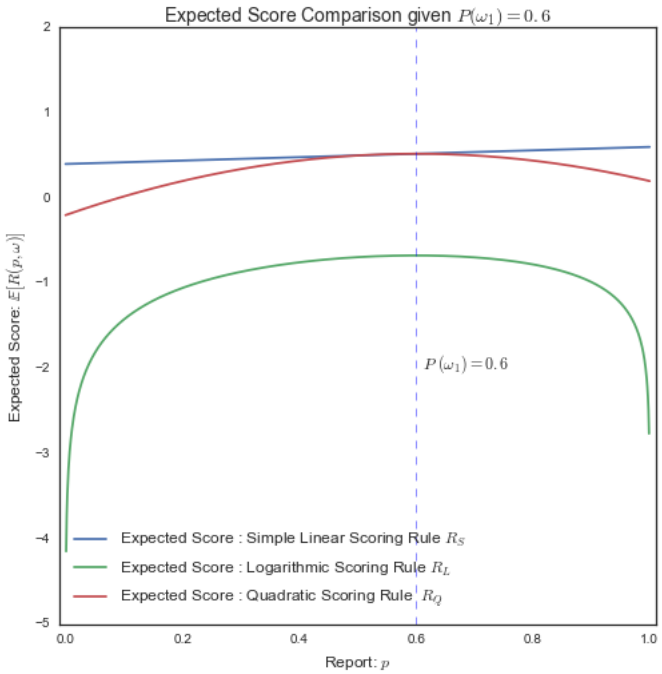


Figure 2.1: Example 1 Summary of Expected Scores

report $\tilde{p}$ *is*

$$\begin{aligned}\mathbb{E}[R_S(\tilde{p},\omega)] &= P(\omega_1)\cdot(2\cdot\tilde{p}-\tilde{p}^2-(1-\tilde{p})^2)+P(\omega_2)\cdot(2\cdot(1-\tilde{p})-(1-\tilde{p})^2-\tilde{p}^2)\\ &= 0.6\cdot(4\tilde{p}-2\tilde{p}^2-1)+0.4\cdot(1-2\tilde{p}^2)\\ &= 2.4\tilde{p}-2\tilde{p}^2-0.2\end{aligned} \tag{2.13}$$

She finds the report that maximizes her expected score by taking the first-order derivative with respect to $\tilde{p}$*,*

$$\begin{aligned}&\frac{\partial}{\partial\tilde{p}}(2.4\tilde{p}-2\tilde{p}^2-0.2)=0\\ \Leftrightarrow\ &2.4-4\tilde{p}=0\\ \Leftrightarrow\ &2.4=4\tilde{p}\\ \Leftrightarrow\ &\tilde{p}=0.6\end{aligned} \tag{2.14}$$

Her expected score is maximized when she reports truthfully, $\tilde{p} = P(\omega_1)$.

Figure 2.1 summarizes the expected scores of the pundit given her belief $P(\omega_1) = 0.6$ *for each scoring rule introduced in this section.*

We should note that for the logarithmic scoring rule the scores are always below zero for all reports and outcomes; the quadratic scoring rule may also compute scores below zero. In terms of payment to the agents, it is not intuitive how to interpret the negative payments. This problem may be partially alleviated by a positive affine transformation of the strictly proper scoring rules.

Proposition 2.1.3. *If a scoring rule* R *is a strictly proper scoring rule, then scoring rule,*

$$R^{'}(p, \omega) = \alpha + \beta \cdot R(p, \omega)$$

derived from positive affine transformation, with $\beta > 0$ *and* $\alpha \in \mathbb{R}$*, is also a strictly proper scoring rule (Parkes & Seuken, 2017).*

Nevertheless, positive affine transformation does not resolve the problems that arise from $-\infty$ score in the logarithmic scoring rule.

2.2. Peer Prediction Mechanisms

An entertainment news website aggregates movie ratings of several movie critics nationwide for an upcoming summer blockbuster movie. The website editor requests approximately 50 movie critics to submit their ratings of the blockbuster movie in the scale of 1 to 5. Because many online visitors rely on the website to provide helpful advice about what upcoming movies to watch, the editor considers it important to receive a helpful and honest rating from each movie critic. In order to incentivize effort, the editor decides to pay each critic a payment for his or her rating; unfortunately, unlike the investment scenario in Section 2.1, the editor has no objective and definitive outcome to score the movie critic's rating. Therefore, scoring rule is not

a viable option. What incentive scheme should the editor adopt to entice the movie critics to report their ratings truthfully?

The above situation is an example of information elicitation without verifiable truth. In these settings, one can adopt a *peer prediction mechanism*, which uses the correlation between the reports of the participating agent in order design a payment rule that correctly aligns the incentives of the agents.

In Section 2.2.1, we discuss the base model of peer prediction. In Section 2.2.2, I present an example of a human computation system in which peer prediction mechanism can play a crucial role. I use this example to demonstrate the properties of different peer prediction mechanisms introduced in the rest of this thesis.

We review two notable peer prediction mechanisms, the *Output Agreement* mechanism and the *Peer Prediction Method*, in Section 2.2.3 and Section 2.2.4, respectively. We go over the required assumptions and the challeneges in their implementations in large-scale human computation systems.

We conclude the chapter with a brief discussion about other peer prediction mechanisms that have been introduced in the information elicitation literature and the different drawbacks in implementing them in human computation systems.

2.2.1 Base Model

A *world state* is represented by a random variable T that can take on a possible value $m \in \mathcal{T} = \{1, ..., M\}$. The probability that T takes on value m is the *state prior* denoted as $P(T = m)$. We assume that every value $m \in \mathcal{T}$ has $P(T = m) > 0$, because any value m that has $P(T = m) = 0$ may be eliminated from the model without affecting the behavior of the mechanism.

There are $I \geq 2$ self-interested and risk-neutral agents. Each agent $i \in \mathcal{I} = \{1, ..., I\}$ privately observes a *signal*, which is denoted by random variable S_i with possible realization of $k \in \mathcal{S} = \{1, ..., K\}$. Each agent's observed signal is identically and independently distributed conditioned on a state, and it's probability of realiza-

tion, the *conditional signal probability*, is represented as $P(S_i = k|T = m),\ \forall k \in \mathcal{S},\ \forall m \in \mathcal{T}$.

The state priors and conditional signal probabilities are collectively called the *belief model* of peer prediction because they form the agent's probabilistic belief about the privately observed signal of another agent, which we call *signal posterior*. The signal posterior, which we compute as follows,

$$P(S_j = s^{'}|S_i = s) = \sum_{h=1}^{M} P(S_j = s^{'}|T = h)P(T = h|S_i = s) \tag{2.15}$$

$$\text{where } P(T = h|S_i = s) = \frac{P(S_i = s|T = h)P(T = t)}{P(S_i = s)} \tag{2.16}$$

represents agent i's belief about the probability of her peer $j \in \mathcal{I}$ observing signal $s^{'}$ given that agent i observed signal s. The *signal prior* $P(S_i = s)$, the denominator in equation 2.16, is

$$P(S_i = s) = \sum_{h=1}^{M} P(S_i = s|T = h)P(T = h) \tag{2.17}$$

In the base model, we assume that every agent shares the same belief model and that the mechanism designer is also aware of this belief model. Also, because all agents share the same belief model, it holds that $P(S_j = s^{'}|S_i = s) = P(S_i = s^{'}|S_j = s)$ for all signals $s, s^{'} \in \mathcal{S}$ and all agents $i, j \in \mathcal{I}$. In Chapter 4, we relax these assumptions and explore models in which the mechanism designer has no knowledge of the belief model and also a model with heterogeneous population of agents.

Based on her privately observed signal S_i, agent i reports $r_i \in \mathcal{S}$ to the mechanism. Conditioned on her signal, agent i reports according to her *strategy*, which is a function $\sigma_i : \mathcal{S} \to \mathcal{S}$.

Definition 2.2.1 (Truthful Reporting Strategy)**.** *Agent i that adopts the* truthful reporting strategy *reports signal $\sigma_i(s_i) = s_i$ for every signal $s_i \in \mathcal{S}$.*

Given a model, the goal of a *peer prediction mechanism* is to elicit agent's signals by designing a payment rule, x, such that the agent's utility is maximized when she truthfully reports her observed signal to the mechanism.

Definition 2.2.2 (Peer Prediction Mechanism)**.** *A* peer prediction mechanism *is a mechanism that collects reports r_i from every agent $i \in \mathcal{I}$ and pays $x(r_i, r_{-i}) \in \mathbb{R}$ to agent i according to the payment rule x and the reports of the other agents r_{-i}.*

Assuming that agent i's peers $\{-i\}$ all report truthfully, the *truthful reporting strategy* by agent i is a *strict, correlated equilibrium* of a peer prediction mechanism if

$$\mathbb{E}_{s_{-i}}[x_i(s_i, s_{-i})] > \mathbb{E}_{s_{-i}}[x_i(r_i, s_{-i})],\ \forall r_i \neq s_i, \forall r_i \in \mathcal{S} \tag{2.18}$$

In words, the truthful reporting strategy strictly maximizes the expected payment of agent i.

Definition 2.2.3 (Truthful Peer Prediction Mechanism)**.** *A peer prediction mechanism is* truthful *if the truthful reporting strategy is the strict, correlated equilibrium for all agents in the mechanism.*

All *truthful peer prediction mechanism* requires the crucial assumption of *Stochastic Relevance* in the belief model (Johnson, Miller, Pratt, & Zeckhauser, 2002).

Definition 2.2.4 (Stochastic Relevance)**.** *Signal S_i for agent i is said to be* Stochastically Relevant *for signal S_j for agent j if for every pair of signal realizations $s^{'}, s^{''}$ of S_i, there exists some realization s of S_j such that*

$$P(S_j = s|S_i = s^{'}) \neq P(S_j = s|S_i = s^{''}), \quad \forall s \in \mathcal{S}$$

2.2.2 Example

Here, I present an example of human computation system that may benefit from adopting a truthful peer prediction mechanism. I use this example throughout the rest of this thesis to illustrate various peer prediction mechanisms.

Suppose a computer scientist is engaged in research in image processing. He devises a new revolutionary machine learning algorithm that can be considered a great advancement in the field of machine learning-based image processing. However, this algorithm requires an extremely large training data set. Fortunately, a colleague donates a large collection of one million images of cats and dogs; unfortunately, none of the images are labeled.

Instead of attempting to label all the images himself, the computer scientist decides to tap into the power of the crowd by hiring human workers on the Amazon Mechanical Turk, a popular crowdsourcing website.

Initially, he decides to pay the workers a fixed payment of \$0.01 per image; however, he soon realizes that the labels submitted by the workers contain many mistakes and are overall unreliable. Labeling an image takes a certain amount of effort and time investment; however, because he is paying the workers a fixed payment per image, they are incentivized to quickly label as many images as possible, which leads to some workers making many mistakes and some workers even intentionally labelling all the images as 'dog' regardless what they observe in the image.

At this time, the computer scientist decides to verify each label submitted by a worker before he pays the worker the fixed payment. While the quality of the labels have certainly improved, the verification process has led to a significant slowdown in the speed of the project. He is now looking at each individual image to verify the labels, and this process is no quicker than looking at the images and labeling them himself. His verfication process has become a bottleneck that prevents the crowdsourcing project from becoming scalable.

The computer scientist wishes to design the crowdsourcing project such that it rewards the workers to put in the effort and honestly report what they observe; nevertheless, he also wants to do away with the verification process that can prevent the project from scaling. A truthful peer prediction mechanism may provide a solution to this challenge.

Example 2. *An image represents a world with two possible states $T = 0$ ('dog') or $T = 1$ ('cat'). The two states are mutually exclusive; in other words, there are no images that contain both a cat and a dog together. Likewise, there are two possible signals $S = 0$ and $S = 1$, which represent a 'dog' and a 'cat', respectively.*

Based on a cursory observation of the collection, the computer scientist concludes that the distribution of the images of cats and dogs are approximately 50/50; therefore, he estimates the state priors as

$$P(T = 0) = P(T = 1) = 0.5 \tag{2.19}$$

Furthermore, based on his past experience with the human workers, the computer scientist assumes the following conditional signal probabilities for all workers,

$$P(S = 0|T = 0) = 0.7 \tag{2.20}$$

$$P(S = 1|T = 1) = 0.8 \tag{2.21}$$

In other words, a worker correctly observes images of dogs 70% of the time and correctly observes images of cats 80% of the time. Moreover, the computer scientist assumes that all workers share the same conditional signal probabilities. Thus, given this belief model, he computes the joint probabilities of the signals $P(S_j, S_i),\ \forall i, j \in \mathcal{I}$,

S_i \ S_j	0	1
0	0.265	0.185
1	0.185	0.365

Table 2.1: Example 2 Joint Signal Probabilities

From the state priors and the conditional signal probabilities he also computes the signal priors; for example, he estimates the probability that a worker observes a

dog in any given image in the collection to be

$$\begin{aligned} P(S=0) &= P(S=0|T=0)\cdot P(T=0) + P(S=0|T=1)\cdot P(T=1) \\ &= 0.7\cdot 0.5 + 0.2\cdot 0.5 \\ &= 0.45 \end{aligned} \tag{2.22}$$

From the signal prior and the joint probabilities of the signals, he computes the signal posteriors. For example, the probability that a worker j observes a dog in an image if worker i observed a dog in the same image is

$$\begin{aligned} P(S_j=0|S_i=0) &= \frac{P(S_j=0, S_i=0)}{P(S_i=0)} \\ &= \frac{0.265}{0.45} \\ &\approx 0.589 \end{aligned} \tag{2.23}$$

Likewise, the signal posterior of worker j observing a dog if worker i observed a cat is

$$\begin{aligned} P(S_j=0|S_i=1) &= \frac{P(S_j=0, S_i=1)}{P(S_i=1)} \\ &= \frac{0.185}{0.55} \\ &\approx 0.331 \end{aligned} \tag{2.24}$$

Computing the rest of the signal posteriors, we note that this belief model satisfies Stochastic Relevance. For any agent $i, j \in \mathcal{I}$,

$$P(S_j=0|S_i=0) \approx 0.589 \neq 0.331 \approx P(S_j=0|S_i=1) \tag{2.25}$$

$$P(S_j=1|S_i=1) \approx 0.664 \neq 0.411 \approx P(S_j=1|S_i=0) \tag{2.26}$$

We should also note that $P(S_j=0|S_i=0) > P(S_j=1|S_i=0)$ and $P(S_j=1|S_i=1) > P(S_j=0|S_i=1)$, which intuitively suggest that worker i believes that

her peer j is more likely to observe the signal that she observed than any other signal.

2.2.3 Output Agreement

We begin our discussion on peer prediction mechanisms with the simplest mechanism, the *Output Agreement* (OA) mechanism.

Mechanism 2.2.1 (**Output Agreement**). *The Output Agreement mechanism is defined as:*

1. *Mechanism receives reports $r_i \in \mathcal{S}$ from every agent $i \in \mathcal{I}$.*
2. *Pays every agent $i \in \mathcal{I}$ a fixed payment of τ if $r_i = r_j$, where r_j is a report from a reference agent j selected from the set of peer agents $\{-i\}$.*

For binary signals, the payment rule of OA can be summarized as follows,

r_i \ r_j	0	1
0	τ	0
1	0	τ

Table 2.2: Binary Payment Rule of Output Agreement

In order for OA to be a truthful peer prediction mechanism, the signal posteriors of the common belief model must satisfy the *Diagonalization Property*.

Definition 2.2.5 (Diagonalization Property). *Signal posteriors of the belief model satisfies the* Diagonalization Property *if for every agent i and j,*

$$P(S_j = s|S_i = s) > P(S_j = s^{'}|S_i = s),\ \forall s, s^{'} \in \mathcal{S} \text{ and } s^{'} \neq s \tag{2.27}$$

Intuitively, the Diagonalization Property describes the setting in which if an agent observes a certain signal, then her peer is most likely to also observe that same signal.

Theorem 2.2.1. *The Output Agreement mechanism is a truthful peer prediction mechanism if and only if the Diagonalization Property holds true for the belief model.*

Proof. The proof begins with the definition of the truthful peer prediction mechanism,

$$\begin{aligned} &\mathbb{E}_{s_j}[x_i(s_i, s_j)|S_i = s_i] > \mathbb{E}_{s_j}[x_i(r_i, s_j)|S_i = s_i],\ \forall s_i, s_j \in \mathcal{S}, s_i \neq r_i \\ \Leftrightarrow\ &P(S_j = s_i|S_i = s_i) \cdot \tau > P(S_j = r_i|S_i = s_i) \cdot \tau \\ \Leftrightarrow\ &P(S_j = s_i|S_i = s_i) > P(S_j = r_i|S_i = s_i) \end{aligned} \tag{2.28}$$

The last inequality of 2.28 is the definition of the Diagonalization Property. □

Example 2 (continued)**.** *The computer scientist confirms that the belief model satisfies the Diagonalization Property.*

$$P(S_j = 0|S_i = 0) \approx 0.589 > 0.411 \approx P(S_j = 1|S_i = 0)$$

$$P(S_j = 1|S_i = 1) \approx 0.664 > 0.336 \approx P(S_j = 0|S_i = 1)$$

Therefore, he can safely assume that OA is a truthful peer prediction mechanism *in this setting, and that if implemented in his project, OA will correctly incentivize the workers to label the images truthfully.*

However, let us consider an example of belief model where the Diagonalization Property does not hold.

Example 3. *The computer scientist learns that although his estimate of the conditional signal probabilities were correct, he made a crucial error in estimating the state priors. In fact, the data set contains far more images of cats than dogs. As such, the computer scientist adjusts the state priors as follows*

$$P(T = 0) = 0.3 \text{ and } P(T = 1) = 0.7 \tag{2.29}$$

From the same conditional signal probabilities, he computes the new joint probabilities

of the signals,

S_i \ S_j	0	1
0	0.13	0.17
1	0.17	0.53

Table 2.3: Example 3 Joint Signal Probabilities

Given the new belief model, the computer scientist observes that the Diagonalization Property no longer holds for the model. For example,

$$P(S_j = 0|S_i = 0) = \frac{0.13}{0.3} \approx 0.43 \not> 0.57 \approx \frac{0.17}{0.3} = P(S_j = 1|S_i = 0) \tag{2.30}$$

Consequently, when a worker observes an image of a 'dog' because she believes that she has a higher chance of being matched with another worker who reported the same image as a 'cat', she lies by reporting it as 'cat' in order to maximize her expected payment.

OA is not a truthful peer prediction mechanism where an agent holds the minority opinion or observation. Nevertheless, because the Diagonalization Property is a reasonable assumption in many circumstances, OA has been successfully adopted and implemented in large scale human computation systems such as the ESP game (von Ahn & Dabbish, 2004).

2.2.4 Peer Prediction Method

The classical *Peer Prediction Method* (CPPM) is the first minimal-reporting truthful peer prediction mechanism (Miller et al., 2005). The mechanism uses the signal posteriors and a strictly proper scoring rule to design a payment rule that correctly aligns the incentive of the agents such that truthful reporting strategy is a strict, correlated equilibrium for all agents.

Mechanism 2.2.2 (Peer Prediction Method). *The Peer Prediction Method is defined as:*

1 *Mechanism receives reports* $r_i \in \mathcal{S}$ *from every agent* $i \in \mathcal{I}$.

2 *Pays agent* i *a payment based on a* strictly proper scoring rule, $x_i(r_i, r_j) = R(P(S_j|S_i = r_i), r_j)$, *where* r_j *is the report from a reference agent* j *and* $P(S_j|S_i = r_i)$ *is the signal posteriors of agent* j *given agent* i*'s report.*

For binary signals, the payment rule of CPPM can be summarized as,

r_i \ r_j	0	1
0	$R(P(S_j\|S_i = 0), 0)$	$R(P(S_j\|S_i = 0), 1)$
1	$R(P(S_j\|S_i = 1), 0)$	$R(P(S_j\|S_i = 1), 1)$

Table 2.4: Binary Payment Rule of CPPM

Theorem 2.2.2. *Given that the belief model satisfies Stochastic Relevance, the Peer Prediction Method is a truthful peer prediction mechanism for any strictly proper scoring rule.*

Proof. Assume that all peer agents $\{-i\}$ report truthfully. For every reference agent $j \in \{-i\}$, her report s_j is distributed according to the signal posterior $P(S_j = s_j|S_i = s_i)$ of the belief model such that

$$\mathbb{E}_{s_j}[R(P(S_j|S_i = s_i), s_j)|S_i = s_i] > \mathbb{E}_{s_j}[R(P(S_j|S_i = r_i), s_j)|S_i = s_i] \tag{2.31}$$

for all $s_i, s_j \in \mathcal{S}$ and all reports $r_i \neq s_i$ because R is a strictly poper scoring rule and $P(S_j = s_j|S_i = s_i) \neq P(S_j = s_j|S_i = r_i)$ due to the Stochastic Relevance.

Note that the left hand side of equation 2.31 denotes the expected payment to agent i if she reports her beliefs about the signal posteriors $P(S_j|S_i = s_i)$ to a strictly proper scoring rule. The right hand side denotes the expected payment to agent i if she reports her beliefs about the signal posteriors $P(S_j|S_i = r_i)$. If the two

probabilities are different, then it follows that the left hand side is greater due to the strictly proper property of R. □

Example 3 (continued)**.** *Let us return to the example in which OA failed to be a truthful peer prediction mechanism. The computer scientist decides to adopt the classical Peer Prediction Method with the quadratic scoring rule. He designs a payment schedule shown in Table 2.5.*

r_i \ r_j	0	1
0	0.35	0.63
1	−0.16	0.88

Table 2.5: Example 3 CPPM Payment Matrix

Assume that worker i observes a dog in an image. If she reports truthfully, her expected payment is

$$\begin{aligned} & P(S_j = 0|S_i = 0) \cdot 0.35 + P(S_j = 1|S_i = 0) \cdot 0.63 \\ = & \ 0.43 \cdot 0.35 + 0.57 \cdot 0.63 \\ \approx & \ 0.6601 \end{aligned}$$

In contrast, if she reports 'cat' instead,

$$\begin{aligned} & P(S_j = 0|S_i = 0) \cdot -0.16 + P(S_j = 1|S_i = 0) \cdot 0.88 \\ = & \ 0.43 \cdot -0.16 + 0.57 \cdot 0.88 \\ \approx & \ 0.4328 \end{aligned}$$

The worker i who observes a dog in an image is better off reporting it as 'dog' to the mechanism. Suppose that the worker i observes a cat instead; if she reports truthfully,

then her expected payment is

$$\begin{aligned} & P(S_j = 0|S_i = 1) \cdot -0.16 + P(S_j = 1|S_i = 1) \cdot 0.88 \\ = & \ 0.24 \cdot -0.16 + 0.76 \cdot 0.88 \\ \approx & \ 0.6304 \end{aligned}$$

In contrast, if the worker reports a 'dog', her expected payment is

$$\begin{aligned} & P(S_j = 0|S_i = 1) \cdot 0.35 + P(S_j = 1|S_i = 1) \cdot 0.63 \\ = & \ 0.24 \cdot 0.35 + 0.76 \cdot 0.63 \\ \approx & \ 0.5628 \end{aligned}$$

We confirm that the worker is better off by reporting truthfully for both observations.

In contrast to OA, CPPM does not require the assumption that the common belief model satisfies the Diagonalization Property. It assumes that the belief model satisfies Stochastic Relevance, which is a less restrictive assumption. On the other hand, CPPM also has three notable shortcomings that makes implementation of it in practice challenging.

- A priori knowledge of the belief model — CPPM requires that the designer knows in advance the state priors and conditional signal probabilities. Miller et al., in the original paper that introduces CPPM, suggest that the designer estimate these probabilities from historical reports of the agents; however, they do not go into detail how the mechanism designer should acquire reliable historical reports before implementing the mechanism or how to compute the probabilities from the historical reports.

- Homogeneity of agents — CPPM requires that all agents share the same belief model. In human computation systems, this requirement is equivalent to modeling all human workers as equally proficient in their tasks. In reality, some

workers exhibit higher error-rates than others. Miller et al. suggest that the designer model different types of agents if one suspects there are systematic differences within the population of agents; however, they do not discuss in detail how the designer should model the different type of agents or how the designer should identify the different types of agents.

- *Uninformative equilibria* - In addition to the truthful strategy, CPPM has Nash equilibria that yields no information to the mechanism designer.

Definition 2.2.6 (Uninformative Equilibria)**.** *A strategy σ_i is* uninformative *for agent i if for every signal s and s', $\sigma_i(s) = \sigma_i(s')$. A Nash equilibrium $\sigma = \{\sigma_1(s_1), ..., \sigma_I(s_I)\}$ is* uninformative *if σ_i is uninformative for every agent $i \in \mathcal{I}$.*

Uninformative equilibria are problematic in peer prediction mechanisms because they reveal no information about the observed signals of the agents. An example of uninformative equilibrium is a strategy profile in which all agents collude to report the same signal. The problem that arises from the uninformative equilibria is demonstrated in the following example.

Example 3 (continued)**.** *Given the joint signal distribution in Table 2.3 and the payment schedule in Table 2.5, the expected payment to a worker for truthful reporting strategy, is*

$$0.13 \cdot 0.35 + 0.17 \cdot 0.63 + 0.17 \cdot -0.16 + 0.53 \cdot 0.88 \approx 0.5918$$

However, this payment schedule also has other Nash equilibria. Consider the joint payment matrix in Table 2.6. Note that strategically coordinating all workers to always report 0 *or* 1 *are* pure-strategy Nash-equilibria*. What's more problematic is that all workers strategically always reporting* 1 *has greater expected payment (i.e.* 0.88*) than the truthful reporting strategy.*

The computer scientist realizes that all the labels coming in from the workers only contain the label 'cat.' Apparently, a word has gotten around among the workers

r_i \ r_j	0	1
0	$(0.35, 0.35)$	$(0.63, -0.16)$
1	$(-0.16, 0.63)$	$(0.88, 0.88)$

Table 2.6: Example 3 CPPM Joint Payment Matrix

that if they coordinate to label all images as 'cat,' they can increase their expected payments.

Many existing peer prediction mechanisms including CPPM are vulnerable to the uninformative equilibria problem (Waggoner & Chen, 2013). Some possible solutions suggested in the literature include verifying a small percentage of the reports (Waggoner & Chen, 2013). Another solution is to introduce a truthful agent (or agents) into the mix (Jurca & Faltings, 2005). Finally, recently introduced *Multi-task 01-Mechanism* (Shnayder, Agarwal, Frongillo, & Parkes, 2016) is shown to be strictly proper against uninformative equilibria but proper against other reporting strategies.

2.3. Other Mechanisms

In this chapter, we only discussed minimal-reporting peer prediction mechanisms, the Output Agreement mechanism and the classical Peer Prediction Method. The *minimal-reporting* property refers to the mechanism's requirement that it only receives the signal reports from the agents. However, in the information elicitation literature, there are many variations of peer prediction mechanism, and in this section, we briefly present the most notable variants.

A noteworthy variation of OA is the *1 / Prior Mechanism* (Jurca & Faltings, 2009, 2011). The 1 / Prior Mechanism has been noted as a "fix" of OA by altering the fixed payment τ with the signal priors. For binary signals, the payment rule of the 1 / Prior Mechanism can be summarized in Table 2.7.

Similar to OA, the mechanism designer of 1 / Prior Mechanism must assume that that the belief model satisfies the *Fractional Diagonalization Property.* Also,

r_i \ r_j	0	1
0	$\frac{\tau}{P(S=0)}$	0
1	0	$\frac{\tau}{P(S=1)}$

Table 2.7: Binary Payment Rule of 1 / Prior mechanism

the mechanism does not require that the designer know the full belief model; it only requires the designer to know the signal priors.

The *Empirical Shadowing Method* (Witkowski & Parkes, 2012) is a minimal-reporting peer prediction mechanism that does not require the agents to share the same belief model and also does not require the designer to know in advance the individual belief models of the agents.

There are also a class of peer prediction mechanisms that do not exhibit the minimal-reporting property. These mechanisms require the agents to make at least two reports. The most notable and earliest example of such mechanism is the *Bayesian Truth Serum* (BTS) (Prelec, 2004). This mechanism requires that an agent make two reports : (1) signal report and (2) prediction report — belief about the signal posteriors. Also, BTS requires "sufficiently large" or countably infinite population of agents.

Building on BTS, another dual-reporting mechanism is the *Robust Bayesian Truth Serum* (RBTS) (Witkowski & Parkes, 2012), which relaxes the large population requirement of BTS. The *1 / Posterior Bayesian Truth Serum* (Radanovic & Faltings, 2013) generalizes RBTS to multi-signal setting. Finally, one of the latest peer prediction mechanisms is the *Knowledge Free Peer Prediction Mechanism* (Zhang & Chen, 2014), which takes iterative steps (two rounds) to receive signal reports from the agents in round one, which then are shared to all agents in round two to elicit prediction reports.

While these dual-reporting peer prediction mechanisms are truthful peer prediction mechanisms, they are difficult to implement in a large scale human compu-

tation system in practice because they require that the agents compute and report the signal posteriors, which is a task that is likely to be cognitively costly for human agents.

2.4. Conclusion

In this chapter, I presented the major concepts in the information eliciation literature. We examined strictly proper scoring rules for the information eliciation with verifiable truth. We also examined peer prediction mechanisms for the information elicitation without the verifiable truth.

We discussed that while CPPM correctly aligns the incentives of the participating agents, implementing it in a large scale human computation system is challenging due to three factors: (1) *a priori* knowledge of the belief model, (2) requiring the same belief model for all agents, and (3) the uninformative equilibria. Factors (1) and (2) are model specific issues of CPPM, and factor (3) is a challenge shared by many other peer prediction mechanisms.

In Chapter 4, I introduce a new class of peer prediction mechanisms called the *empirical peer prediction mechanisms* that directly address these concerns. However, before I introduce the empirical peer prediction mechanisms, we examine machine learning-based models and algorithms to infer the error-rates of human workers in human computation systems.

Chapter 3: Latent Confusion Matrix and The EM Algorithms

Facilitated by the rise in popularity of crowdsourcing platforms, human computation has emerged as a powerful method for processing large volume of data that cannot be processed by traditional computing methods. Scientists have particularly benefitted from human computation systems, with one of the most notable success story being Galaxy Zoo, a crowdsourced astronomical image labeling platform.

Human computation systems have garnered special interest among computer scientists as they have relied on the systems to build large datasets that are used in training and testing of machine learning algorithms in various applications (Ipeirotis & Paritosh, 2011). For example, the ESP game (von Ahn & Dabbish, 2004) tapped into human workers to label images, which is an easy task for a typical human being but difficult for any machine learning-based image processing system even in 2016. In turn, these human-generated labels from the ESP game, and others, have contributed to the advancement of machine learning-based image processing systems by serving as training and testing data set (Welinder et al., 2010).

However, the labels submitted by the human workers are also prone to errors. For this reason, researchers have had keen interest in finding methods to improve the human computation systems by identifying those workers who exhibit high error-rates or who may even have malicious intent to sabotage human computation systems (Joglekar, Garcia-Molina, & Parameswaran, 2013; Welinder et al., 2010; Whitehill et al., 2009).

Interestingly, the method that has recently gained a broad interest among the machine learning researchers in quantifying the error-rates of the human workers is a method that was introduced before the emergence of the human computation systems or even the Internet. In 1979, Dawid and Skene introduced a model that uses confusion matrices to quanitify the error-rates of medical clinicians diagnosing patients. They used the Expectation Maximization algorithm (Dempster et al., 1977) to estimate the error-rates of the clinicians and also infer the true diagnoses of the patients. Many researchers in computer science and statistics have adopted their model and have introduce other models and algorithms that build on it (Ghahramani & Kim, 2003; Lakkaraju et al., 2015; Liu & Wang, 2012).

In Section 3.1, we briefly discuss the general concept behind the Expectation Maximization algorithm. The EM algorithm computes the maximum likelihood estimate (MLE) of the confusion matrices for the probabilistic models introduced in the later sections of this chapter. It also serves as a crucial step in empirical peer prediction mechanisms that I introduce in Chapter 4.

In Section 3.2, we discuss the CommonConfusion model (Liu & Wang, 2012) in which all agents share the same confusion matrix. Then, in Section 3.3, we review the PrivateConfusion model (Dawid & Skene, 1979) in which every agent has her own unique confusion matrix. For each model, I describe the EM algorithm that computes the confusion matrices and recovers the true states based on the reports of the agents.

In Section 3.4, I introduce the GroupConfusion model which models the individual agent as a member of a group that shares similar confusion matrices. I also propose k-Means-Confusion, a variant of k-means-- algorithm (Chawla & Gionis, 2013), which indentifies the group membership of the agents and finds the *central* confusion matrices of the groups.

Finally, in Section 3.5, we conclude this chapter with a brief discussion about extensions of the PrivateConfusion model in the machine learning literature.

3.1. Expectation Maximization Algorithm

The Expectation Maximization (EM) algorithm (Dempster et al., 1977) is an unsupervised learning algorithm frequently used in data clustering. The EM algorithm finds the *local* maximum likelihood estimate (MLE) of parameters of a statistical model involving latent variables. In this section, we briefly discuss the general concept behind the EM algorithm . We adopt the notation and description of the EM algorithm from Bishop (2006).

3.1.1 General EM Algorithm

Consider a probabilistic model in which we denote the set of observable variables by $\boldsymbol{X}$ and the set of unobservable, latent variables by $\boldsymbol{Z}$. We describe $\{\boldsymbol{X}, \boldsymbol{Z}\}$ as the *complete* data set and the observed data $\boldsymbol{X}$ as *incomplete* data. The joint distribution of the complete data $P(\boldsymbol{X}, \boldsymbol{Z}|\theta)$ is parameterized by a set of parameters denoted by θ. Suppose that given the complete data $\{\boldsymbol{X}, \boldsymbol{Z}\}$, finding the MLE of the log-likelihood of the complete data set $\ln P(\boldsymbol{X}, \boldsymbol{Z}|\theta)$ is easy. However, we do not observe $\boldsymbol{Z}$, and suppose that finding the MLE of the log-likelihood of incomplete data $\ln P(\boldsymbol{X}|\theta)$ is difficult.

The goal of the EM algorithm is to find the the maximum likelihood estimate (MLE) of the parameters θ given the a likelihood function of the observed data $\boldsymbol{X}$. In summary,

$$\theta_{MLE} = \arg\max_{\theta} \ln P(\boldsymbol{X}|\theta) \tag{3.1}$$

$$\text{where } \ln P(\boldsymbol{X}|\theta) = \ln(\sum_{\boldsymbol{Z}} P(\boldsymbol{X}, \boldsymbol{Z}|\theta)) \tag{3.2}$$

While we do not have the values of the latent variables $\boldsymbol{Z}$, we can compute the posterior distribution $P(\boldsymbol{Z}|\boldsymbol{X}, \theta)$ given the observed data $\boldsymbol{X}$ and the estimate for the set of parameters θ^0. Using the posterior distribution of latent values given by $P(\boldsymbol{Z}|\boldsymbol{X}, \theta)$, we can find the expectation of the log-likelihood of $P(\boldsymbol{X}, \boldsymbol{Z}|\theta)$, which we

denote as $Q(\theta, \theta^0)$:

$$Q(\theta, \theta^0) = \mathbb{E}[\ln P(\boldsymbol{X}, \boldsymbol{Z}|\theta)] = \sum_{\boldsymbol{Z}} P(\boldsymbol{Z}|\boldsymbol{X}, \theta^0) \ln P(\boldsymbol{X}, \boldsymbol{Z}|\theta) \tag{3.3}$$

From the expectation of the log-likelihood $Q(\theta, \theta^0)$, we find a new estimate of the parameter of model,

$$\hat{\theta} = \arg\max_{\theta} Q(\theta, \theta^0) \tag{3.4}$$

The EM algorithm takes iterative steps to solve for θ_{MLE}. The algorithm begins by setting initial values for θ^0. In the E-step, which is the step that finds the expectation of the log-likelihood $Q(\theta, \theta^0)$, the algorithm computes the posterior distribution of the latent variable $P(\boldsymbol{Z}|\boldsymbol{X}, \theta)$. In the M-step, the algorithm finds the MLE values of the parameters θ from $Q(\theta, \theta^0)$. The EM algorithm alternates between the E-step and the M-step repeatedly until there is convergence of the estimates of θ between iterations.

We describe the general EM algorithm below.

Algorithm 3.1.1 (The General Expectation Maximization Algorithm)**.**

1. *Choose initial values for the estimate of parameters* θ^0*.*

2. *Iterate until convergence:*

   1 *E-Step: Compute* $P(\boldsymbol{Z}|\boldsymbol{X}, \theta^0)$*.*

   2 *M-Step: Compute* $\hat{\theta} = \arg\max_{\theta} Q(\theta, \theta^0)$*.*

   3 *Check for convergence in* $P(\boldsymbol{X}|\theta)$ *(or,* θ*). If no convergence, continue the iteration after*

   $$\theta^0 \leftarrow \hat{\theta}$$

3. *Return* $\hat{\theta}$*.*

We should note that the likelihood function may have multiple maxima. While the EM algorithm is guaranteed to converge toward a maximum of the likelihood function, this maximum may not be the *global* maximum (Bishop, 2006).

In the following sections, we examine applications of the EM algorithm in finding the MLE of the confusion matrices and the state priors in the models.

3.2. CommonConfusion Model

We begin with a simple model, the CommonConfusion model, in which all agents in a human computation system share the same confusion matrix. The CommonConfusion model was introduced by Liu and Wang (2012) as a simplification of the PrivateConfusion model, which we discuss in Section 3.3.

3.2.1 Model

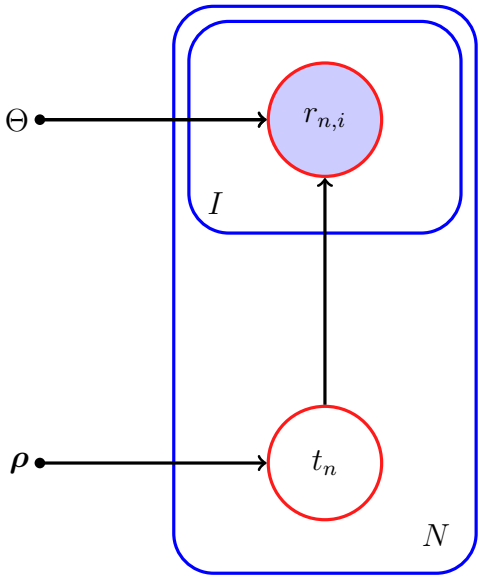


Figure 3.1: CommonConfusion Model

We adopt some notations from the base model of peer prediction mechanism presented in Chapter 2 Section 2.2.1. However, instead of one world, there are N different worlds (or, *items*) where T_n represents the state of the n-th item. Each item is independently and identically distributed according to a Multinomial distribution

parameterized by state priors $\boldsymbol{\rho}$:

$$t_n \sim Multinomial(\boldsymbol{\rho})$$

where $\boldsymbol{\rho} = \begin{bmatrix} \rho_1 & \dots & \rho_M \end{bmatrix}$ is a probability vector such that $\sum_{h=1}^{M} \rho_h = 1$.

Collectively, $\boldsymbol{t} = \{t_1, ..., t_N\}$ represents the set of true states of all N items. The true states are not observable to the system designer. As noted in the beginning of Chapter 2, this may be due to the fact that in some settings, such as movie reviews, there is no objective standard to identify the true state or in a large-scale human computation system, it may not be feasible for the designer to know the true state of every item in a very large data set.

The $M \times K$ *confusion matrix* Θ represents the shared error-rates of the agents for every state $m \in \mathcal{T}$,

$$\Theta = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_M \end{bmatrix} \tag{3.5}$$

where the m-th row of the confusion matrix, $\boldsymbol{\theta}_m = \begin{bmatrix} \theta_{m,1} & \dots & \theta_{m,K} \end{bmatrix}$, is a probability vector where $\sum_{g=1}^{K} \theta_{m,g} = 1$. Moreover, $\theta_{m,k}$, which is the value in the m-th row and the k-th column of Θ, represents the probability that a agent will report signal k conditioned on the true state m, $P(r_{n,i} = k | T_n = m)$, for any item $n \in (1, ..., N)$.

For each item $n \in (1, ..., N)$, every agent $i \in \mathcal{I}$ reports her observation of the true state t_n, which we denote as $r_{n,i}$. The report can take on any value of $k \in \mathcal{S} = \{1, ..., K\}$, and it is distributed according to a Multinomial distribution parameterized by $\boldsymbol{\theta}_m$ if $t_n = m$,

$$r_{n,i} \sim Multinomial(\boldsymbol{\theta}_m)$$

We denote all reports by every agent for every item as $\boldsymbol{r} = \{\boldsymbol{r}_n : n \in (1, ..., N)\}$ where $\boldsymbol{r}_n = \{r_{n,i} : i \in (1, ..., I)\}$.

Unlike the base model of peer prediction, the CommonConfusion model does not include agent's strategy σ. In the machine learning literature, researchers are primarily interested in inferring the error-rates of the human agents, and they model the reporting strategy as a factor that is conveyed in the error-rates.

The graphical model representation in Figure 3.1 summarizes the Common-Confusion model. Arrows indicate conditional dependence. For example, the true state t_n of item n is conditionally dependent on the state priors $\boldsymbol{\rho}$. Shaded node indicates that that the corresponding variable is an observed variable. Finally, the blue box with label N is a shorthand method to represent N number of t_n variables. For the reports $r_{n,i}$, the number of reports $N \times I$ is represented by including the node $r_{n,i}$ inside both boxes N and I.

3.2.2 Algorithm

We review the EM algorithm to find the maximum likelihood estimate of two parameters Θ and $\boldsymbol{\rho}$ given the observed reports $\boldsymbol{r}$. Formally, given $P(\boldsymbol{r}|\Theta, \boldsymbol{\rho})$, the likelihood function that models the observed data, the goal of the EM algorithm is to compute

$$\Theta_{MLE} = \arg\max_{\Theta} P(\boldsymbol{r}|\Theta, \boldsymbol{\rho}) \tag{3.6}$$

$$\boldsymbol{\rho}_{MLE} = \arg\max_{\boldsymbol{\rho}} P(\boldsymbol{r}|\Theta, \boldsymbol{\rho}) \tag{3.7}$$

Consider item n and agent i, the combined likelihood of the report $r_{n,i}$ and true label $t_n = h$, given the common confusion matrix Θ is

$$P(r_{n,i}, t_n = h|\Theta) = \prod_{g=1}^{K} \Theta_{h,g}^{\mathbb{I}(r_{n,i}=g)} \tag{3.8}$$

Since all agents observe their signals independently, the likelihood of the collective

reports of all I agents for item n is

$$P(\boldsymbol{r}_n, t_n = h|\Theta) = \prod_{g=1}^{K} \Theta_{h,g}^{\sum_{i=1}^{I} \mathbb{I}(r_{n,i}=g)} \tag{3.9}$$

and unconditioned on $t_n = h$ and where $\boldsymbol{\rho}$ is known,

$$P(\boldsymbol{r}_n, t_n|\boldsymbol{\rho}, \Theta) = \prod_{h=1}^{M} \{\rho_h \cdot \prod_{g=1}^{K} \Theta_{h,g}^{\sum_{i=1}^{I} \mathbb{I}(r_{n,i}=g)}\}^{\mathbb{I}(t_n=h)} \tag{3.10}$$

Finally, since each item is independently distributed,

$$P(\boldsymbol{r}, \boldsymbol{t}|\boldsymbol{\rho}, \Theta) = \prod_{n=1}^{N} \prod_{h=1}^{M} \{\rho_h \cdot \prod_{g=1}^{K} \Theta_{h,g}^{\sum_{i=1}^{I} \mathbb{I}(r_{n,i}=g)}\}^{\mathbb{I}(t_n=h)} \tag{3.11}$$

If we know the true states of N items, $\boldsymbol{t}$, we can find the MLE of Θ and $\boldsymbol{\rho}$ by solving the following,

$$\hat{\Theta}_{m,k} = \frac{\sum_{n=1}^{N} \sum_{i=1}^{I} \mathbb{I}(t_n = m) \cdot \mathbb{I}(r_{n,i} = k)}{\sum_{g=1}^{K} \sum_{n=1}^{N} \sum_{i=1}^{I} \mathbb{I}(t_n = m) \cdot \mathbb{I}(r_{n,i} = g)}, \ \forall m \in \mathcal{T}, \ \forall k \in \mathcal{S} \tag{3.12}$$

$$\hat{\rho}_m = \frac{\sum_{n=1}^{N} \mathbb{I}(t_n = m)}{\sum_{h=1}^{M} \sum_{n=1}^{N} \mathbb{I}(t_n = h)}, \forall m \in \mathcal{T} \tag{3.13}$$

If the parameters of the likelihood Θ and $\boldsymbol{\rho}$ are known, but the true states $\boldsymbol{t}$ are unknown, we can use the Bayes rule to compute the posterior probabilities of the true states as well.

$$\begin{aligned} P(t_n = m|\boldsymbol{r}_n, \boldsymbol{\rho}, \Theta) &= \frac{P(\boldsymbol{r}_n, \Theta|t_n = m) \cdot P(t_n = m|\boldsymbol{\rho})}{\sum_{j=1}^{M} P(\boldsymbol{r}_n, \Theta|t_n = j)} \\ &\text{where prior, } P(t_n = m|\boldsymbol{\rho}) = \rho_m \\ &= \frac{\rho_m \prod_{g=1}^{K} \Theta_{m,g}^{\sum_{i=1}^{I} \mathbb{I}(r_{n,i}=g)}}{\sum_{j=1}^{M} \rho_j \prod_{g=1}^{K} \Theta_{j,g}^{\sum_{i=1}^{I} \mathbb{I}(r_{n,i}=g)}} \end{aligned} \tag{3.14}$$

However, the system designer does not have knowledge of the true states $\boldsymbol{t}$, the dis-

tribution of the states $\boldsymbol{\rho}$, or the confusion matrix Θ. This is where the EM algorithm is used.

First, we find the expectation of the likelihood function,

$$\begin{aligned} Q(\Theta, \boldsymbol{\rho}, \Theta^0, \boldsymbol{\rho}^0) &= \mathbb{E}[P(\boldsymbol{r}_n, t_n|\Theta, \boldsymbol{\rho})] \\ &= \sum_{g=1}^{M} P(t_n = g|\boldsymbol{r}_n, \Theta^0, \boldsymbol{\rho}^0) \cdot P(\boldsymbol{r}_n, t_n = g|\Theta, \boldsymbol{\rho}) \\ &= \sum_{g=1}^{M} \frac{\rho_m \prod_{g=1}^{K} \Theta_{m,g}^{\sum_{i=1}^{I} \mathbb{I}((r_{n,i}=g)}}{\sum_{j=1}^{M} \rho_j \prod_{g=1}^{K} \Theta_{j,g}^{\sum_{i=1}^{I} \mathbb{I}((r_{n,i}=g)}} \cdot \{\rho_h \cdot \prod_{g=1}^{K} \Theta_{h,g}^{\sum_{i=1}^{I} \mathbb{I}(r_{n,i}=g)}\}^{\mathbb{I}(t_n=g)} \\ &= \sum_{g=1}^{M} \rho_m \prod_{g=1}^{K} \Theta_{m,g}^{\sum_{i=1}^{I} \mathbb{I}(r_{n,i}=g)} \end{aligned} \tag{3.15}$$

We denote the posterior of the true states $Z_{n,m} = P(t_n = m|\boldsymbol{r}_n, \Theta^0, \boldsymbol{\rho}^0)$ and use it to compute the MLE of Θ and $\boldsymbol{\rho}$ in place of $\mathbb{I}(t_n = m)$,

$$\hat{\Theta} = \arg\max_{\Theta} \; Q(\Theta, \boldsymbol{\rho}, \Theta^0, \boldsymbol{\rho}^0) \tag{3.16}$$

$$\Rightarrow \hat{\Theta}_{m,k} = \frac{\sum_{i=1}^{I} \sum_{n=1}^{N} Z_{n,m} \cdot \mathbb{I}(r_{n,i} = k)}{\sum_{j=1}^{K} \sum_{i=1}^{I} \sum_{n=1}^{N} Z_{n,m} \cdot \mathbb{I}(r_{n,i} = j)}, \; \forall m \in \mathcal{T}, \; \forall k \in \mathcal{S} \tag{3.17}$$

$$\hat{\boldsymbol{\rho}} = \arg\max_{\boldsymbol{\rho}} \; Q(\Theta, \boldsymbol{\rho}, \Theta^0, \boldsymbol{\rho}^0) \tag{3.18}$$

$$\Rightarrow \hat{\rho}_m = \frac{\sum_{n=1}^{N} Z_{n,m}}{\sum_{h=1}^{M} \sum_{n=1}^{N} Z_{n,h}}, \forall m \in \mathcal{T} \tag{3.19}$$

Since computing the posteriors of the states $Z_{n,m}$ depend on the estimates of Θ and $\boldsymbol{\rho}$ and estimating Θ and $\boldsymbol{\rho}$ depend on the posteriors $Z_{n,m}$, we take an iterative approach using the EM algorithm. We summarize the full EM algorithm for the CommonConfusion model below:

Algorithm 3.2.1 (EM Algorithm for CommonConfusion Model)**.**

Given $\boldsymbol{r}$ *and parameter* λ*,*

1. *Initialize:*

$$\hat{\rho}_m = \frac{1}{M}$$

$$\hat{\Theta}_{h,g} = \begin{cases} \frac{\lambda}{\lambda+K} & \text{if } h = g, \\ \frac{1}{\lambda+K} & \text{otherwise} \end{cases}$$

2. *Iterate until convergence:*

   1 *E-Step:*

   Compute $P(T_n = m|\boldsymbol{r}_n, \Theta, \boldsymbol{\rho}),\ \forall m \in \mathcal{T},\ \forall n \in (1, ..., N),$

$$Z_{n,m} = P(T_n = m|\boldsymbol{r}_n, \Theta, \boldsymbol{\rho}) = \frac{\rho_m \prod_{k=1}^{K} \Theta_{m,k}^{\sum_{i=1}^{I} \mathbb{I}(r_{n,i}=k)}}{\sum_{j=1}^{M} \rho_j \prod_{k=1}^{K} \Theta_{j,k}^{\sum_{i=1}^{I} \mathbb{I}(r_{n,i}=k)}} \tag{3.20}$$

   2 *M-Step:*

   Compute $\hat{\Theta} = \arg\max_{\Theta} Q(\Theta, \boldsymbol{\rho}, \Theta^0, \boldsymbol{\rho}^0)$

$$\hat{\Theta}_{m,k} = \frac{\sum_{i=1}^{I} \sum_{n=1}^{N} Z_{n,m} \cdot \mathbb{I}(r_{n,i} = k)}{\sum_{j=1}^{K} \sum_{i=1}^{I} \sum_{n=1}^{N} Z_{n,m} \cdot \mathbb{I}(r_{n,i} = j)},\ \forall m \in \mathcal{T},\ \forall k \in \mathcal{S} \tag{3.21}$$

   Compute $\hat{\boldsymbol{\rho}} = \arg\max_{\rho} Q(\Theta, \boldsymbol{\rho}, \Theta^0, \boldsymbol{\rho}^0)$

$$\hat{\rho}_m = \frac{\sum_{n=1}^{N} Z_{n,m}}{\sum_{h=1}^{M} \sum_{n=1}^{N} Z_{n,h}},\ \forall m \in \mathcal{T} \tag{3.22}$$

3. *Compute:*

$$\hat{t}_n = \underset{m \in (1,...,M)}{\arg\max}\ Z_{n,m},\ \forall n \in (1, ..., N) \tag{3.23}$$

4. *Return:* $\hat{\Theta}$*,* $\hat{\boldsymbol{\rho}}$*, and* $\hat{\boldsymbol{t}}$

In order to demonstrate how the EM algorithm solves for the parameters $\boldsymbol{\rho}$ and Θ in the CommonConfusion model, we create five synthetic agents with the

following common confusion matrix,

$$\Theta = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0.15 & 0.15 & 0.7 \end{bmatrix}$$

We generate $N = 1000$ states from the following state priors,

$$\boldsymbol{\rho} = \begin{bmatrix} 0.7 & 0.2 & 0.1 \end{bmatrix}$$

From the true states and the confusion matrix, we sample reports $\boldsymbol{r} = \{r_{n,i} |\ n \in (1, ..., 1000),\ i \in (1, ..., 5)\}$ according to the generative model described in Section 3.2.1. Setting parameter $\lambda = 3$, we run the EM algorithm to compute $\hat{\Theta}$ and $\hat{\boldsymbol{\rho}}$. The steps of the EM algorithm in regards to Θ are as follows,

$$\hat{\Theta}^{(0)} = \begin{bmatrix} 0.500 & 0.167 & 0.167 \\ 0.167 & 0.500 & 0.167 \\ 0.167 & 0.167 & 0.500 \end{bmatrix} \Rightarrow \hat{\Theta}^{(1)} = \begin{bmatrix} 0.726 & 0.185 & 0.089 \\ 0.267 & 0.625 & 0.109 \\ 0.242 & 0.202 & 0.556 \end{bmatrix} \Rightarrow \cdots$$

$$\Rightarrow \hat{\Theta}^{(31)} = \begin{bmatrix} 0.691 & 0.214 & 0.095 \\ 0.198 & 0.697 & 0.104 \\ 0.140 & 0.178 & 0.681 \end{bmatrix} \Rightarrow \cdots$$

$$\Rightarrow \hat{\Theta}^{(61)} = \begin{bmatrix} 0.691 & 0.214 & 0.095 \\ 0.197 & 0.699 & 0.104 \\ 0.140 & 0.178 & 0.681 \end{bmatrix}$$

We observe that the estimate of the confusion matrix after the first iteration of the EM algorithm, $\hat{\Theta}^{(1)}$, already appears closer to the true Θ than $\hat{\Theta}^{(0)}$. We see the convergence of the algorithm at the 31st iteration where $\hat{\Theta}^{(31)}$ appears close to the

final value, $\hat{\Theta}^{(61)}$. We also observe the steps of the algorithm in regards to $\hat{\boldsymbol{\rho}}$:

$$\begin{aligned}\hat{\boldsymbol{\rho}}^{(0)} = \begin{bmatrix}0.333 & 0.333 & 0.333\end{bmatrix} \Rightarrow &\hat{\boldsymbol{\rho}}^{(1)} = \begin{bmatrix}0.581 & 0.278 & 0.141\end{bmatrix} \Rightarrow \dots \\ \Rightarrow &\hat{\boldsymbol{\rho}}^{(31)} = \begin{bmatrix}0.686 & 0.206 & 0.107\end{bmatrix} \Rightarrow \dots \\ \Rightarrow &\hat{\boldsymbol{\rho}}^{(61)} = \begin{bmatrix}0.687 & 0.206 & 0.108\end{bmatrix}\end{aligned}$$

By the first iteration, we see that the estimate of the state priors is moving toward the true state prior. By the time the algorithm terminates after the 61st iteration, $\hat{\boldsymbol{\rho}}^{(61)}$ closely reflects the true state priors.

3.2.3 Analysis

We use synthetic data to examine the accuracy of the EM algorithm in the CommonConfusion model under different environments. We fix the set of states as $\mathcal{T} = \{1, 2, 3\}$ and the set of signals as $\mathcal{S} = \{1, 2, 3\}$. We define the confusion matrix to be:

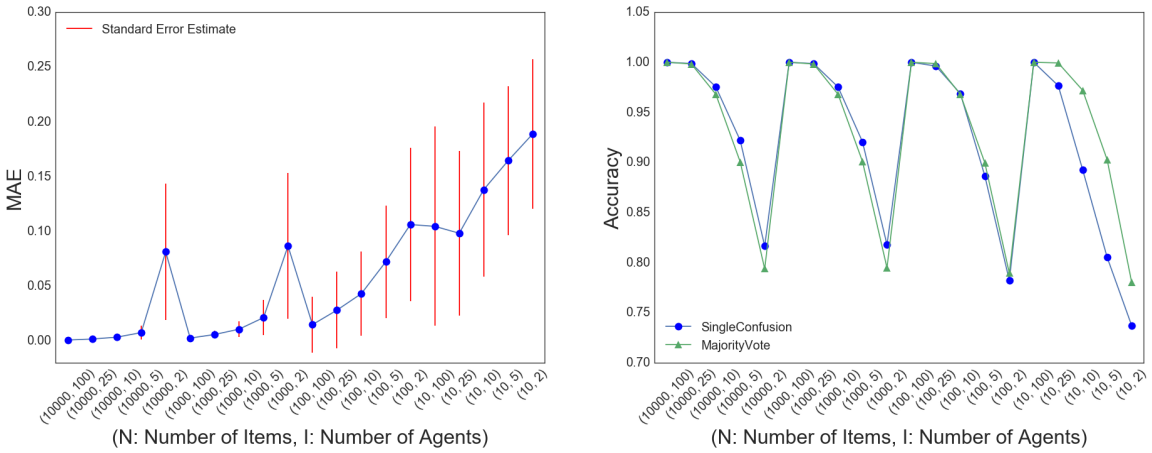
$$\Theta = \begin{bmatrix}0.7 & 0.2 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0.15 & 0.15 & 0.7\end{bmatrix}$$

We vary the number of items, $N = (10, 100, 1000, 10000)$, and the number of agents, $I = (2, 5, 10, 25, 100)$. For each permutation $(N \times I)$, we experiment on three different state priors — $\boldsymbol{\rho}_1 = \begin{bmatrix}1/3 & 1/3 & 1/3\end{bmatrix}$, $\boldsymbol{\rho}_2 = \begin{bmatrix}0.7 & 0.2 & 0.1\end{bmatrix}$, and $\boldsymbol{\rho}_3 = \begin{bmatrix}0.95 & 0.025 & 0.025\end{bmatrix}$ — by generating the states and reports synthetically from the parameters. We run 100 simulations per each permutation and state priors and report the average numbers.

We measure the accuracy in recovering the confusion matrix with the mean absolute error (MAE),

$$MAE(\hat{\Theta}, \Theta) = \frac{1}{M \cdot K} \sum_{h=1}^{M} \sum_{g=1}^{K} |\hat{\Theta}_{h,g} - \Theta_{h,g}| \tag{3.24}$$

We also measure the accuracy in recovering the true labels using the EM algorithm and compare it against the accuracy of the majority vote, which is a simple algorithm that selects the mode signal of the item reports $\boldsymbol{r}_n$ for every item $n \in (1, ..., N)$.



(a) Mean Absolute Errors of $\hat{\Theta}$ **(b) Accuracy Score of $\hat{t}$**

Figure 3.2: CommonConfusion Model: EM Performance

Figure 3.2 summarizes the accuracy in recovering the confusion matrix and the true labels. As the number of data points $(N \times I)$ declines, the error of the recovered confusion matrix, as measured by MAE, increases.

We see in Figure 3.2a that the MAE is more sensitive to the number of agents than the number of items. The algorithm performs significantly worse, as measured by MAE, in (10000×2) than in (1000×5) even though the former has 15000 more data points. This is also true when we compare the algorithm's performance in (1000×2) and in (100×5).

The accuracy of the recovered labels as shown in Figure 3.2b shows a similar trend where the decline in the number of agents lead to deterioration in the EM algorithm's performance. We also find that the EM algorithm outperforms the majority vote method consistently until N is 100 and below. We should note the troubling trend where $N = 10$; the EM algorithm performs quite poorly against the major-

ity vote in these environments. Where the number of items is quite small, the EM algorithm may not be well suited for the CommonConfusion model.

3.3. PrivateConfusion Model

In 1979, Dawid and Skene introduced a model that uses the confusion matrices to quantify the individual error-rates of medical clinicians evaluating the patients. Recently, this model, which we call the PrivateConfusion model, has gained traction among researchers in machine learning and statistics (Ghahramani & Kim, 2003; Lakkaraju et al., 2015; Liu & Wang, 2012) to model the error-rates of workers in the human computation systems. We review the PrivateConfusion model in this section.

3.3.1 Model

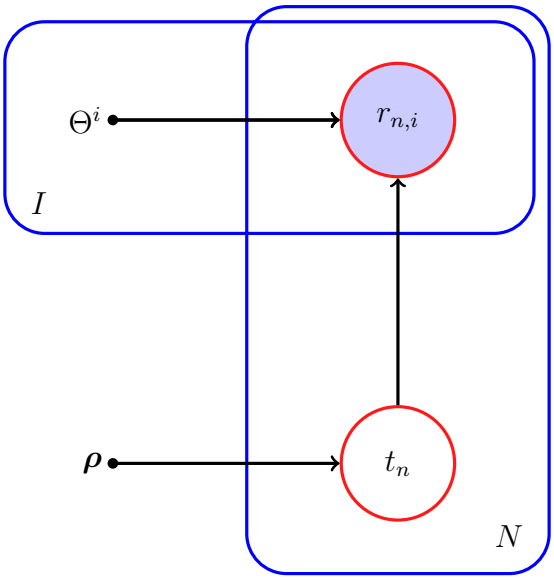


Figure 3.3: PrivateConfusion Model

In the PrivateConfusion model, every agent $i \in \mathcal{I}$ has her own unique

confusion matrix Θ^i. We denote $\boldsymbol{\Theta}$ as a set of confusion matrices:

$$\boldsymbol{\Theta} = \{\Theta^1, ..., \Theta^I\}$$

The reports by agent i on the n-th item is distributed according to Θ^i, agent i's unique individual confusion matrix, and the true state $t_n = m$:

$$r_{n,i} \sim Multinomial(\boldsymbol{\theta}^i_m)$$

where $\boldsymbol{\theta}^i_m$ represents the m-th row of the confusion matrix Θ^i and a probability vector $\boldsymbol{\theta}^i_m = \begin{bmatrix}\theta^i_{m,1} & \dots & \theta^i_{m,K}\end{bmatrix}$ such that $\sum_{g=1}^K \theta^i_{m,g} = 1$.

The graphical model representation in Figure 3.3 summarizes the PRIVATE-CONFUSION model.

3.3.2 Algorithm

The EM algorithm to compute the MLE of the $P(\boldsymbol{r}|\boldsymbol{\Theta}, \boldsymbol{\rho})$ in the PRIVATE-CONFUSION model is similar to that of the COMMONCONFUSION model discussed in Section 3.2. The posteriors of the true states now account for multiple unique confusion matrices.

$$Z_{n,m} = P(T_n = m|\boldsymbol{\Theta}, \boldsymbol{\rho}) = \frac{\rho_m \prod_{i=1}^I \prod_{k=1}^K (\Theta^i_{m,k})^{\mathbb{I}(r_{n,i}=k)}}{\sum_{j=1}^M \rho_j \prod_{i=1}^I \prod_{k=1}^K (\Theta^i_{j,k})^{\mathbb{I}(r_{n,i}=k)}} \tag{3.25}$$

We summarize the full EM algorithm for the PRIVATECONFUSION model below:

Algorithm 3.3.1 (EM Algorithm for PrivateConfusion Model)**.**
Given $\boldsymbol{r}$ *and parameter* λ*,*

1. *Initialize:*

$$\hat{\rho}_m = \frac{1}{M}$$

$$\hat{\Theta}^i_{h,g} = \begin{cases} \frac{\lambda}{\lambda+K} & \textit{if } h = g, \\ \frac{1}{\lambda+K} & \textit{otherwise} \end{cases}$$

2. Iterate until convergence:

1 E-Step:

Compute $P(T_n = m|\Theta, \rho),\ \forall m \in \mathcal{T},\ \forall n \in (1, ..., N)$,

$$Z_{n,m} = P(T_n = m|\boldsymbol{\Theta}, \boldsymbol{\rho}) = \frac{\rho_m \prod_{i=1}^{I} \prod_{k=1}^{K} (\Theta^i_{m,k})^{\mathbb{I}(r_{n,i}=k)}}{\sum_{j=1}^{M} \rho_j \prod_{i=1}^{I} \prod_{k=1}^{K} (\Theta^i_{j,k})^{\mathbb{I}(r_{n,i}=k)}} \tag{3.26}$$

2 M-Step:

Compute $\hat{\Theta} = \arg\max_{\Theta} Q(\Theta, \boldsymbol{\rho}, \Theta^0, \boldsymbol{\rho}^0)$

$$\hat{\Theta}_{m,k} = \frac{\sum_{i=1}^{I} \sum_{n=1}^{N} Z_{n,m} \cdot \mathbb{I}(r_{n,i} = k)}{\sum_{j=1}^{K} \sum_{i=1}^{I} \sum_{n=1}^{N} Z_{n,m} \cdot \mathbb{I}(r_{n,i} = j)},\ \forall m \in \mathcal{T},\ \forall k \in \mathcal{S} \tag{3.27}$$

Compute $\hat{\boldsymbol{\rho}} = \arg\max_{\rho} Q(\Theta, \boldsymbol{\rho}, \Theta^0, \boldsymbol{\rho}^0)$

$$\hat{\rho}_m = \frac{\sum_{n=1}^{N} Z_{n,m}}{\sum_{h=1}^{M} \sum_{n=1}^{N} Z_{n,h}},\ \forall m \in \mathcal{T} \tag{3.28}$$

3. Compute:

$$\hat{t}_n = \underset{m \in (1,...,M)}{\arg\max}\ Z_{n,m},\ \forall n \in (1, ..., N)$$

4. Return: $\hat{\boldsymbol{\Theta}}$, $\hat{\boldsymbol{\rho}}$, *and* $\hat{\boldsymbol{t}}$

3.3.3 Analysis

Similar to the experiment for the CommonConfusion model in Section 3.2.3, we use synthetic data to examine the accuracy of the EM algorithm in the Private-Confusion model under different environments.

We define the set of states as $\mathcal{T} = \{1, 2, 3\}$ and the set of signals as $\mathcal{S} =$

$\{1, 2, 3\}$. We vary the number of items, $N = (10, 100, 1000, 10000)$, and the number of agents, $I = (2, 5, 10, 25, 100)$. For each permutation of $(N \times I)$, we experiment on three different state priors — $\boldsymbol{\rho}_1 = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$, $\boldsymbol{\rho}_2 = \begin{bmatrix} 0.7 & 0.2 & 0.1 \end{bmatrix}$, and $\boldsymbol{\rho}_3 = \begin{bmatrix} 0.95 & 0.025 & 0.025 \end{bmatrix}$. We synthesize our data for each permutation of $(N \times I)$ and state priors for each run, and we report the average number and standard error of 100 runs.

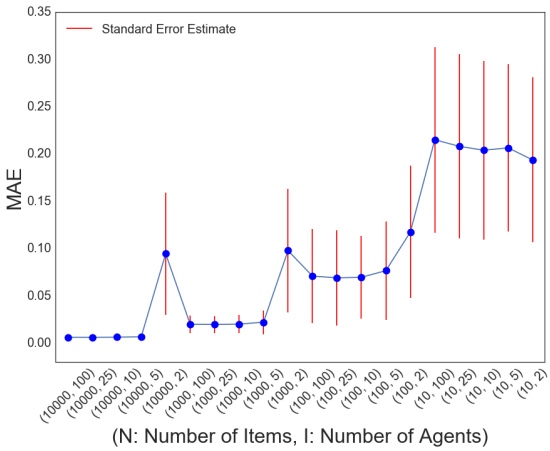
For each run, we sample a unique confusion matrix for agent i, by sampling it from a hyperparameter:

$$\Lambda = \begin{bmatrix} 10 & 1 & 1 \\ 1 & 10 & 1 \\ 1 & 1 & 10 \end{bmatrix}$$

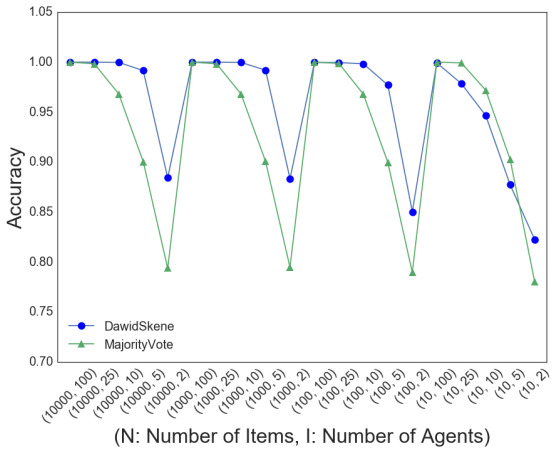
The m-th row of Θ^i, which is denoted as $\boldsymbol{\theta}_m^i$, is sampled from a Dirichlet distribution,

$$\boldsymbol{\theta}_m^i \sim Dirichlet(\lambda_m)$$

where λ_m denotes the m-th row of Λ. This generates I unique confusion matrices for each run.



(a) Mean Absolute Errors of $\hat{\Theta}$



(b) Accuracy Score of $\hat{t}$

Figure 3.4: PrivateConfusion Model: EM Performance

Figure 3.4 summarizes the accuracy of recovering the confusion matrices and

the true states using the EM algorithm in the PrivateConfusion model. In Figure 3.4a, similar to the outcome for the EM algorithm in the CommonConfusion model, we note the deterioration in performance of the EM algorithm in estimating the confusion matrices, as measured by MAE, along with the number of data points ($N \times I$). Also, similar to the outcome in the CommonConfusion model, the algorithm's performance is apparently more sensitive to the number of agents than the number of items.

In Figure 3.4, we observe the strong performance advantage in recovering the states using the EM algorithm versus the majority vote method. However, when the number of items declines to 10, the majority vote method outperforms the EM algorithm.

3.4. GroupConfusion Model

Consider a scenario where a computer scientist engaged in research in machine translation aims to build Japanese-English bilingual corpora via a human computation system with bilingual translators who are proficient in Japanese and English. The computer scientist suspects that there are three different type of translators — "Advanced", "Intermediate", and "Novice". He is interested in classifying each translator into one of the three types based on their error-rates and study the correlated attributes of the translators in each group.

In another example, let us consider a Massive Open Online Course (MOOC) in machine learning that is taken primarily by statistics and computer science students. We could imagine a student with the computer science background to generally perform better than those with statistics background in problems related to algorithms and data structures. We could also imagine that a student with statistics background to perform better than those with the computer science background in problems about Bayesian statistical models. We are interested in identifing the students' backgrounds based on the error-rates of the students and study correlated attributes such

as gender, race, age, and so forth within each group.

In this section, I introduce the GroupConfusion model. In this new model, an individual agent is a member of a group that shares similar confusion matrices. In Section 3.4.2, we introduce k-Means-Confusion, a variation of k-Means-- algorithm (Chawla & Gionis, 2013), to cluster the agents based on their confusion matrices.

3.4.1 Model

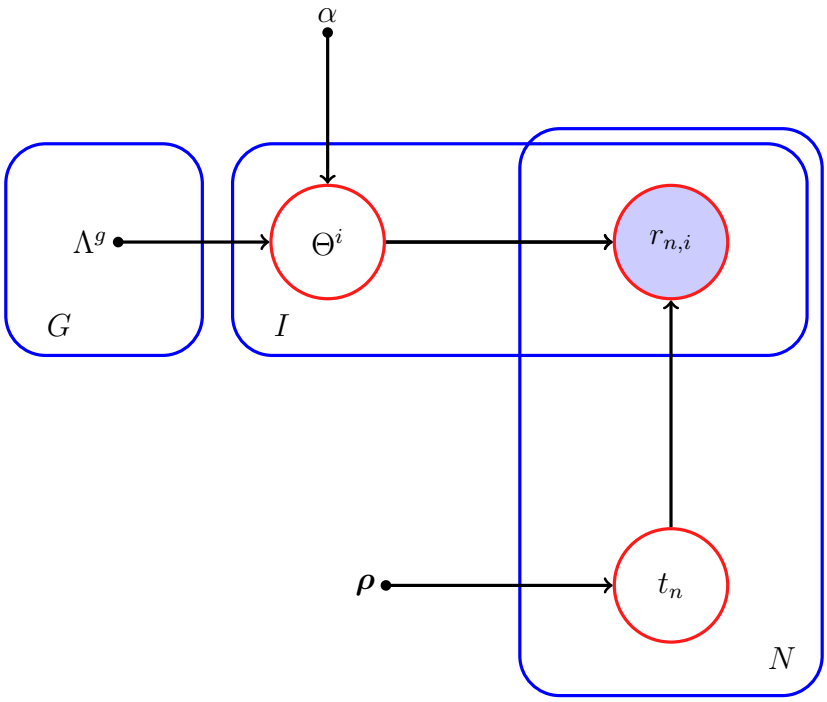


Figure 3.5: GroupConfusion Model

The GroupConfusion model is an extension of the PrivateConfusion model. There are G groups, and every agent $i \in \mathcal{I}$ belongs to a group $g \in \mathcal{G} = \{1, ..., G\}$. The group membership of agent i, which we denote as $\gamma_i = g$ is distributed according to a Multinomial distribution parameterized by hyperparameter $\alpha = \begin{bmatrix} \alpha_1 & \dots & \alpha_G \end{bmatrix}$, where $\sum_{g=1}^{G} \alpha_g = 1$:

$$\gamma_i \sim Multinomial(\alpha)$$

Collectively, $\Gamma = \{\gamma_1, ..., \gamma_I\}$, represents a set that denotes the membership of every agent $i \in \mathcal{I}$.

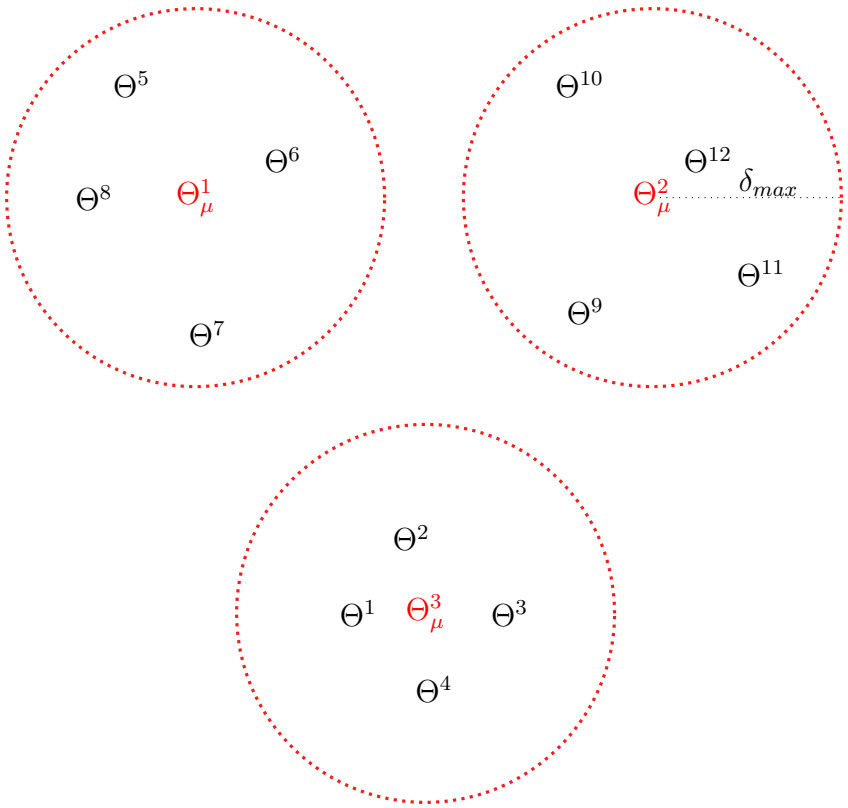


Figure 3.6: An Example of Agent Groups and Confusion Matrices

The confusion matrix of agent i belonging to group g is sampled from hyper-parameter:

$$\Lambda^g = \begin{bmatrix} \boldsymbol{\lambda}_1^g \\ \vdots \\ \boldsymbol{\lambda}_M^g \end{bmatrix} \tag{3.29}$$

$$\text{where } \boldsymbol{\lambda}_m^g = \begin{bmatrix} \lambda_1^g & \dots & \lambda_K^g \end{bmatrix} \text{ and } \sum_{j=1}^{K} \lambda_j^g = 1 \tag{3.30}$$

such that the m-th row of confusion matrix Θ^i, denoted as $\boldsymbol{\theta}_m^i$ is generated from a

Dirichlet distribution parameterized by $\boldsymbol{\lambda}_m^g$,

$$\boldsymbol{\theta}_m^i \sim Dirichlet(\boldsymbol{\lambda}_m^g)$$

For each group $g \in \mathcal{G}$, the agents' individual confusion matrices are clustered around the *central confusion matrix* Θ_μ^g. Collectively, $\boldsymbol{\Theta}_\mu = \{\Theta_\mu^1, ..., \Theta_\mu^G\}$ denotes a set of central confusion matrices for all groups. Figure 3.6 depicts an example of three groups of agents and the individual confusion matrices clustered around three central confusion matrices.

The graphical model representation in Figure 3.5 summarizes the GROUP-CONFUSION model.

3.4.2 Algorithm

In this section, we introduce the k-Means-Confusion (kMC) algorithm, a variant of the k-means-- algorithm (Chawla & Gionis, 2013), for clustering the agents in human computation systems using their confusion matrices. Given a set of confusion matrices $\boldsymbol{\Theta} = \{\Theta^1, ..., \Theta^I\}$, parameter G that specifies the number of groups, and an algorithm parameter δ_{max}, kMC returns the group membership of the agents $\Gamma = \{\gamma_1, ..., \gamma_I\}$ and the *central confusion matrices* $\boldsymbol{\Theta}_\mu = \{\Theta_\mu^1, ..., \Theta_\mu^G\}$.

Before we present kMC algorithm, we introduce a few important notations. We assume a distance function $d : \Theta \times \Theta \to \mathbb{R}$ defines the Euclidean distance between two confusion matrices,

$$d(\Theta, \Theta') = \frac{1}{M \cdot K} \sum_{h=1}^{M} \sum_{g=1}^{K} (\Theta_{h,g} - \Theta'_{h,g})^2 \tag{3.31}$$

Given a set of confusion matrices $\boldsymbol{\Theta} = \{\Theta^1, ..., \Theta^I\}$, $mean : \boldsymbol{\Theta} \to \Theta$ is a function that

defines the element-wise average matrix of the set,

$$mean(\mathbf{\Theta}) = \begin{bmatrix} \sum_{i=1}^{I} \Theta_{1,1}^{i} & \cdots & \sum_{i=1}^{I} \Theta_{1,K}^{i} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{I} \Theta_{M,1}^{i} & \cdots & \sum_{i=1}^{I} \Theta_{M,K}^{i} \end{bmatrix} \tag{3.32}$$

Finally, given $\mathbf{\Theta}_{\mu} = \{\Theta_{\mu}^{1}, ..., \Theta_{\mu}^{G}\}$, a set of G central confusion matrices, we define the distance of agent i's individual confusion matrix Θ^{i} to the closest central matrix as

$$d(\Theta^{i}|\mathbf{\Theta}_{\mu}) = \min_{\Theta_{\mu}^{g} \in \mathbf{\Theta}_{\mu}} d(\Theta^{i}, \Theta_{\mu}^{g}) \tag{3.33}$$

We describe the k-Means-Confusion algorithm below:

Algorithm 3.4.1 (K-means-Confusion)**.**
Given $\mathbf{\Theta}$*,* G*, and the maximum distance parameter* δ_{max}*,*

1. *Randomly assign group membership,* $\Gamma = \{\gamma_1, ..., \gamma_I\}$ *where* $\gamma_i \leftarrow g \in \mathcal{G}$ *for all* $i \in \mathcal{I}$*.*

2. *Based on* Γ*, compute the central matrices*

$$\mathbf{\Theta}^{g} \leftarrow \{\Theta^{i} \in \mathbf{\Theta}^{'} | \gamma_i = g\}, \ \forall i \in \mathcal{I}$$

$$\hat{\Theta}_{\mu}^{g} = mean(\mathbf{\Theta}^{g}), \ \forall g \in \mathcal{G}$$

3. *Iterate until convergence:*

    1 Initialize empty set of outliers : $\mathcal{O} \leftarrow \{\}$

    2 Compute $d(\Theta^{i}|\hat{\mathbf{\Theta}}_{\mu})$ $\forall i \in \mathcal{I}$

    3 If $d(\Theta^{i}|\hat{\mathbf{\Theta}}_{\mu}) > \delta_{max}$*:*

$$\mathcal{O} \leftarrow i$$

4 Exclude the outliers,

$$\mathbf{\Theta}^{'} \leftarrow \mathbf{\Theta} \setminus \{\Theta^i | \forall i \in \mathcal{O}\}$$

$$\mathcal{I}^{'} \leftarrow \mathcal{I} \setminus \mathcal{O}$$

5 For every $i \in \mathcal{O}$:

$$\gamma_i = NULL$$

6 For every $i \in \mathcal{I}^{'}$:

Assign Θ^i to set $\mathbf{\Theta}^g$ by finding the closest central matrix:

$$\gamma_i = \arg\min_{g \in \mathcal{G}} d(\Theta^i, \hat{\Theta}^g_\mu)$$

$$\mathbf{\Theta}^g \leftarrow \{\Theta^i \in \mathbf{\Theta}^{'} | \gamma_i = g\}$$

Compute:

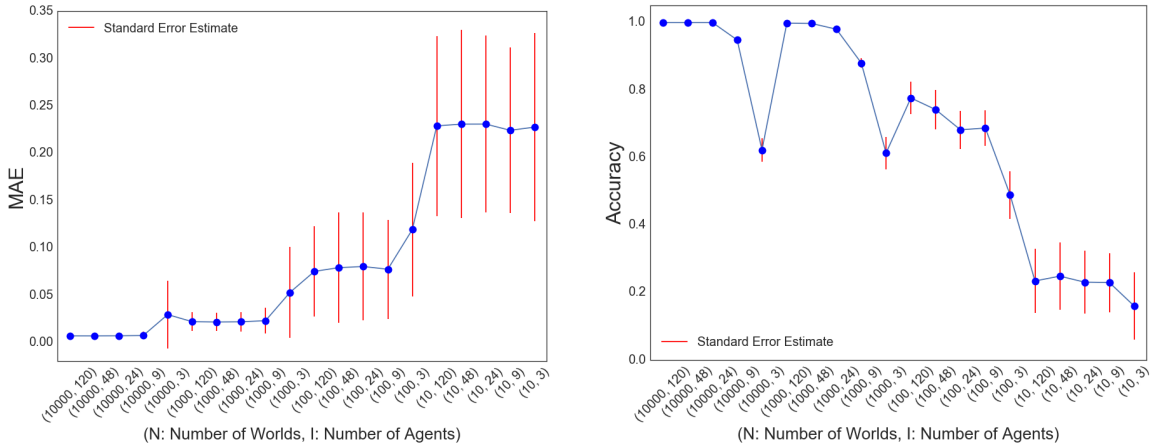
$$\hat{\Theta}^g_\mu = mean(\mathbf{\Theta}^g), \ \forall g \in \mathcal{G}$$

4. Return $\hat{\mathbf{\Theta}}_\mu$, Γ, and $\mathcal{O}$

3.4.3 Analysis

We examine the performance of the k-Means-Confusion algorithm using synthetic data.

We fix the set of states as $\mathcal{T} = \{1, 2, 3\}$ and the set of signals as $\mathcal{S} = \{1, 2, 3\}$. We vary the number of items, $N = (10, 100, 1000, 10000)$ and the number of agents, $I = (3, 9, 24, 48, 120)$. For each permutation of $(N \times I)$, we experiment on three different state priors — $\boldsymbol{\rho}_1 = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$, $\boldsymbol{\rho}_2 = \begin{bmatrix} 0.7 & 0.2 & 0.1 \end{bmatrix}$, and $\boldsymbol{\rho}_3 = \begin{bmatrix} 0.95 & 0.025 & 0.025 \end{bmatrix}$. We also generate three agent groups such that each group contains at least one agent. The three agent groups have the following hyperparameters



(a) Mean Absolute Errors of $\hat{\mathbf{\Theta}}$ **(b) Accuracy Score of $\hat{\Gamma}$**

Figure 3.7: k-Means-Confusion: Performance

for each group's confusion matrices:

$$\Lambda^0 = \begin{bmatrix} 98 & 1 & 1 \\ 1 & 98 & 1 \\ 1 & 1 & 98 \end{bmatrix} \quad \Lambda^1 = \begin{bmatrix} 76 & 23 & 1 \\ 12 & 76 & 12 \\ 12 & 12 & 76 \end{bmatrix} \quad \Lambda^2 = \begin{bmatrix} 50 & 25 & 25 \\ 25 & 50 & 25 \\ 25 & 25 & 50 \end{bmatrix}$$

We set the hyperparameter $\alpha = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$ and also fix the δ_{max} parameter of the algorithm to 0.2.

Figure 3.7a shows the MAE of the recovered individual matrices $\hat{\mathbf{\Theta}}$. As expected, the accuracy of the confusion matrix estimation deteriorates with decrease in the total number of data points. However, this deterioration in the accuracy of the confusion matrices estimation also impacts in the accuracy of group membership recovery.

In Figure 3.7b, we see near perfect accuracy when there are a large number of data points with a large number of items, $N \geq 1000$, and a large number of agents, $I \geq 9$. However, we see sharp decline in the accuracy of group membership recovery when the number of agents drops to 3, regardless of the number of items. Moreover, we also observe steep deterioration in accuracy as the number of item falls to $N \leq 100$.

Where the number of items is 10, the k-Means-Confusion algorithm performs quite poorly.

This experiment shows that the k-Means-Confusion algorithm works best when there's a sufficiently large number of data points generated from a large number of agents. Since the k-Means-Confusion algorithm is reliant on the EM algorithm to estimate the confusion matrices, if the accuracy of that estimation is poor, we should expect the k-Means-Confusion algorithm to perform poorly as well.

3.5. Extensions

Since the introduction of the PrivateConfusion model by Dawid and Skene, researchers in computer science and statistics have proposed new models that build on the PrivateConfusion model.

One of the earliest work is the *Bayesian classifier combination method* (BCC) by Ghahramani and Kim (2003). BCC uses Bayesian hierarchical model to add deeper complexity to the original *PrivateConfusion* model. Similarly, Liu and Wang (2012) introduce the HybridConfusion model in the same paper that introduces the CommonConfusion model. Finally, perhaps most similar to the GroupConfusion model, there is the JointConfusion model (Lakkaraju et al., 2015), which assigns the agents to clusters based on various other attributes besides the error-rates. These models are Bayesian statistical models that require the Markov Chain Monte Carlo (MCMC) sampling methods, which are computationally expensive.

To our knowledge, the GroupConfusion model is the only new model since the PrivateConfusion model in which the estimation of the confusion matrices and clustering of the agents can be done with the EM algorithm and k-Means-Confusion algorithm.

Chapter 4: Emprical Peer Prediction Mechanisms

Originally, peer prediction mechanisms were proposed as mechanisms to elicit honest feedback from reviewers in online recommendation systems. Since then the researchers in the information elicitation field have found natural application of peer prediction mechanisms in human computation systems. However, real world application has been limited to simple mechanisms such as the Output Agreement mechanism, and application of more theoretically complex mechanisms such as the classical Peer Prediction Method has faced barriers due to their restrictive assumptions (Waggoner & Chen, 2013).

Independent of the research in peer prediction mechanisms, researchers in machine learning and statistics have used different models and algorithms to quantify the error-rates of the workers in human computation systems. They have been interested in quantifying the error-rates of the workers because knowing the error-rates offers the system designer the choice of excluding the reports of the highly error-prone workers or excluding those workers from future tasks. However, their models have largely ignored the incentives of the human workers.

While they developed independently of one another, two fields of research ultimately share the same goal, which is to improve human computation systems. One of the leading researchers in peer prediction mechanisms, Jens Witkowski stated, "... integrating peer prediction mechanisms with machine learning models will be mutually beneficial" (Witkowski, 2014). I believe that the two fields are for ripe for marriage today.

In this chapter, I introduce an unified approach that brings together the mod-

els and techniques of the two fields. By integrating the error-rate models from the machine learning literature with the incentive models of the peer prediction mechanism in the information elicitation literature, we design a new class of peer prediction mechanisms, which I call *empirical peer prediction mechanisms*.

In Section 4.1, we explore the CommonBelief model and the *Empirical Peer Prediction Method*. Miller et al. who introduced the classical Peer Prediction Method left open the task of estimating the state priors and the conditional signal probabilities of the common belief model as a future direction of peer prediction research (Miller et al., 2005); I directly address this open challenge with the CommonBelief model and the Empirical Peer Prediction Method.

In Section 4.2, we explore the GroupBelief model, which addresses the systematic differences in "tastes" among the agents. Miller et al. stated that the systematic differences in the population of agents should be modeled explicitly, and they also left this task open as a future research opportunity. I introduce the *k-Means Peer Prediction Method* as an answer to this open challenge.

Finally, in Section 4.3, we further relax the homogeneity assumption and introduce the PrivateBelief model where every individual agent has her own unique biases and capabilities. I introduce a new peer prediction mechanism called the *Empirical Scoring Rule Mechanism* that correctly incentivizes agents in this model.

For each mechanism, we analyze its properties using simulated data. I demonstrate empirically that given a sufficiently large amount of data, empirical peer prediction mechanisms are truthful peer prediction mechanisms and are robust against various reporting strategies including collusion among the agents.

4.1. Empirical Peer Prediction Method

One of the hurdles in implementing the classical Peer Prediction Method (CPPM) in a large-scale human computation system is its restrictive requirement that the mechanism designer must know the commonly shared belief model of the

agents. (Waggoner & Chen, 2013)

In their original paper that introduces CPPM, Miller et al. briefly discuss that the mechanism designer should use historical reports submitted by the agents to compute the state priors and the conditional signal probabilities. However, they do not discuss this topic in detail and instead leave it open as a future research opportunity in peer prediction mechanisms.

In this section, I introduce the *Empirical Peer Prediction Method* (EPPM) as an answer to the challenge left open by Miller et al. EPPM incorporates the EM algorithm to compute the commonly shared belief model of the agents and deploys payment rule similar to that of CPPM. Therefore, I view EPPM as a product of marriage between information elicitation and machine learning.

In this section, I also introduce a new scoring rule that is useful in the empirical setting to maintain the robustness of EPPM against collusion among the agents.

4.1.1 Model

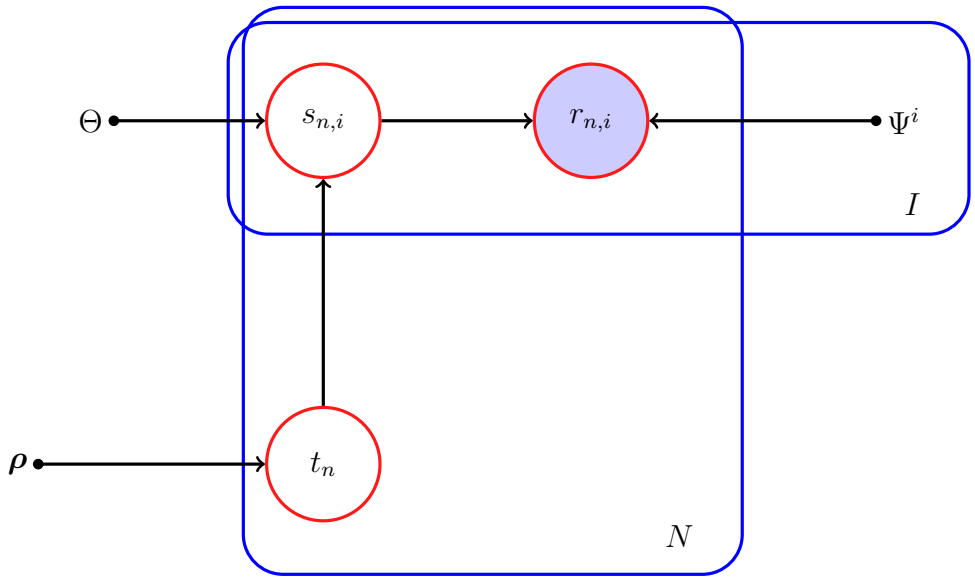


Figure 4.1: CommonBelief Model

The CommonBelief model is closely related to the base model of peer pre-

diction mechanism and the CommonConfusion model. We model N items with random variable $T_n = t_n \in \mathcal{T}$ for $n \in (1, ..., N)$ and $N \times I$ privately observed signals with random variable $S_{n,i} = s_{n,i} \in \mathcal{S}$ for agent i and $n \in (1, ..., N)$.

The state $t_n \in \mathcal{T}$ for every item $n \in (1, ..., N)$ is distributed independently and identically according to state prior $P(T_n = t_n)$ similar to the base model of peer prediction; however, we adopt the notation from the machine learning literature to denote the state priors as $\boldsymbol{\rho} = \begin{bmatrix} \rho_1 & \dots & \rho_M \end{bmatrix}$ where $\rho_m = P(T_n = m),\ \forall m \in \mathcal{T},\ \forall n \in (1, ..., N)$.

In the base model of peer prediction, the conditional signal probability $P(S = k|T = m)$, the probability that an agent observes signal $k \in \mathcal{S}$ given that the true state of the world is $m \in \mathcal{T}$, is not "explicitly" modeled as an error-rate of the observation. In the CommonBelief model, we explicitly model the conditional signal probabilities as the error-rates of observation, and altogether, they are represented by $M \times K$ confusion matrix Θ:

$$\Theta = \begin{bmatrix} \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_M \end{bmatrix} \tag{4.1}$$

where the m-th row represents a probability vector $\boldsymbol{\theta}_m = \begin{bmatrix} \theta_{m,1} & \dots & \theta_{m,K} \end{bmatrix}$, and each element $\theta_{m,k} = P(S_{n,i} = k|T_n = m),\ \forall n \in (1, ..., N),\ \forall i \in \mathcal{I}$ such that $\sum_{g=1}^{K} \theta_{m,g} = 1$. Given the confusion matrix Θ, agent i's observed signal of the n-th item, $s_{n,i}$, is distributed according a Multinomial distribution parameterized by $\boldsymbol{\theta}_m$ conditioned on $t_n = m$,

$$s_{n,i} \sim Multinomial(\boldsymbol{\theta}_m)$$

The representation of the conditional signal probabilities with confusion matrix is relevant where the set of states $\mathcal{T}$ and the set of signals $\mathcal{S}$ have 1-to-1 relationship. For example, in a typical human computation system the signals represent the noisy observations about the true states.

However, the CommonBelief model does not necessarily exclude settings

where the set of states $\mathcal{T}$ and the set of signals $\mathcal{S}$ do not have 1-to-1 relationship. For example, consider an online recommendation system of restaurant where the true possible state of a restaurants is "High Quality" or "Low Quality". Given a state, a reviewer observes a signal from a star rating in a scale of 1 to 5. While we do not refer to the conditional signal probabilities as error-rates or confusion matrix in such setting, the CommonBelief model is still able to represent the conditional signal probabilities as 2×5 matrix.

In the CommonBelief model, unlike the existing models in the machine learning literature, we explicitly model the reporting strategies of the agents. For every agent $i \in \mathcal{I}$, her reporting strategy is represented by $K \times K$ *strategy matrix*:

$$\Psi^i = \begin{bmatrix} \boldsymbol{\psi}_1^i \\ \vdots \\ \boldsymbol{\psi}_K^i \end{bmatrix} \tag{4.2}$$

$$\boldsymbol{\psi}_k^i = \begin{bmatrix} \psi_{k,1}^i & \dots & \psi_{k,K}^i \end{bmatrix} \tag{4.3}$$

For all $k \in \mathcal{S}$, $\boldsymbol{\psi}_k^i$ represents a probability distribution such that $\sum_{g=1}^{K} \psi_{k,g}^i = 1$, and $\psi_{k,h}^i$ represents $P(r_{n,i} = h | S_{n,i} = k)$, the probability of agent i reporting h given that she observes signal k in item n. Therefore, a report $r_{n,i}$ is distributed according to a Multinomial distribution parameterized by $\boldsymbol{\psi}_k^i$ conditioned on signal $s_{n,i} = k$.

$$r_{n,i} \sim Multinomial(\boldsymbol{\psi}_k^i)$$

The truthful reporting strategy is represented by $K \times K$ identity matrix,

$$\Psi^i = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{4.4}$$

We adopt the graphical model representation from the machine learning literature to summarize the CommonBelief model in Figure 4.1.

4.1.2 Mechanism

In this section, I introduce the *Empirical Peer Prediction Method* (EPPM). Unlike CPPM, EPPM does not require that the designer know *a priori* the belief model, $\boldsymbol{\rho}$ and Θ. These probabilities are estimated from the agent reports $\boldsymbol{r}$ using the EM algorithm. Once the mechanism estimates the belief model, it computes the payments to the agents in a manner similar to that of CPPM. However, we should note that EPPM withholds payments until all reports are submitted by the participating agents.

Suppose that agent i chooses not to report truthfully. If the mechanism utilizes all the agents' reports $\boldsymbol{r}$, including the reports of agent i, to compute the estimated confusion matrix $\hat{\Theta}$ and the estimated state priors $\hat{\boldsymbol{\rho}}$ using the EM algorithm of the CommonConfusion model, then the reports by agent i may obfuscate the estimates. Therefore, for each agent $i \in \mathcal{I}$, EPPM uses the reports of agent i's peers, $\boldsymbol{r}_{-i}$, to compute $\hat{\Theta}$ and $\hat{\boldsymbol{\rho}}$.

If agent i's peers all report truthfully such that strategy matrices of the peers $\boldsymbol{\Psi^{-i}}$ are the identity matrices, then because all agents share the same Θ and $\boldsymbol{\rho}$, the mechanism is able to simplify the CommonBelief model to the CommonConfusion model that we discussed in Chapter 3 because

$$\Psi^j \cdot \Theta = \Theta, \ \forall j \in \{-i\} \tag{4.5}$$

Thus, applying the EM algorithm of the CommonConfusion model, EPPM can accurately compute $\hat{\Theta}_i \approx \Theta$ and $\hat{\boldsymbol{\rho}}_i \approx \boldsymbol{\rho}$ given a sufficiently large numbers of items and peer agents. Note that we use the subscript $_i$ to indicate that $\hat{\Theta}_i$ and $\hat{\boldsymbol{\rho}}_i$ are not the unique belief model of agent i. This is not to be confused with the superscript in the PrivateConfusion model in Chapter 3 Section 3.3 where Θ^i represents the

unique confusion matrix of agent i.

From the estimates of Θ and $\boldsymbol{\rho}$, the mechanism can compute the signal priors $\boldsymbol{\phi}$ and signal posteriors $\boldsymbol{\vartheta}$. The signal priors are represented by a probabiliity vector,

$$\boldsymbol{\phi} = \begin{bmatrix} \phi_1 & \dots & \phi_K \end{bmatrix} \tag{4.6}$$

$$\text{where } \phi_k = P(S_i = k) = \sum_{h=1}^{M} P(S_i = k | T = h) P(T = h) \tag{4.7}$$

The signal posteriors are represented by $K \times K$ matrix,

$$\boldsymbol{\vartheta} = \begin{bmatrix} \boldsymbol{\vartheta}_1 \\ \vdots \\ \boldsymbol{\vartheta}_K \end{bmatrix} \tag{4.8}$$

where $\boldsymbol{\vartheta}_k = \begin{bmatrix} \vartheta_{k,1} & \dots & \vartheta_{k,K} \end{bmatrix}$ is a probability vector such that $\sum_{g=1}^{K} \vartheta_{k,g} = 1$, and

$$\vartheta_{k,g} = P(S_{n,j} = g | S_{n,i} = k) = \sum_{h=1}^{M} \Theta_{h,g} \cdot \frac{\Theta_{h,k} \cdot \rho_h}{\phi_k} \tag{4.9}$$

Recall that CPPM pays agent i using a strictly proper scoring rule R:

$$x_i(r_i, r_j) = R(\boldsymbol{\vartheta}_{r_i}, r_j) \tag{4.10}$$

Borrowing the idea from market scoring rule (Hanson, 2007), I introduce a new payment rule for EPPM, which to my knowledge has not been used in a peer prediction mechanism.

$$x_i(r_{n,i}, r_{n,j}) = R(\boldsymbol{\vartheta}_{r_{n,i}}, r_{n,j}) - R(\boldsymbol{\phi}, r_{n,j}) \tag{4.11}$$

In words, the new payment rule is the difference between the score of the signal posteriors of agent i conditioned on her report $r_{n,i}$ for item n and the score from signal prior.

We show that the new payment rule is strictly proper with a simple example. Consider the following signal posteriors:

$$\boldsymbol{\vartheta} = \begin{bmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{bmatrix}$$

and signal priors:

$$\boldsymbol{\phi} = \begin{bmatrix} 0.35 & 0.65 \end{bmatrix}$$

Suppose that agent i observes signal $S_{n,i} = 0$ for item n. If she reports truthfully, the expected payment to agent i using the quadratic scoring rule is

$$\begin{aligned} &P(S_{n,j} = 0 | S_{n,i} = 0) \cdot \{R_Q(\boldsymbol{\vartheta}_0, 0) - R_Q(\boldsymbol{\phi}, 0)\} \\ &+ P(S_{n,j} = 1 | S_{n,i} = 0) \cdot \{R_Q(\boldsymbol{\vartheta}_0, 1) - R_Q(\boldsymbol{\phi}, 1)\} \\ &= 0.6 \cdot 0.525 + 0.4 \cdot -0.475 \\ &= 0.125 \end{aligned}$$

If she reports $r_{n,i} = 1$ instead, her expected payment is

$$\begin{aligned} &P(S_{n,j} = 0 | S_{n,i} = 0) \cdot \{R_Q(\boldsymbol{\vartheta}_1, 0) - R_Q(\boldsymbol{\phi}, 0)\} \\ &+ P(S_{n,j} = 1 | S_{n,i} = 0) \cdot \{R_Q(\boldsymbol{\vartheta}_1, 1) - R_Q(\boldsymbol{\phi}, 1)\} \\ &= 0.6 \cdot -0.135 + 0.4 \cdot 0.065 \\ &= -0.055 \end{aligned}$$

Agent i is better off reporting truthfully to signal $S_{n,i} = 0$. On the other hand, if she

observes $S_{n,i} = 1$ and reports truthfully, then her expected payment is

$$\begin{aligned}
&P(S_{n,j} = 0|S_{n,i} = 1) \cdot \{R_Q(\boldsymbol{\vartheta}_1, 0) - R_Q(\boldsymbol{\phi}, 0)\} \\
&+ P(S_{n,j} = 1|S_{n,i} = 1) \cdot \{R_Q(\boldsymbol{\vartheta}_1, 1) - R_Q(\boldsymbol{\phi}, 1)\} \\
&= 0.3 \cdot -0.135 + 0.7 \cdot 0.065 \\
&= 0.005
\end{aligned}$$

If she reports $r_{n,i} = 0$, then her expected payment is

$$\begin{aligned}
&P(S_{n,j} = 0|S_{n,i} = 1) \cdot \{R_Q(\boldsymbol{\vartheta}_0, 0) - R_Q(\boldsymbol{\phi}, 0)\} \\
&+ P(S_{n,j} = 1|S_{n,i} = 1) \cdot \{R_Q(\boldsymbol{\vartheta}_0, 1) - R_Q(\boldsymbol{\phi}, 1)\} \\
&= 0.3 \cdot 0.525 + 0.7 \cdot -0.475 \\
&= -0.175
\end{aligned}$$

For either signal, agent i is better off reporting truthfully. This result is also true in settings where the number of signals $K > 2$.

Mechanism 4.1.1 (**Empirical Peer Prediction Method).** *The Empirical Peer Prediction Method is defined as:*

1. *Mechanism receives reports $r_{n,i} \in \mathcal{S}$ from every agent $i \in \mathcal{I}$ for every item $n \in (1, ..., N)$.*

2. *For every agent $i \in \mathcal{I}$, estimates the confusion matrix $\hat{\Theta}_i$ and the state priors $\hat{\boldsymbol{\rho}}_i$ from $\boldsymbol{r}_{-i}$, reports of agent i's peers $\{-i\}$ using the EM algorithm of the* CommonConfusion *model under the truthful assumption.*

3. *Pays agent i for item n a payment based on $x_i(r_{n,i}, r_{n,j}) = R(\hat{\boldsymbol{\vartheta}}_{i_{r_{n,i}}}, r_{n,j}) - R(\hat{\boldsymbol{\phi}}_i, r_{n,j})$, where $r_{n,j}$ is the report from a reference agent j selected from $\{-i\}$ and R is a strictly proper scoring rule.*

We demonstrate EPPM in practice with an example.

Example 4. *Let us return to the example introduced in Chapter 2. In Example 3, the computer scientist adopted CPPM, and he luckly managed to accurately compute the true state priors and the conditional signal probabilities, which were*

$$\boldsymbol{\rho} = \begin{bmatrix} 0.3 & 0.7 \end{bmatrix}$$

$$\Theta = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}$$

Let us suppose that the computer scientist does not have knowledge of the belief model in advance. Now, he would not be able to implement CPPM in his human computation system, but instead, he can adopt EPPM. He segments the total data set of images into batches of 1,000 images. He withholds the payments until all workers submit their labels for all the images in each batch. Once he receives all the reports, he pays his workers accordingly using EPPM.

For worker i, EPPM computes the belief model based on her peer's reports, which is

$$\hat{\boldsymbol{\rho}}_i = \begin{bmatrix} 0.29 & 0.71 \end{bmatrix}$$

$$\hat{\Theta}_i = \begin{bmatrix} 0.69 & 0.31 \\ 0.22 & 0.78 \end{bmatrix}$$

From these estimates, EPPM computes $\hat{\boldsymbol{\phi}}_i$ and $\hat{\boldsymbol{\vartheta}}_i$,

$$\hat{\boldsymbol{\phi}}_i = \begin{bmatrix} 0.36 & 0.64 \end{bmatrix}$$

$$\hat{\boldsymbol{\vartheta}}_i = \begin{bmatrix} 0.48 & 0.52 \\ 0.28 & 0.72 \end{bmatrix}$$

Based on these estimated probabilities, EPPM computes the payment schedule for worker i using the quadratic scoring rule, which is shown in Table 4.1.

For item n, suppose that worker i observed $S_{n,i} = 0$. If she reported truthfully,

r_i \ r_j	0	1
0	0.28	-0.20
1	-0.21	0.10

Table 4.1: Example 4 EPPM Payment Matrix

then worker i's expected payment for item n would be

$$\begin{aligned} & P(S_{n,j} = 0 | S_{n,i} = 0) \cdot 0.28 + P(S_{n,j} = 1 | S_{n,i} = 0) \cdot -0.20 \\ \approx\ & 0.43 \cdot 0.28 + 0.57 \cdot -0.20 \\ \approx\ & 0.0064 \end{aligned}$$

In contrast, if she reported $r_{n,i} = 1$, then her expected payment would be

$$\begin{aligned} & P(S_{n,j} = 0 | S_{n,i} = 0) \cdot -0.21 + P(S_{n,j} = 1 | S_{n,i} = 0) \cdot 0.10 \\ \approx\ & 0.43 \cdot -0.21 + 0.57 \cdot 0.1 \\ \approx\ & -0.0333 \end{aligned}$$

For the observed signal $S_{n,i} = 0$, worker i would be better off reporting truthfully.

Instead, suppose that worker i observed $S_{n,i} = 1$ for item n. If she reported truthfully, then her expected payment would be

$$\begin{aligned} & P(S_{n,j} = 0 | S_{n,i} = 1) \cdot -0.21 + P(S_{n,j} = 1 | S_{n,i} = 1) \cdot 0.10 \\ \approx\ & 0.24 \cdot 0.21 + 0.76 \cdot 0.10 \\ \approx\ & 0.0256 \end{aligned}$$

However, if she reported $r_{n,i} = 0$, then her expected payment would be

$$\begin{aligned} & P(S_{n,j} = 0 | S_{n,i} = 1) \cdot 0.28 + P(S_{n,j} = 1 | S_{n,i} = 1) \cdot -0.20 \\ \approx\ & 0.24 \cdot 0.28 + 0.76 \cdot -0.20 \\ \approx\ & -0.0848 \end{aligned}$$

Therefore, worker i would be better off reporting truthfully in either signal observation.

Although a recently introduced peer prediction mechanism, *Multi-task 01-Mechanism* (Shnayder et al., 2016), is robust against uniformative equilibria, most peer prediction mechanisms that have been introduced in the information eliciation literature cannot avoid the problem of the uninformative equilibria (Waggoner & Chen, 2013). However, EPPM is robust against uninformative equilibria. This property is due to EPPM's empirical estimation of the belief models using the EM algorithm and the new payment rule. We demonstrate this property of EPPM with an example.

Example 3 (continued)**.** *We return to the example at the end of Chapter 2 where the computer scientist's human computation system is under attack by collusion of multiple workers. Multiple workers have coordinated to label every image as 'cat' regardless of what they observe, and they are maximizing their payments this way.*

The computer scientist implements EPPM using the quadratic scoring rule in hopes of thwarting the collusion attack. After receving the reports on the first $1,000$ *images, he finds that the estimate of the state priors and the conditional signal probabilities for all workers are the exactly same:*

$$\hat{\boldsymbol{\rho}}_i = \begin{bmatrix} 0.0 & 1.0 \end{bmatrix}$$

$$\hat{\Theta}_i = \begin{bmatrix} 0.0 & 1.0 \\ 0.0 & 1.0 \end{bmatrix}$$

As a result, the estimated signal priors and signal posteriors are:

$$\hat{\boldsymbol{\phi}}_i = \begin{bmatrix} 0.0 & 1.0 \end{bmatrix}$$

$$\hat{\boldsymbol{\vartheta}}_i = \begin{bmatrix} NaN & NaN \\ 0.0 & 1.0 \end{bmatrix}$$

Consequently, the payment to each worker is

$$\begin{aligned} x(1,1) &= R_Q((0.0, 1.0), 1) - R_Q((0.0, 1.0), 1) \\ &= 1.0 - 1.0 = 0.0 \end{aligned}$$

While the workers involved in the collusive attack wasted the computer scientist's time, they did not inflict any financial cost. By adopting EPPM, the computer scientist successfully thwarted the collusive attack.

As demonstrated in Example 3 with binary signals, if all agents collude to report the same signal repeatedly, then the estimates of the signal posteriors and the signal priors will be such that the scores from two strictly proper scoring rule will cancel each other out. This results in 0 payments for all agents. This result is also true in settings where $K > 2$ as we shall see with simulated data in the following section.

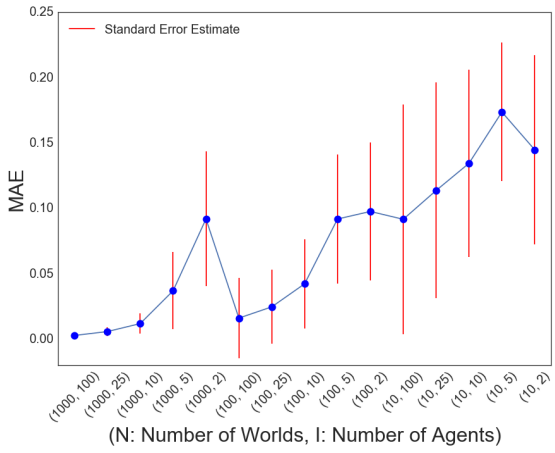
4.1.3 Analysis

In this section, we empirically demonstrate properties of EPPM.

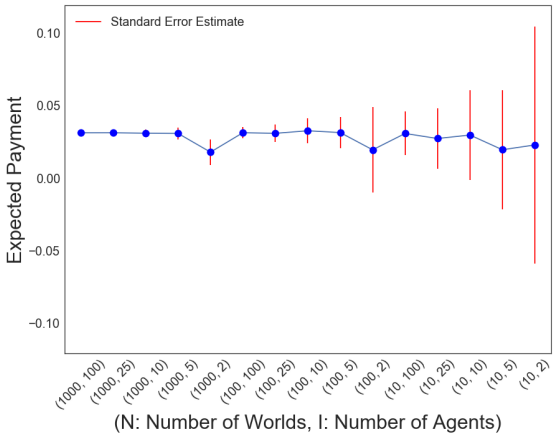
Performance of the EM Algorithm and the Expected Payment of EPPM

We examine the expected payment to a truthfully reporting agent in EPPM using simulated data. This experiment is similar to the experiment for the Common-Confusion model in Section 3.2.3. However, our primary interest in this section is the expected payment of EPPM with respect to varying number of data.

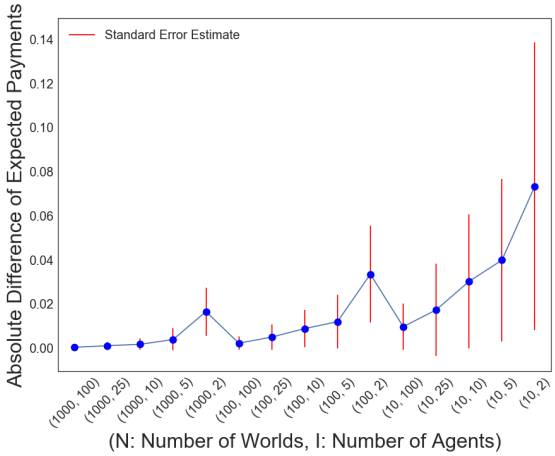
We fix the states to $\mathcal{T} = \{0, 1, 2\}$ and the signals to $\mathcal{S} = \{0, 1, 2\}$. We vary the number of items, $N = (10, 100, 1000)$ and the number of agents, $I = (2, 5, 10, 25, 100)$. For each permutation of $(N \times I)$, we experiment on three different state priors — $\boldsymbol{\rho}_1 = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$, $\boldsymbol{\rho}_2 = \begin{bmatrix} 0.7 & 0.2 & 0.1 \end{bmatrix}$, and $\boldsymbol{\rho}_3 = \begin{bmatrix} 0.95 & 0.025 & 0.025 \end{bmatrix}$. For each permutation and state priors, we run 50 experiments and report on the average results. Throughout the experiment, we create synthetic agents using the following



(a) MAEs of $\hat{\Theta}$



(b) Expected Payment of EPPM



(c) Absolute Difference CPPM vs EPPM

Figure 4.2: Empirical Peer Prediction Method: Accuracy

single, common confusion matrix,

$$\Theta = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0.15 & 0.15 & 0.7 \end{bmatrix}$$

In Figure 4.2a, we observe that the accuracies of recovering Θ, as measured by MAE, exhibits similar trend to what we observed in Section 3.2.3.

Figure 4.2b shows the average expected payments of a truthful agent in EPPM. As the performance of the EM algorithm suffers from shortage of data, we see increased uncertainty in the expected payment of EPPM as measured by the standard error.

If the mechanism designer can perfectly estimate the belief model, or if the designer has advance knowledge of the belief model, then the expected payments to the agents should be the same as that of CPPM with the new payment rule. Therefore, we set the expected payments of CPPM as the benchmark to measure the accuracies of expected payments in EPPM.

Figure 4.2c shows a plot of the absolute difference in the expected payments between these two mechanisms as defined by :

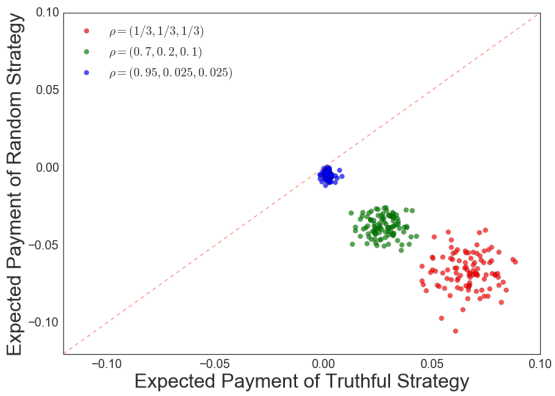
$$|\mathbb{E}_{EPPM}[x_i(s_i, s_j)|S_i = s_i] - \mathbb{E}_{CPPM}[x_i(s_i, s_j)|S_i = s_i]|$$

We observe that when there are a large number of data points, $N \geq 100$ and $I \geq 5$, the expected payments of EPPM do not deviate far from those of CPPM. However, as the number of data points decreases, the accuracy of EPPM's expected payment deteriorates in a trend similar to that of MAEs of $\hat{\Theta}$.

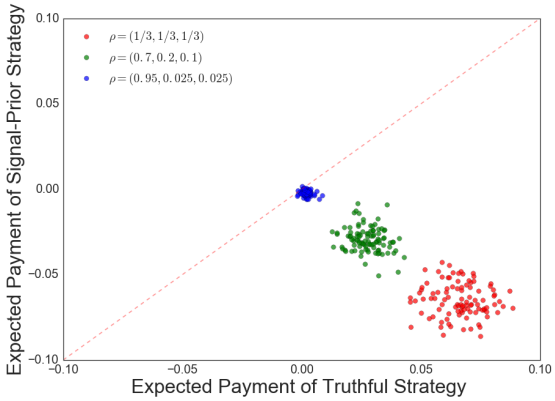
Robustness of EPPM against Strategic Reporting

In this experiment, we compare the expected payments of the truthful reporting strategy and those of other reporting strategies in EPPM.

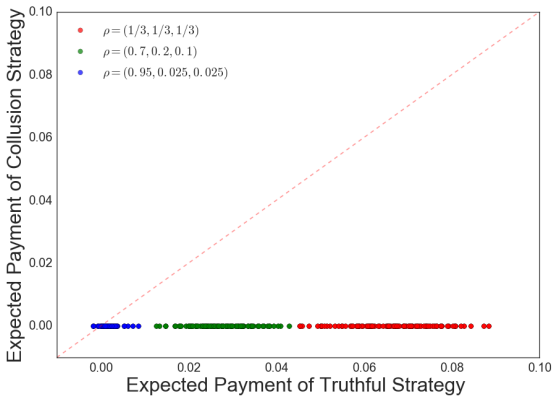
We maintain the same parameters — the states, the signals, and the common confusion matrix — as the previous experiment. From the experiment above, we note that $N \geq 1000$ and $I \geq 10$ generates sufficiently large data points to yield accurate expected payments in EPPM; therefore, in this experiment, we fix the number of items to $N = 1000$ and the total number agents to $I = 12$. We also run the experiment on the three different state priors that were introduced in the previous experiment. For each state prior, we perform 100 simulation runs.



(a) Truthful Strategy vs Random Reporting Strategy



(b) Truthful Strategy vs Signal-Prior Reporting Strategy



(c) Truthful Strategy vs Collusion Strategy

Figure 4.3: Empirical Peer Prediction Method: Robustness

First, we examine the expected payment of the random reporting strategy. Under this strategy, the agent reports any signal with 1/3 probability regardless of her observation. For each run, we first make all twelve agents report truthfully and compute their expected payments. Afterwards, while all other agents stay truthful, we select one agent to adopt the random reporting strategy and run the simulation again in the same environment and compute her expected payment under the new strategy.

Figure 4.3a shows a scatter plot of the expected payments to an agent when she reports truthfully versus the expected payments to the same agent when she reports

randomly. Each point in the plot indicates the expected payment in one simulation, showing the truthful reporting payments along the x-axis and the random reporting payments along the y-axis. A point lying below the dotted red line indicates that the expected payment is higher under the truthful reporting strategy; likewise, a point lying above the red line indicates that the random reporting strategy has the higher expected payment. Figure 4.3a shows that in all three state priors, agents are better off reporting truthfully.

Second, we examine the truthful reporting strategy against a reporting strategy that we call *Signal-prior strategy*. Under the Signal-prior strategy, the agent keeps track of the number of signal observations and builds the signal priors based on the historical counts. In each round, agent reports randomly based on the signal priors that she computed before that round. In Figure 4.3b, we plot the expected payments of the truthful reporting strategy against that of the Signal-prior strategy. Similar to the outcomes from the random reporting strategy, the expected payments under the Signal-prior strategy are almost all below those of the truthful reporting strategy.

Finally, we examine the expected payments under the collusion strategy in which all twelve agents coordinate to report the same signal repeatedly. We repeat the experiment for all three signals and plot the expected payments versus that of the truthful reporting strategy in Figure 4.3c. As shown in Example 3 in Section 4.1.2, the expected payments for every agent under this strategy is zero. Hence, in almost all cases, the agents are better off by reporting truthfully with a few exceptions when the state priors are heavily imbalanced. In the case where the state priors are $\boldsymbol{\rho} = \begin{bmatrix} 0.95 & 0.025 & 0.025 \end{bmatrix}$, the truthful reporting strategy may yield lower expected payment because the errors in signal observation can cause the agent to report 1 and 2 more frequently than desired.

4.2. k-Means Peer Prediction Method

In the original paper that introduces CPPM, Miller et al. state that the agents may exhibit systematic differences and if so, the mechanism designer must explicitly model the different types of agents. They suggest that the mechanism designer model the different types of agent based on the historical reports; however, Miller et al. do not go into a detailed discussion about how the designer should model the different type of agents or suggest how the designer should identify each agent's type. Instead, they leave these details as future research opportunities.

In this section, I relax the assumption of homogeneity of agents and introduce the GroupBelief model, which models systematically different type of agents. I also introduce the *k-Means Peer Prediction Method* (kPPM) as an answer to the challenge left open by Miller et al. in such a model.

Modeling the different types of agent may be useful in designing payment rule that favors a certain type of agents over the others. For example, in Chapter 3 Section 4.2.1, I presented an example of human computation system that builds bilingual corpora. The system designer suspects that there are three type of translators —"Advanced," "Intermediate," and "Novice". Once the types are identified, the designer can reward the Advanced translators with higher expected payments. This can incentivize the participants to not only report truthfully about their observed signals but also induce them to make more effort in the signal observation so that the mechanism may classify them as an Advanced translator.

4.2.1 Model

The GroupBelief model is an extension of the base model of peer prediction mechanisms and the GroupConfusion model introduced in Chapter 3 Section 3.4. The graphical model representation in Figure 4.4 summarizes the GroupBelief model.

We note that in the GroupBelief model the mechanism designer does not

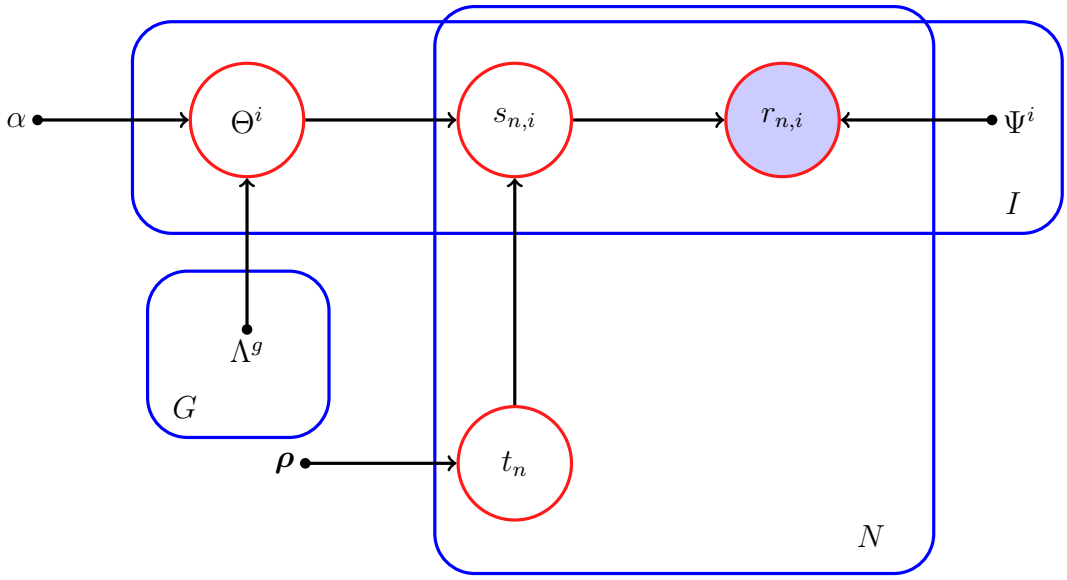


Figure 4.4: GroupBelief Model

know in advance the belief models of each group. In addition, the designer also does not know Γ, the group membership of the agents.

4.2.2 Mechanism

Recall that EPPM uses reports of agent i's peers, $\boldsymbol{r}_{-i}$ to estimate Θ and $\boldsymbol{\rho}$. If agent i's peers report truthfully, this approach enables EPPM to accurately estimate the belief model for agent i even if she does not report her signals truthfully.

Similarly, in the GroupBelief model, if the group membership of the agents Γ is known to the mechanism, then with a sufficiently large number of data points, the mechanism can accurately estimate Θ^g for every $g \in \mathcal{G}$ using the reports of agents' group peers. In other words, the designer can partition the agents based on the group membership and run separate EPPM on each group.

However, in the GroupBelief model, the mechanism designer does not know the group memberships in advance. As a result, unless the designer trusts that all agents reported truthfully, the mechanism cannot reliably estimate Θ^g. As a solution, kPPM does not rely on the estimate of individual confusion matrix $\hat{\Theta}^i$ to compute

the payments to agent i. Instead, the mechanism estimates the centeral confusion matrices $\hat{\mathbf{\Theta}}_\mu = \{\hat{\Theta}_\mu^1, ..., \hat{\Theta}_\mu^G\}$ using the k-Means-Confusion algorithm introduced in Section 3.4.

Once kPPM estimates the central confusion matrices of the groups, it is able to also compute the signal priors $\boldsymbol{\phi}^g$ and signal posteriors $\boldsymbol{\vartheta}^g$ of every group $g \in \mathcal{G}$. With these estimates as well as inferred group membership information, kPPM proceeds to compute the payments for the agents in each group using the new payment rule introduced for EPPM.

Mechanism 4.2.1 (**k-Means Peer Prediction Method).** *The k-Means Peer Prediction Method is defined as:*

1. *Mechanism receives reports $r_{n,i} \in \mathcal{S}$ from every agent $i \in \mathcal{I}$ for every world states indexed $n \in (1, ..., N)$.*

2. *Estimates the confusion matrices $\hat{\mathbf{\Theta}}$ and the state priors $\hat{\boldsymbol{\rho}}$ using the EM algorithm of the* PrivateConfusion *model assuming truthful reporting.*

3. *Computes the group membership Γ, the central confusion matrices $\hat{\mathbf{\Theta}}_\mu^g$, and the outliers $\mathcal{O}$ from the k-Means-Confusion given the estimated confusion matrices $\hat{\mathbf{\Theta}}$ and maximum distance parameter δ_{max}.*

4. *For each group, computes the estimated state posterior $\hat{\boldsymbol{\vartheta}}^g$ and the estimated state priors $\hat{\boldsymbol{\phi}}^g$ from the estimated group confusion matrices $\hat{\mathbf{\Theta}}_\mu^g$ and the estimated state priors $\hat{\boldsymbol{\rho}}$.*

5. *For any agent $o \in \mathcal{O}$, randomly selects $g \in \mathcal{G}$ and classifies $\gamma_o = g$.*

6. *Pays agent i, a member of group $g \in \mathcal{G}$, for item n a payment based on $x_i(r_{n,i}, r_{n,j}) = R(\hat{\boldsymbol{\vartheta}}^g_{r_{n,i}}, r_{n,j}) - R(\hat{\boldsymbol{\phi}}^g, r_{n,j})$, where $r_{n,j}$ is the report from a reference agent j selected from group g.*

If there is 1-to-1 relationship between the set of states and the set of signals such that the mechanism designer can model the conditional signal probabilities as

error-rates of the signal observation, then due to the implementation of strictly proper scoring rules, kPPM naturally rewards the agents who exhibit lower (higher) error-rates with higher (lower) expected payments. By rewarding the agents with lower error-rates, kPPM is able to incentivize not only truthful reporting but also induce effort by the agents to accurately observe the signals. We demonstrate this property with an example.

Example 5. *We return to the example of the computer scientist and his image labeling human computation system. After running EPPM with a small number of workers, he decides to expand the number of workers. However, based on his past experience, he believes there are two classes of workers - Professionals and Amateurs.*

Professionals tends to be workers who make their living working for human computation systems. They value their reputation and make earnest effort to provide high quality work for human computation systems.

On the other hand, Amateurs are in general hobbyists. They partake in human computation systems to make a quick buck on the side, but they are not fully committed to working for human computation systems for their living. As such, their work tends to be of lower quality.

The computer scientist wishes to identify the Professionals and reward them with higher payments. He adopts kPPM for his human computation system. After receiving all the reports, the mechanism computes the central confusion matrices of the Professionals and the Amateurs:

$$\hat{\Theta}_{\mu}^{Pr} = \begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix} \quad \hat{\Theta}_{\mu}^{Am} = \begin{bmatrix} 0.75 & 0.25 \\ 0.26 & 0.74 \end{bmatrix}$$

Using the estimates of the state priors and the group confusion matrices, the mechanism estimates the following signal priors and signal posteriors for each group:

$$\hat{\boldsymbol{\phi}}^{Pr} = \begin{bmatrix} 0.28 & 0.72 \end{bmatrix} \quad \hat{\boldsymbol{\phi}}^{Am} = \begin{bmatrix} 0.39 & 0.61 \end{bmatrix}$$

$$\hat{\boldsymbol{\vartheta}}^{Pr} = \begin{bmatrix} 0.96 & 0.04 \\ 0.02 & 0.98 \end{bmatrix} \quad \hat{\boldsymbol{\vartheta}}^{Am} = \begin{bmatrix} 0.52 & 0.48 \\ 0.31 & 0.69 \end{bmatrix}$$

Based on these estimates, kPPM computes the payment schedules for the Professionals and the Amateurs, which are shown in Table 4.2 and Table 4.3, respectively.

r_i \ r_j	0	1
0	1.03	−1.69
1	−0.88	0.16

Table 4.2: Example 5 kPPM Professional Payment Matrix

r_i \ r_j	0	1
0	0.28	−0.24
1	−0.21	0.11

Table 4.3: Example 5 kPPM Amateur Payment Matrix

The mechanism identifies worker j as a Professional and worker i as an Amateur. Assuming all her group peers reported truthfully, if worker j observed signal $S_{n,j} = 0$ and reported truthfully, her expected payment would be

$$\begin{aligned} & P(S_{n,h} = 0 | S_{n,j} = 0) \cdot x_j(0,0) + P(S_{n,h} = 1 | S_{n,j} = 0) \cdot x_j(0,1) \\ \approx\ & 0.96 \cdot 1.03 + 0.04 \cdot -1.69 \\ \approx\ & 0.9212 \end{aligned}$$

For signal $S_{n,j} = 1$, her expected payment from truthful reporting would be

$$\begin{aligned} & P(S_{n,h} = 0 | S_{n,j} = 1) \cdot x_j(1,0) + P(S_{n,h} = 1 | S_{n,j} = 1) \cdot x_j(1,1) \\ \approx\ & 0.02 \cdot -0.88 + 0.98 \cdot 0.16 \\ \approx\ & 0.1392 \end{aligned}$$

Compare these expected payments for the Professional worker against those of

an Amateur worker. Assuming all her group peers reported truthfully, if worker i observed signal $S_{n,i} = 0$, the expected payment from truthful reporting is

$$\begin{aligned} & P(S_{n,f} = 0|S_{n,i} = 0) \cdot x_i(0,0) + P(S_{n,f} = 1|S_{n,i} = 0) \cdot x_i(0,1) \\ \approx\ & 0.52 \cdot 0.28 + 0.48 \cdot -0.24 \\ \approx\ & 0.0304 \end{aligned}$$

For signal $S_{n,i} = 1$, the expected payment is

$$\begin{aligned} & P(S_{n,f} = 0|S_{n,i} = 1) \cdot x_i(1,0) + P(S_{n,f} = 1|S_{n,i} = 1) \cdot x_i(1,1) \\ \approx\ & 0.31 \cdot -0.21 + 0.69 \cdot 0.11 \\ \approx\ & 0.0108 \end{aligned}$$

For either signals, if the Professional reports truthfully, her expected payment is higher than that of the Amateur. This difference in the expected payment across the different types of worker can provide an incentive for the Amateurs to make more effort to accurately observe their signals in order to be classified as Professionals in the mechanism. The overall effect is improvement in the accuracy of the reports for the human computation system.

Let us assume that the peers of agent i reported truthfully. If agent i did not report truthfully, one of the three outcomes are possible in kPPM:

1. Agent i is correctly classified into $g \in \mathcal{G}$ where $\gamma_i = g$
2. Agent i is incorrectly classified into $g^{'} \in \mathcal{G}$ where $\gamma_i \neq g^{'}$
3. Agent i is classified as an outlier because $d(\Theta^i|\hat{\mathbf{\Theta}}_\mu) > \delta_{max}$, and she is classified into a randomly selected $g^{''} \in \mathcal{G}$

If outcome 1 is realized, agent i is worse off than if she had reported truthfully because within her group the expected payment is maximized only via truthful reporting strategy. If outcome 2 is realized, agent i's belief about the signal posteriors of

her peer agents are inaccurate; therefore, her reports cannot maximize her expected payment. Finally, if outcome 3 is realized, then agent i faces either outcome 1 or outcome 2 depending on her randomly assigned group membership. We demonstrate this property of kPPM with an example.

Example 6. *Worker x joins the image labeling human computation system. Worker x is an Amateur; however, unlike his group peers, he decides to adopt randomly reporting strategy. Instead of truthfully reporting his observations, worker x decides to randomly label 'cat' and 'dog' with 50/50 probability.*

From his reports, kPPM incorrectly estimates his confusion matrix:

$$\hat{\Theta}^x = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

and the mechanism classifies him as an outlier, neither a Professional nor an Amateur.

Let us assume that kPPM randomly assigns worker x as a Professional. In this case, the true state posteriors with respect to worker x is shown in Table 4.4.

$S_{n,x}^{Am}$ \ $S_{n,j}^{Pr}$	0	1
0	0.552	0.448
1	0.134	0.866

Table 4.4: Example 6 kPPM Signal Posteriors

If his true observed signal is $S_{n,x} = 0$, then, the expected payment for worker x is

$$\begin{aligned} & 0.5 \cdot \{P(S_{n,j}^{Pr} = 0 | S_{n,x}^{Am} = 0) \cdot x_x(0,0) + P(S_{n,j}^{Pr} = 1 | S_{n,x}^{Am} = 0) \cdot x_x(0,1)\} \\ & + 0.5 \cdot \{P(S_{n,j}^{Pr} = 0 | S_{n,x}^{Am} = 0) \cdot x_x(1,0) + P(S_{n,j}^{Pr} = 1 | S_{n,x}^{Am} = 0) \cdot x_x(1,1)\} \\ & \approx 0.5 \cdot \{0.552 \cdot 1.03 + 0.448 \cdot -1.69\} + 0.5 \cdot \{0.552 \cdot -0.88 + 0.448 \cdot 0.16\} \\ & \approx -0.301 \end{aligned}$$

For signal $S_{n,x} = 1$,

$$\begin{aligned}
&0.5 \cdot \{P(S^{Pr}_{n,j} = 0 | S^{Am}_{n,x} = 1) \cdot x_x(0,0) + P(S^{Pr}_{n,j} = 1 | S^{Am}_{n,x} = 1) \cdot x_x(0,1)\} \\
&+ 0.5 \cdot \{P(S^{Pr}_{n,j} = 0 | S^{Am}_{n,x} = 1) \cdot x_x(1,0) + P(S^{Pr}_{n,j} = 1 | S^{Am}_{n,x} = 1) \cdot x_x(1,1)\} \\
&\approx \ 0.5 \cdot \{0.134 \cdot 1.03 + 0.866 \cdot -1.69\} + 0.5\{0.134 \cdot -0.88 + 0.866 \cdot 0.16\} \\
&\approx \ -0.652
\end{aligned}$$

The total expected payment is

$$\begin{aligned}
&P(S^{Am}_{n,x} = 0) \cdot -0.301 + P(S^{Am}_{n,x} = 1) \cdot -0.652 \\
&\approx \ 0.39 \cdot -0.301 + 0.61 \cdot -0.652 \\
&\approx \ -0.515
\end{aligned}$$

On the other hand, let us assume that kPPM randomly, but correctly, assigns worker x as an Amateur. If his observed signal is $S_{n,x} = 0$, then the expected payment for worker x is

$$\begin{aligned}
&0.5 \cdot \{P(S^{Am}_{n,i} = 0 | S^{Am}_{n,x} = 0) \cdot x_x(0,0) + P(S^{Am}_{n,i} = 1 | S^{Am}_{n,x} = 0) \cdot x_x(0,1)\} \\
&+ 0.5 \cdot \{P(S^{Am}_{n,i} = 0 | S^{Am}_{n,x} = 0) \cdot x_x(1,0) + P(S^{Am}_{n,i} = 1 | S^{Am}_{n,x} = 0) \cdot x_x(1,1)\} \\
&\approx \ 0.5 \cdot \{0.52 \cdot 0.28 + 0.48 \cdot -0.24\} + 0.5 \cdot \{0.52 \cdot -0.21 + 0.48 \cdot 0.11\} \\
&\approx \ -0.013
\end{aligned}$$

For signal $S_{n,x} = 1$,

$$\begin{aligned}
&0.5 \cdot \{P(S^{Am}_{n,i} = 0 | S^{Am}_{n,x} = 1) \cdot x_x(0,0) + P(S^{Am}_{n,i} = 1 | S^{Am}_{n,x} = 1) \cdot x_x(0,1)\} \\
&+ 0.5 \cdot \{P(S^{Am}_{n,i} = 0 | S^{Am}_{n,x} = 1) \cdot x_x(1,0) + P(S^{Am}_{n,i} = 1 | S^{Am}_{n,x} = 1) \cdot x_x(1,1)\} \\
&\approx \ 0.5 \cdot \{0.31 \cdot 0.28 + 0.69 \cdot -0.24\} + 0.5 \cdot \{0.31 \cdot -0.21 + 0.69 \cdot 0.11\} \\
&\approx \ -0.034
\end{aligned}$$

The total expected payment is,

$$\begin{aligned} & P(S_{n,x}^{Am} = 0) \cdot -0.301 + P(S_{n,x}^{Am} = 1) \cdot -0.652 \\ \approx\ & 0.39 \cdot -0.013 + 0.61 \cdot -0.034 \\ \approx\ & -0.026 \end{aligned}$$

Compare these results to the expected payment of truthful reporting strategy for worker x for all signals,

$$\begin{aligned} & P(S_{n,x}^{Am} = 0) \cdot 0.0304 + P(S_{n,x}^{Am} = 1) \cdot 0.0108 \\ \approx\ & 0.39 \cdot 0.0304 + 0.61 \cdot 0.0108 \\ \approx\ & 0.0184 \end{aligned}$$

Worker x is far worse off being classified as a Professional. Even if he is classified as an Amateur, his expected payment is less than if he had reported truthfully. Whether worker x is assigned as a Professional or as an Amateur, his expected payment is greater if he had reported truthfully.

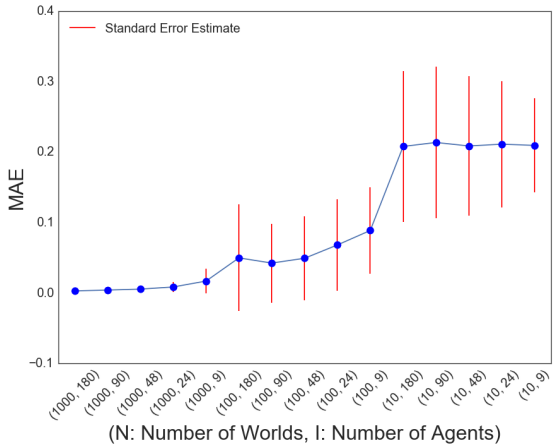
Given a sufficently large numer of data points, kPPM is also robust against collusion. The same as EPPM, this property is due to the empirical estimation of the probabilities and the new scoring rule. We demonstrate this property of kPPM with a simulation in the following section.

4.2.3 Analysis

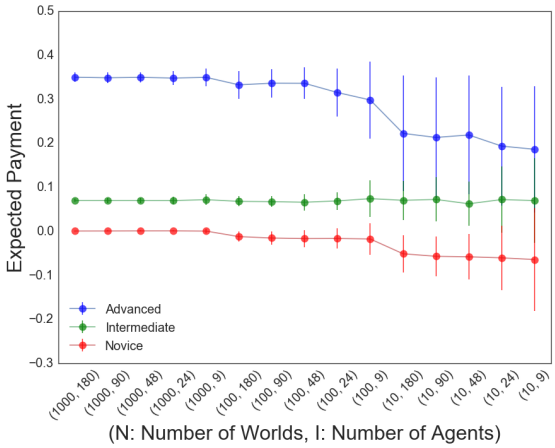
Performance of EM Algorithm and Expected Payment of kPPM

Similar to the experiment of EPPM, we examine the expected payments to truthfully reporting agents in kPPM using simulated data. As before, we fix the states to $\mathcal{T} = \{0, 1, 2\}$ and the signals to $\mathcal{S} = \{0, 1, 2\}$.

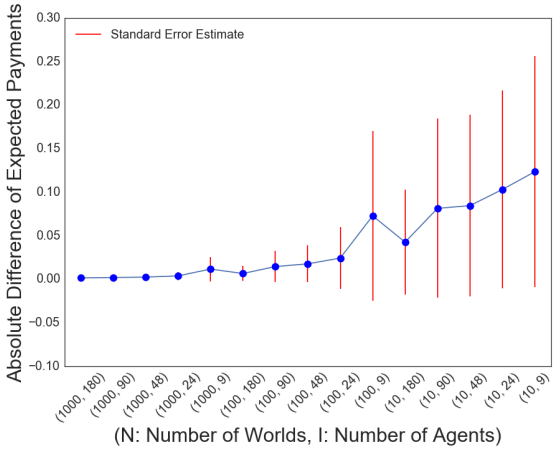
We create three different types of agents - (1) Advanced, (2) Intermediate, and (3) Novice. Every agent belong to one of the three groups, and the group as-



(a) MAEs of $\hat{\Theta}$



(b) Expected Payment of EPPM



(c) Absolute Difference CPPM vs kPPM

Figure 4.5: k-Means Peer Prediction Method: Accuracy

signment are distributed according to a Multinomial distribution with parameter $\alpha = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$. This results in a group assignment with roughly even membership in all three groups.

The hyperparameters for the confusion matrices for each groups are as follows,

$$\Lambda^{Adv} = \begin{bmatrix} 98 & 1 & 1 \\ 1 & 98 & 1 \\ 1 & 1 & 98 \end{bmatrix} \quad \Lambda^{Int} = \begin{bmatrix} 76 & 23 & 1 \\ 12 & 76 & 12 \\ 12 & 12 & 76 \end{bmatrix} \quad \Lambda^{Nov} = \begin{bmatrix} 50 & 25 & 25 \\ 25 & 50 & 25 \\ 25 & 25 & 50 \end{bmatrix}$$

We vary the number of items, $N = (10, 100, 1000)$ and the number of agents, $I = (9, 24, 48, 90, 180)$. Similar to the experiment in Section 4.1.3, for each permutation of $(N \times I)$, we experiment on three different state priors. For each permutation and state priors, we run 50 experiments and report the average of the results.

In addition, because the GroupBelief model assumes that there is a sufficiently large number of agents such that each group has at least two agents, we sample the groups so that every group contains at least two agents.

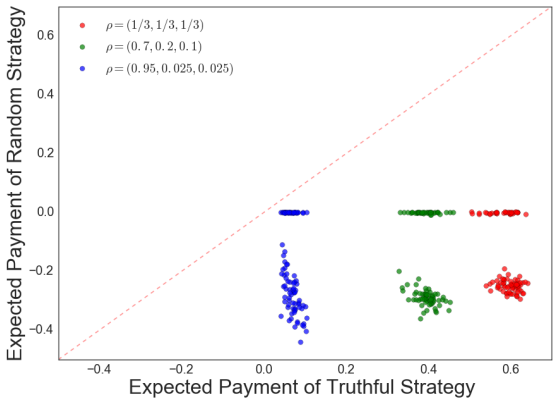
Figure 4.5a shows the MAE of recovered $\hat{\mathbf{\Theta}}$ from the EM algorithm. We note that the trend is similar to what we observed in Figure 3.4a in Chapter 3 Section 3.3.3. However, our primary interest lies in the expected payments of kPPM in various environment.

Figure 4.5b shows the expected payments of the three groups of agents. The Advanced agents consistently receive higher expected payments than the Intermediate agents and the Novice agents, and the Intermediate agents likewise receive higher expected payments than the Novice agents. As the the number of data points decline, the standard errors of the expected payments also increase for all groups; nevertheless, kPPM manages to maintain the gaps in the expected payments between the three groups.

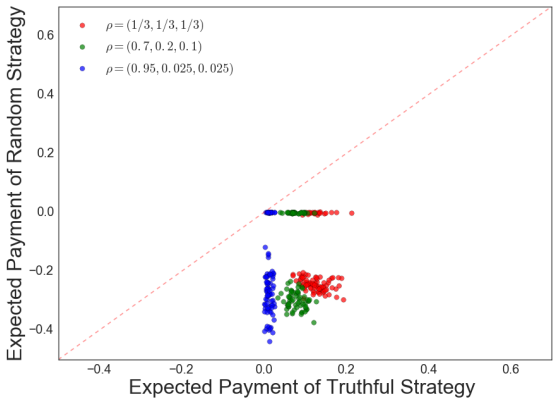
Finally, we compute the absolute differences between the expected payments of all three groups in kPPM against the expected payments of CPPM in which the designer has knowledge of all the parameters of the model including the group membership. In Figure 4.5c, we observe sharp increase in the standandard errors of the absolute difference as the number of data points go to $(N \times I) < 2400$.

Robustness of kPPM against Strategic Reporting

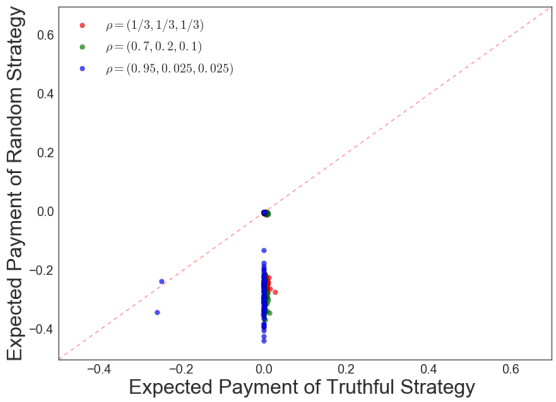
Similar to the experiment in Section 4.1.3, we compare the expected payments of the truthful reporting strategy against various reporting strategies in kPPM. We fix the number of items to $N = 1000$ and the number of agents to $I = 24$. We sample



(a) Advanced Agent



(b) Intermediate Agent



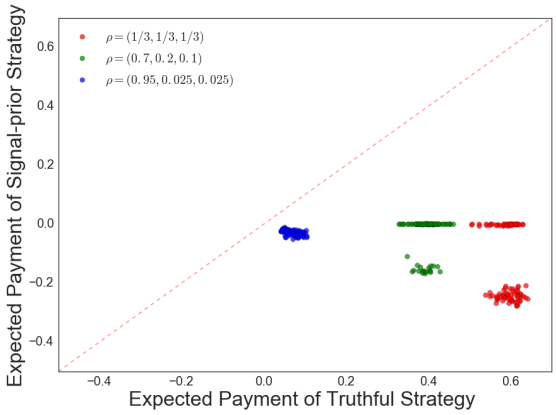
(c) Novice Agent

Figure 4.6: k-Means Peer Prediction Method: Robustness – Random Strategy

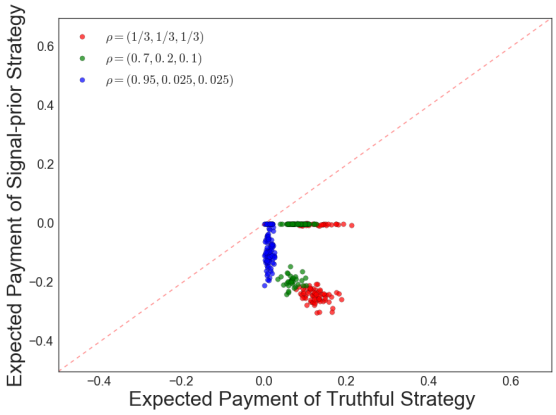
the group membership such that each group has exactly eight agents.

First, in Figure 4.6, we examine the expected payments of the three different agent types if they adopted the random reporting strategy. For all three types of agents, we see that there is a bimodal distribution of the expected payments under the random reporting strategy, which was not observed in the EPPM experiment. One mode is centered close to 0, and the other is centered around -2.5 with standard deviation of approximately 0.1.

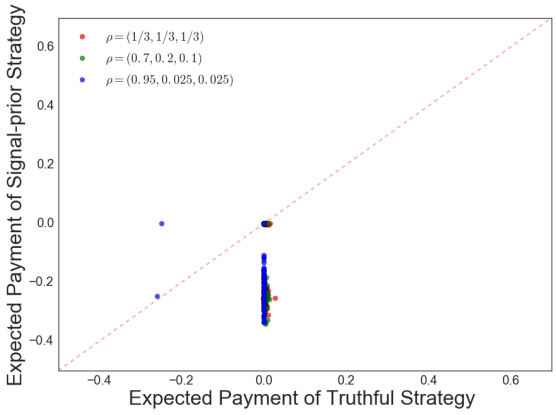
In kPPM, if an agent does not report truthfully she can expect one of three possible outcomes: (1) be assigned to the correct group, (2) be assigned to a wrong



(a) Advanced Agent



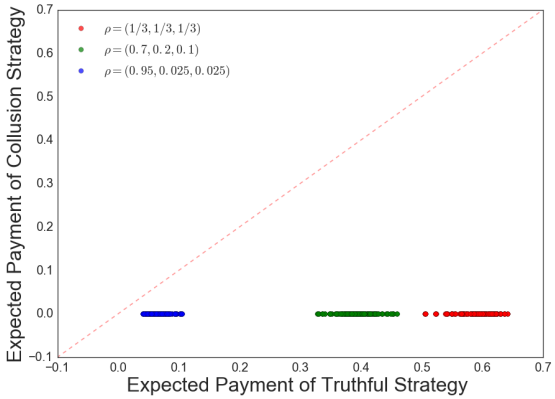
(b) Intermediate Agent



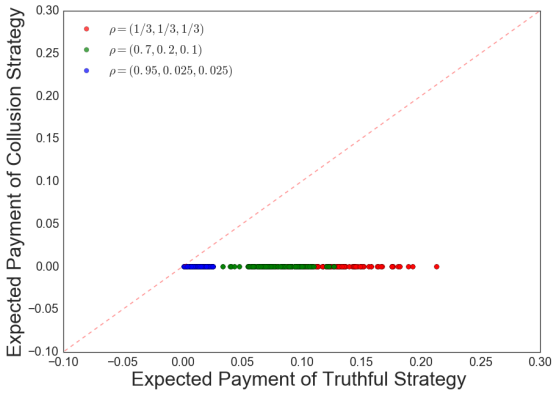
(c) Novice Agent

Figure 4.7: k-Means Peer Prediction Method: Robustness – Signal-prior Strategy

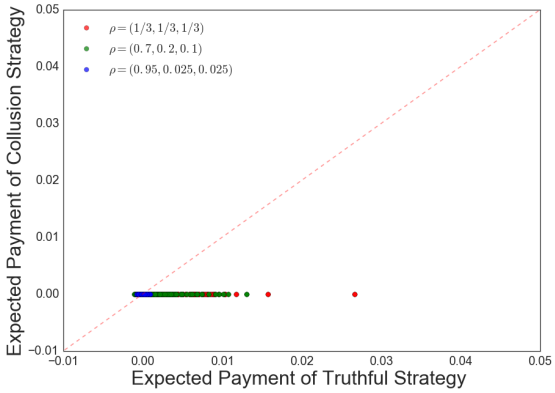
group, or (3) if identified as an outlier, be randomly selected into any group. For the randomly reporting agent, kPPM should identify this agents an outlier and will randomly assign her to one of the three groups. The consequence of this random assignment is the bimodal distribution of expected payments. If she were correctly assigned to her true group by chance, her expected payments would be closer to zero. On the other hand, if she were assigned to another group by chance, her expected payments would be even lower in the negatives. All in all, Figure 4.6 shows that the agents' expected payments are higher in general under the truthful reporting strategy for all three agent types.



(a) Advanced Agent



(b) Intermediate Agent



(c) Novice Agent

Figure 4.8: k-Means Peer Prediction Method: Robustness – Collusion Strategy

Likewise, we experiment with the Signal-prior reporting strategy that was introduced in Section 4.1.3. Figure 4.7 also exhibits a bimodal distribution of the expected payments under the Signal-prior strategy. kPPM classifies the strategically reporting agents as outliers, then randomly assigns them to one of the three groups. The end result is that for all three agent types the agents are worse off than if they had stuck with the truthful reporting strategy.

Finally, in Figure 4.8, we examine collusion among all 24 agents. For all three agent types, the consequence of collusion is that their expected payment becomes zero, and they are better off reporting truthfully.

4.3. Empirical Scoring Rule Mechanism

While the GroupBelief model relaxes the assumption on the homogeneity of agents, it is still restrictive in a setting where every agent holds her own unique, private belief. For example, a human computation system may employ a small number of workers, and the system designer cannot assume that every worker belongs to a group in which the members share similar beliefs. In such a setting, the designer should assume that every worker is her own type of worker with unique, private beliefs of her own.

However, Radanovic and Faltings (2015) show that when there is a complete heterogeneity of agents, no existing peer prediction mechanism in the information elicitation literature can strictly incentivize the agents to report truthfully. This problem also extends to the Empirical Peer Prediction Method or the k-Means Peer Prediction Method.

In this section, I introduce the *Empirical Scoring Rule Mechanism* (ESRM). I demonstrate empirically that ESRM is a truthful peer prediction mechanism for the PrivateBelief model, which assumes complete heterogeneity of agents. In addition, I show that ESRM can also serve as a truthful peer prediction mechanism in other models.

4.3.1 Model

The PrivateBelief model is an extension of the base model of peer prediction and the PrivateConfusion model. Every agent $i \in \mathcal{I}$ has own unique confusion matrix Θ^i. As were the cases in the CommonBelief model and the GroupBelief model, the mechanism designer does not know the state priors $\boldsymbol{\rho}$ or the conditional signal probabilities of the individual agents $\boldsymbol{\Theta} = \{\Theta^1, ..., \Theta^I\}$.

The graphical model representation in Figure 4.9 summarizes the PrivateBelief model.

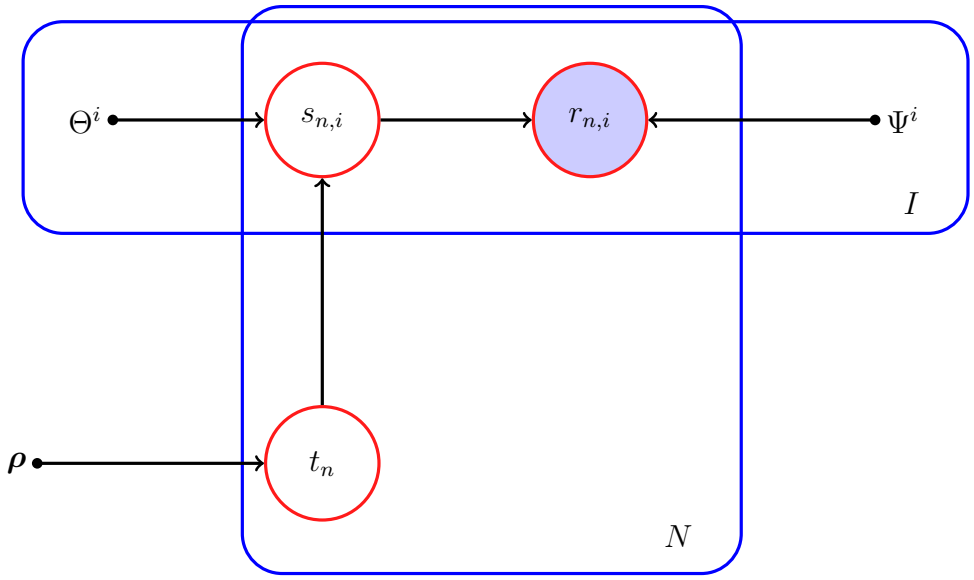


Figure 4.9: PrivateBelief Model

4.3.2 Mechanism

In Chapter 2 Section 2.2, we noted the symmetry of signal posteriors in the base model of peer prediction mechanism,

$$P(S_j = s^{'}|S_i = s) = P(S_i = s^{'}|S_j = s)\ \forall i, j \in \mathcal{I} \tag{4.12}$$

The common belief model satisfies this property because the conditional signal probabilities are equivalent,

$$P(S_j = k|T = m) = P(S_i = k|T = m) \tag{4.13}$$

However, the symmetrical property is not satisfied in the PrivateBelief model due to differences in the confusion matrices Θ^i and Θ^j.

Radanovic and Faltings (2015) show that when the agents hold private beliefs, existing peer prediction mechanisms cannot reliably elicit truthful reports from the agents. Let us use an example from Radanovic and Faltings (2015) to demonstrate this problem.

Consider the case in which $\mathcal{I} = \{i, j\}$ and $\mathcal{S} = \{0, 1\}$. Suppose that a private belief model yields the following relationship for agent i and agent j:

$$P(S_j = 0|S_i = 0) = P(S_i = 1|S_j = 0) \tag{4.14}$$

$$P(S_j = 1|S_i = 1) = P(S_i = 0|S_j = 1) \tag{4.15}$$

In this case, if both agent i and agent j observed the same signal such that $S_i = S_j$, then the expected payments for agent i and agent j would be the same for reporting the opposite signals, preventing the mechanism from strictly incentivizing the agents to report truthfully to both signals. In this case, CPPM, EPPM, and kPPM will fail to be truthful peer prediction mechanisms.

The Empirical Scoring Rule Mechanism provides a solution to this problem in the PrivateBelief model. Instead of using the reports of the peers and the signal posteriors, ESRM uses the estimated state posteriors $P(T_n|S_{n,i} = k)$ and the estimated true state $\hat{t}_n$ and pays the agent as if the agent is engaged in the information elicitation with verifiable truth using strictly proper scoring rule as seen in Chapter 2 Section 2.1.

The state posteriors represent the agent i's belief about the probability of the true state t_n of item n given her observed signal of $S_{n,i}$. Let us denote the state posteriors of agent i as a $K \times M$ matrix:

$$\Pi^i = \begin{bmatrix} \boldsymbol{\pi}_1^i \\ \vdots \\ \boldsymbol{\pi}_K^i \end{bmatrix} \tag{4.16}$$

where $\boldsymbol{\pi}_k^i = \begin{bmatrix} \pi_{k,1}^i & \dots & \pi_{k,M}^i \end{bmatrix}$, a probability vector, denotes the k-th row of Π^i and $\pi_{k,m}^i = P(T_n = m|S_{n,i} = k)$ represents agent i's probabilistic belief about the true state of item n is m conditioned on her signal observation of k.

Similar to the new payment introduced for EPPM and kPPM, the payment

rule for ESRM is as follows:

$$x_i(r_{n,i}, \hat{t}_n) = R(\hat{\boldsymbol{\pi}}^i_{r_{n,i}}, \hat{t}_n) - R(\hat{\boldsymbol{\rho}}, \hat{t}_n) \tag{4.17}$$

where $\hat{\boldsymbol{\pi}}^i_{r_{n,i}}$ is the estimated state posteriors of agent i for signal $r_{n,i}$. In words, ESRM's payment rule is the difference between scores of agent i's estimated state posteriors and the mechanism's estimated state priors.

Mechanism 4.3.1 (**Empirical Scoring Rule Mechanism).** *The Empirical Scoring Rule Mechanism is defined as:*

1. *Mechanism receives reports $r_{n,i} \in \mathcal{S}$ from every agent $i \in \mathcal{I}$ for every world states indexed $n \in (1, ..., N)$.*

2. *Estimates the confusion matrices $\hat{\boldsymbol{\Theta}}$, the state priors $\hat{\boldsymbol{\rho}}$, and the true states $\hat{\boldsymbol{t}}$ from the reports $\boldsymbol{r}$ using the EM algorithm of the* PrivateConfusion *model assuming truthful reporting.*

3. *Computes state posteriors $\hat{\Pi}^i$ from the estimated confusion matrix $\hat{\Theta}^i$ and the state priors $\hat{\boldsymbol{\rho}}$ for every agent $i \in \mathcal{I}$.*

4. *Pays agent i for item n a payment based on*

$$x_i(r_{n,i}, \hat{t}_n) = R(\hat{\boldsymbol{\pi}}^i_{r_{n,i}}, \hat{t}_n) - R(\hat{\boldsymbol{\rho}}, \hat{t}_n)$$

where R is a strictly proper scoring rule.

We should note that the agents do not have advance knowledge or belief about the details of the model. As long as the agents believe that the mechanism accurately estimates the parameters of the model and accurately infers the true labels, they can be confident that the mechanism will use the recovered true labels to award them appropriately for truthful reports.

Assuming that with sufficiently large N and I ESRM accurately estimates the parameters and recovers the true states such that $\hat{\boldsymbol{\Theta}} \approx \boldsymbol{\Theta}$, $\hat{\boldsymbol{\rho}} \approx \boldsymbol{\rho}$, and $\hat{\boldsymbol{t}} \approx \boldsymbol{t}$, then ESRM is a truthful empirical peer prediction mechanism due to the strictly proper property of its payment rule. We demonstrate the truthful property of ESRM with an example.

Example 7. *Let us return to the computer scientist and his image labeling human computation system. He decides to hire six workers for his project, and he believes that each worker has unique level of competency. In other words, each worker has her own unique confusion matrix. For this problem, the computer scientist decides to implement ESRM.*

The state priors of the model are

$$\boldsymbol{\rho} = \begin{bmatrix} 0.3 & 0.7 \end{bmatrix}$$

The confusion matrix and state posteriors of worker i are as follows,

$$\Theta^i = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}$$

$$\Pi^i = \begin{bmatrix} 0.6 & 0.4 \\ 0.1385 & 0.8615 \end{bmatrix}$$

Suppose that worker i and all her peers reported truthfully. ESRM runs the EM algorithm to estimate the state priors and the individual confusion matrix of every worker. Suppose that the estimated state prior is

$$\hat{\boldsymbol{\rho}} = \begin{bmatrix} 0.29 & 0.71 \end{bmatrix}$$

Furthermore, worker i's estimated confusion matrix is

$$\hat{\Theta}^i = \begin{bmatrix} 0.69 & 0.31 \\ 0.22 & 0.78 \end{bmatrix}$$

From $\hat{\boldsymbol{\rho}}$ and $\hat{\Theta}^i$, the mechanism computes the estimated state posteriors of the worker i,

$$\hat{\Pi}^i = \begin{bmatrix} 0.56 & 0.44 \\ 0.14 & 0.86 \end{bmatrix}$$

From these estimates, the mechanism designs the payment schedule for worker i shown in Table 4.5.

$r_{n,i}$ \ $\hat{t}_n$	0	1
0	0.614	−0.45
1	−0.488	0.132

Table 4.5: Example 7 ESRM Truthful Reporting Payment Matrix

The total expected payment of worker i from reporting truthfully is

$$\begin{aligned} &P(T_n = 0, S_{n,i} = 0) \cdot x_i(0,0) + P(T_n = 1, S_{n,i} = 0) \cdot x_i(0,1) \\ &+ P(T_n = 0, S_{n,i} = 1) \cdot x_i(1,0) + P(T_n = 1, S_{n,i} = 1) \cdot x_i(1,1) \\ &= 0.21 \cdot 0.614 + 0.14 \cdot -0.45 + 0.09 \cdot -0.488 + 0.56 \cdot 0.132 \\ &= 0.096 \end{aligned} \tag{4.18}$$

However, let us suppose that worker i reported randomly. The EM algorithm would had computed her confusion matrix and state posteriors as follows,

$$\hat{\Theta}^i = \begin{bmatrix} 0.53 & 0.47 \\ 0.48 & 0.52 \end{bmatrix}$$

$$\hat{\Pi}^i = \begin{bmatrix} 0.33 & 0.67 \\ 0.28 & 0.72 \end{bmatrix}$$

Based on these estimates, ESRM computes the payment schedule for worker i shown in Table 4.6.

$r_{n,i}$ \ $\hat{t}_n$	0	1
0	0.062	−0.029
1	−0.062	0.026

Table 4.6: Example 7 ESRM Random Reporting Payment Matrix

The total expected payment of worker i from the randomly reporting strategy is

$$\begin{aligned} &P(T_n = 0, S_{n,i} = 0) \cdot x_i(0,0) + P(T_n = 1, S_{n,i} = 0) \cdot x_i(0,1) \\ &+ P(T_n = 0, S_{n,i} = 1) \cdot x_i(1,0) + P(T_n = 1, S_{n,i} = 1) \cdot x_i(1,1) \\ &= 0.21 \cdot 0.062 + 0.14 \cdot -0.029 + 0.09 \cdot -0.062 + 0.56 \cdot 0.026 \\ &= 0.018 \end{aligned} \tag{4.19}$$

For worker i, she would have had higher expected payment under the truthful reporting strategy.

We note that ESRM is also robust against collusion. If all agents coordinate to report one signal repeatedly, the $\hat{\boldsymbol{\rho}}$ and $\hat{\boldsymbol{\Theta}}$ (consequently, $\hat{\boldsymbol{\Pi}}$) will be uninformative of the true probabilities. However, the result is deterimental for the colluding agents as well. We demonstrate this property with an example.

Example 8. *Let us return to the computer scientist in the earlier example. Now, he believes that each worker has unique capabilities, and he decides to deploy ESRM.*

The computer scientist's system faces collusion attack from multiple agents who have coordinated to label 'cat' for all images. Consquently, when the mechanism performs the EM algorithm to compute the model parameters, it returns the the

following result:

$$\hat{\boldsymbol{\rho}} = \begin{bmatrix} 0.0 & 1.0 \end{bmatrix}$$

and for every worker i, her confusion matrix is

$$\hat{\Theta}^i = \begin{bmatrix} 0.0 & 1.0 \\ 0.0 & 1.0 \end{bmatrix}$$

As a result, the estimate of worker i's state posteriors are

$$\hat{\Pi}^i = \begin{bmatrix} NaN & NaN \\ 0.0 & 1.0 \end{bmatrix}$$

Moreover, the recovered labels are

$$\hat{\boldsymbol{t}} = \{1, ..., 1\}$$

The collusive workers provided no useful information for the computer scientist; however, their coordinated attack was also deterimental to their own payments because the payment for every worker was as follows

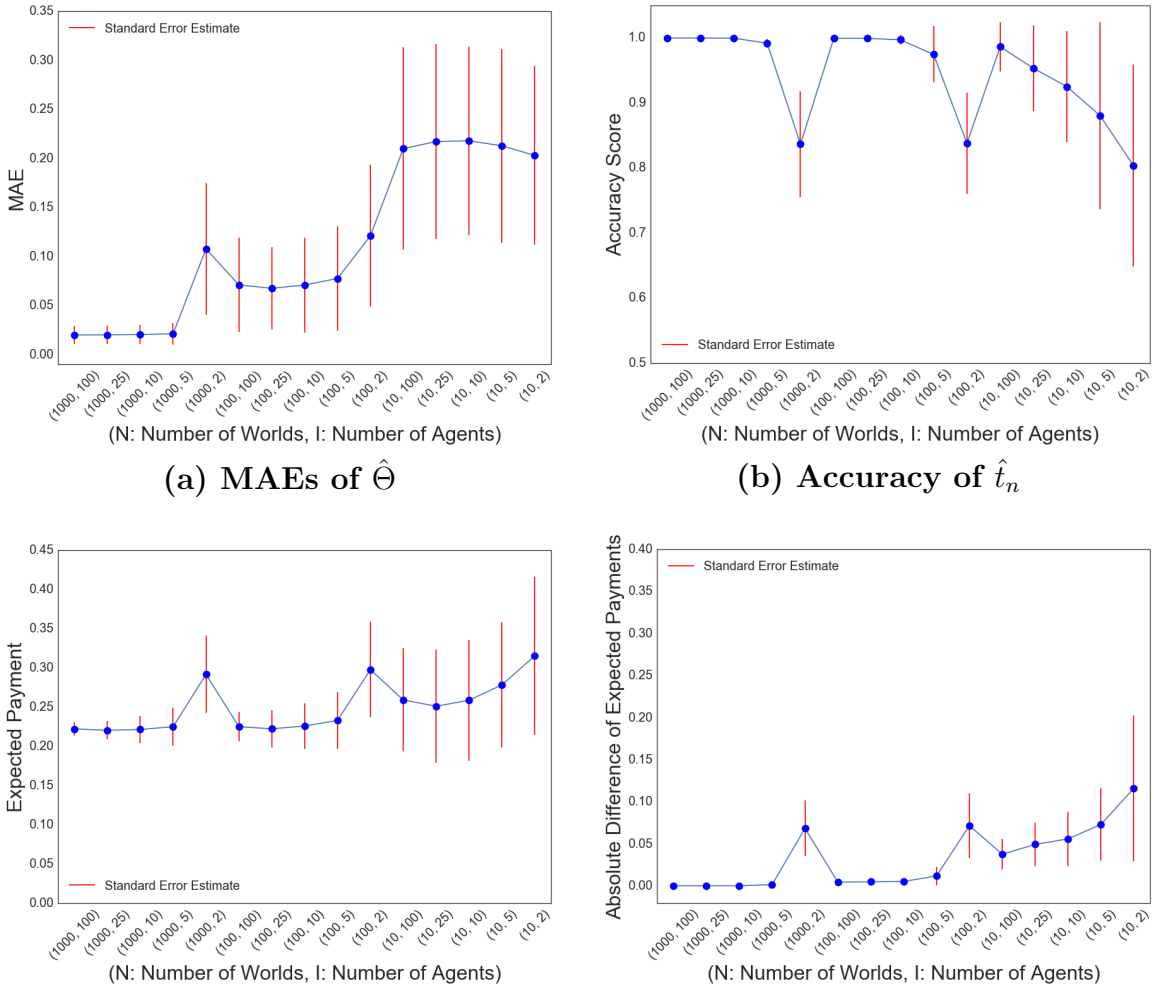
$$\begin{aligned} x(0,0) &= R_Q((0.0, 1.0), 1) - R_Q((0.0, 1.0), 1) \\ &= 1.0 - 1.0 = 0.0 \end{aligned}$$

Although the colluding workers watsted his time, they did not inflict any financial cost.

4.3.3 Analysis

Performance of EM Algorithm and Expected Payment of ESRM

Using sythetic data, we examine the expected payment to a truthfully reporting agent in ESRM under different environment. As before, we fix the states to



(a) MAEs of $\hat{\Theta}$ **(b) Accuracy of $\hat{t}_n$**

(c) Expected Payment of ESRM **(d) Absolute Difference**

Figure 4.10: Empirical Scoring Rule Mechanism: Accuracy

$\mathcal{T} = \{0, 1, 2\}$ and the signals to $\mathcal{S} = \{0, 1, 2\}$.

We vary the number of items $N = (10, 100, 1000)$ and the number of agents $I = (2, 5, 10, 25, 100)$. Similar to the experiments in Section 4.1.3 and Section 4.2.3, for each permutation of $(N \times I)$, we experiment on the three different state priors. For each permutation of $(N \times I)$ and state priors, we run 50 experiments and report the average of the results.

For each experiment, we sample I unique confusion matrices, as described in

Section 3.4.1, using the following hyperparameter:

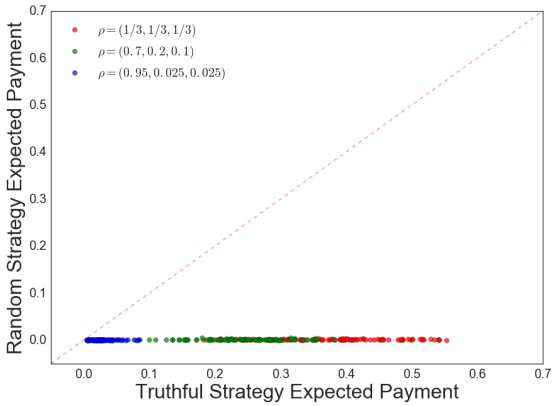
$$\Lambda = \begin{bmatrix} 10 & 1 & 1 \\ 1 & 10 & 1 \\ 1 & 1 & 10 \end{bmatrix}$$

Unlike EPPM and kPPM, the accuracy of ESRM payment also depends on the accuracy of the recovered true labels $\hat{\boldsymbol{t}}$. Therefore, we examine the accuracy of the EM algorithm in recovering the true labels as well. Figure 4.10a and Figure 4.10b show the accuracies of recovered confusion matrices and the true labels, respectively. We note that the two figures are quite similar to Figure 3.4 from Chapter 3 Section 3.3.3. We notice steep decline in the accuracies of the EM algorithm where the number of agents is 2 even if the number of items is as large as 1000.

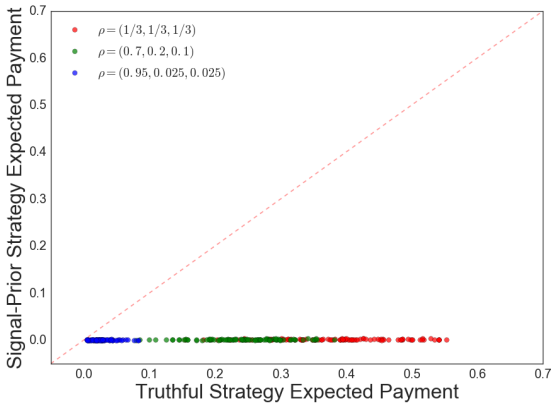
The deterioration in the accuracy of the EM algorithm corresponds to the spike in the expected payment of ESRM as shown in Figure 4.10c. Lastly, we compare the expected payments of ESRM against the theoretical payments of ESRM if the designer had *a priori* knowledge of the $\boldsymbol{\rho}$, $\boldsymbol{\Theta}$, and $\boldsymbol{t}$. We note the steep increase in the absolute difference between the theoretical score and the empirical score where the number of agents is 2. The expected payment in ESRM appears to be deviate far from the theoretical expected payment when there's deterioration in the accuracy of the recovered states.

As we shall see in the subsequent section where we compare EPPM and kPPM against ESRM, with a larger number of data points ESRM yields more accurate results in terms of absolute difference between the expected payments of empirical mechanism versus the theoretical mechanism. Whereas with a smaller number of data points, ESRM also yields less accurate results compared to the other empirical peer prediction mechanisms.

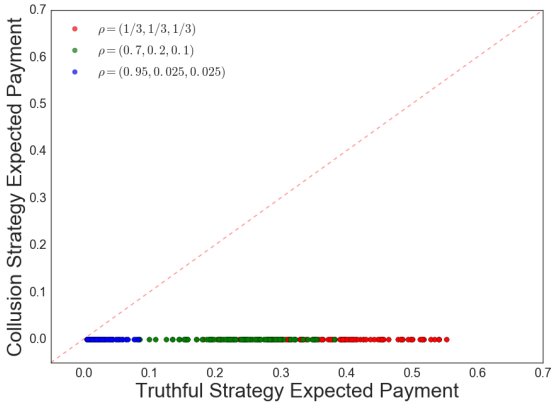
Robustness of ESRM against Strategic Reporting



(a) Expected Payment of Truthful Strategy vs Random Reporting Strategy



(b) Expected Payment of Truthful Strategy vs Signal-Prior Reporting Strategy



(c) Expected Payment of Truthful Strategy vs Collusion Strategy

Figure 4.11: Empirical Scoring Rule Mechanism: Robustness

Similar to the experiments for EPPM and kPPM, we test the robustness of ESRM against various reporting strategies. We fix the number of items to $N = 1000$ and the number agents to $I = 12$. Each agent has her own unique confusion matrix that is sampled from Λ.

Figure 4.11 shows the expected payments of all three reporting strategies — random, Signal-prior, and collusion — against the truthful reporting strategy. While all three reporting strategies have lower expected payments than the truthful reporting strategy, the pattern is quite different from those we observed in EPPM and

kPPM.

The payment rule of ESRM is the difference between the quadratic score of the agent's state posteriors and the quadratic score of the state priors:

$$x_i(r_{n,i}, \hat{t}_n) = R(\hat{\boldsymbol{\pi}}^i_{r_{n,i}}, \hat{t}_n) - R(\hat{\boldsymbol{\rho}}, \hat{t}_n) \tag{4.20}$$

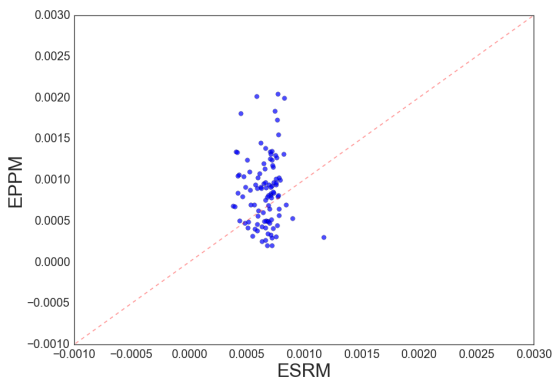
Intuitively, ESRM rewards the agent more if she provides more information about the state of an item than the baseline knowledge, which is the state priors. Thus, agents are incentivized to not only report truthfully but are also incentivized to make more effort to provide accurate reports to the mechanism. This interpretation of ESRM is attractive for human computation systems because it also means that the mechanism rewards the agent who contributes more to the accurate inferrence of the state of the items. Hence, ESRM aligns the incentives of the agent whose goal is to maximize her expected payment with the goal of human computation systems, which is to accurately infer the true states of the items.

Consequently, since strategic reports do not provide additional information about any item, ESRM punishes the agents who adopt such reporting strategies with near 0 payments. This outcome is demonstrated in Figure 4.11.

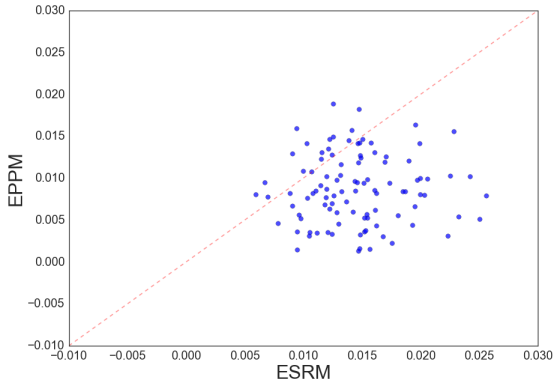
Performance of ESRM in the CommonBelief Model

The Empirical Scoring Rule Mechanism is also applicable to the CommonBelief model. Here, we compare the expected payments of ESRM against those of EPPM in the CommonBelief model.

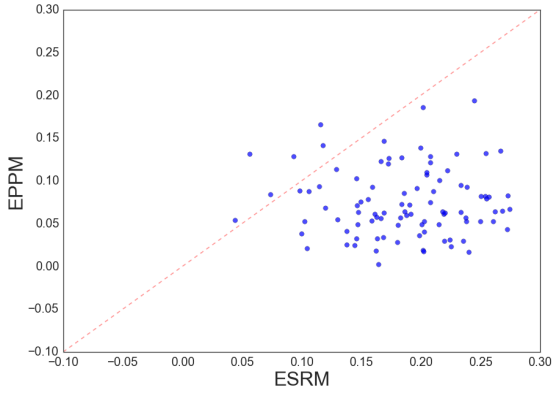
First, in Figure 4.12, we compare the accuracies of the expected payments of the two mechanisms for different number of data points. We compare three different combination of N and I — (1000×100), (100×10), and (10×2). Recall from Chapter 3 Section 3.2.3 that the EM algorithm accurately estimates the confusion matrix and the true states when $N = 1000$ and $I = 100$. In contrast, when $N = 10$ and $I = 2$, the EM algorithm exhibited high errors in estimating the confusion matrix and the



(a) $N = 1,000 \times I = 100$



(b) $N = 100 \times I = 10$



(c) $N = 10 \times I = 2$

Figure 4.12: ESRM vs EPPM: Absolute Difference in Expected Payment

true states.

As a reminder, the accuracy of the mechanism's expected payment refers to the absolute difference between the mechanism's expected payment and the theoretical counterpart where the designer has full knowledge of the belief model. For the theoretical counterpart of ESRM, we assume that the mechanism designer also has full knowledge of the true states.

Figure 4.12 shows the comparison between the accuracy of expected payments of ESRM and EPPM in the CommonBelief model. As shown in Figure 4.12a, where there is a large number of data points $N = 1000$ and $I = 100$, then the expected payments of both mechanisms are close to the theoretical payments. Moreover, ESRM

exhibits smaller variance for the absolute difference between the expected payments.

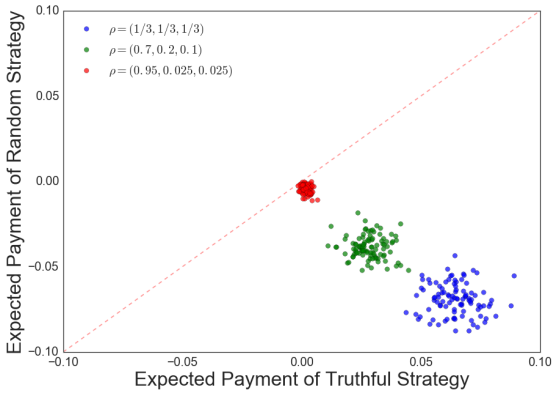
However, as the number of data points decreased to 100×10, the absolute difference started to rise for both mechanisms with the difference increasing slighter more for ESRM than EPPM. At $N = 10$ and $I = 2$, as shown in Figure 4.12c, there is a large increase in the absolute difference between the expected payments of the theoretical mechanisms and their empirical counterpart. In addition, the absolute difference of expected payments of ESRM exhibits higher variance.

Whereas EPPM only relies on the estimate of the confusion matrix Θ and the state priors $\boldsymbol{\rho}$, ESRM must also recover the true states $\boldsymbol{t}$ on top of Θ and $\boldsymbol{\rho}$. Consequently, where there is a shortage of data, the expected payments of ESRM may deviate more from the theoretical payments than those of EPPM.

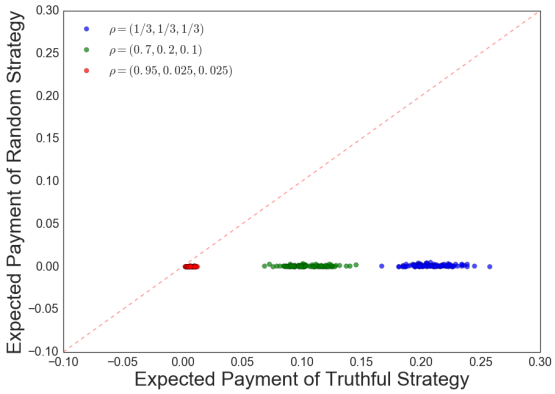
We also examine the robustness of the two mechanisms against the different reporting strategies in the CommonBelief model. We fix the number of items $N = 1000$ and the number of agents $I = 12$ and compared the ratios of the expected payments of strategic reporting over those of truthful reporting. Figure 4.13 summarizes our findings.

Both mechanisms punish colluding agents with zero payments. However, for other strategies, we observe that the two mechanisms punishes strategic reporting differently. First, EPPM is more punishing with negative expected payments with high variance. On the other hand, ESRM pays the agents approximately zero payments with low variance.

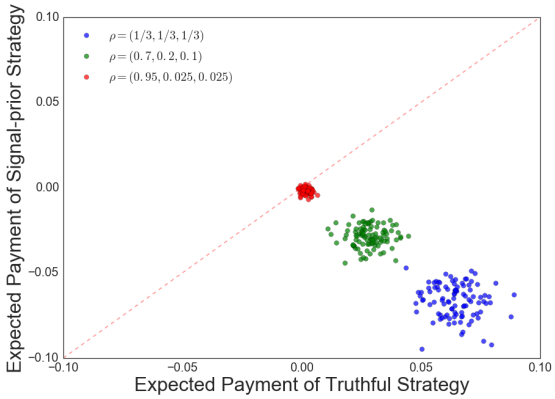
All in all, the simulated data shows that with a sufficiently large number of data points, ESRM can also incentivize the agents to report truthfully in the CommonBelief model just as well as EPPM. One drawback of ESRM is that because the mechanism relies on both the parameters of the model, Θ and $\boldsymbol{\rho}$, and the recovered states $\boldsymbol{t}$ it appears to be more vulnerable to less accurate payments than EPPM in settings with a smaller number of data points.



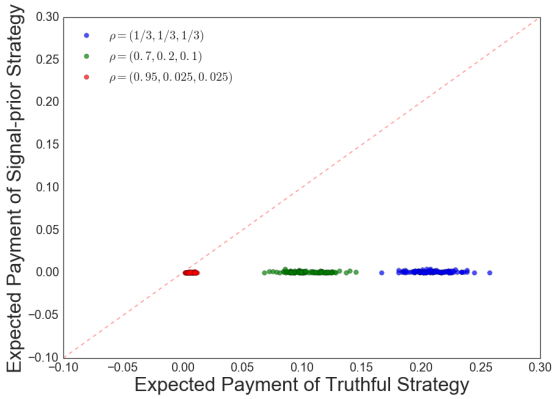
(a) EPPM: Truthful vs Random



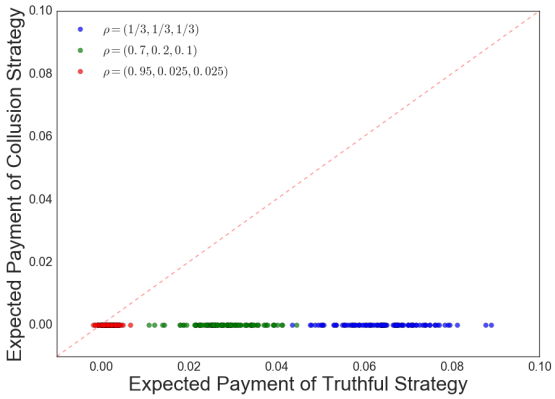
(b) ESRM: Truthful vs Random



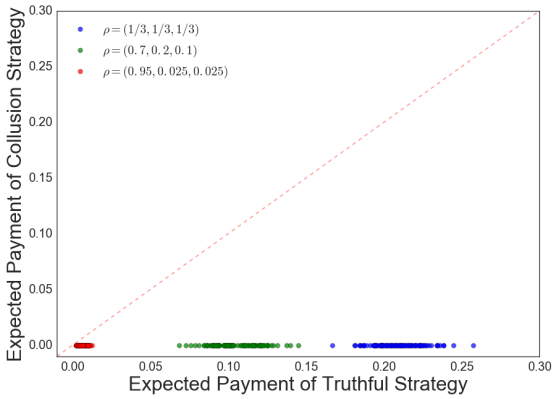
(c) EPPM: Truthful vs Signal-prior



(d) ESRM: Truthful vs Signal-prior



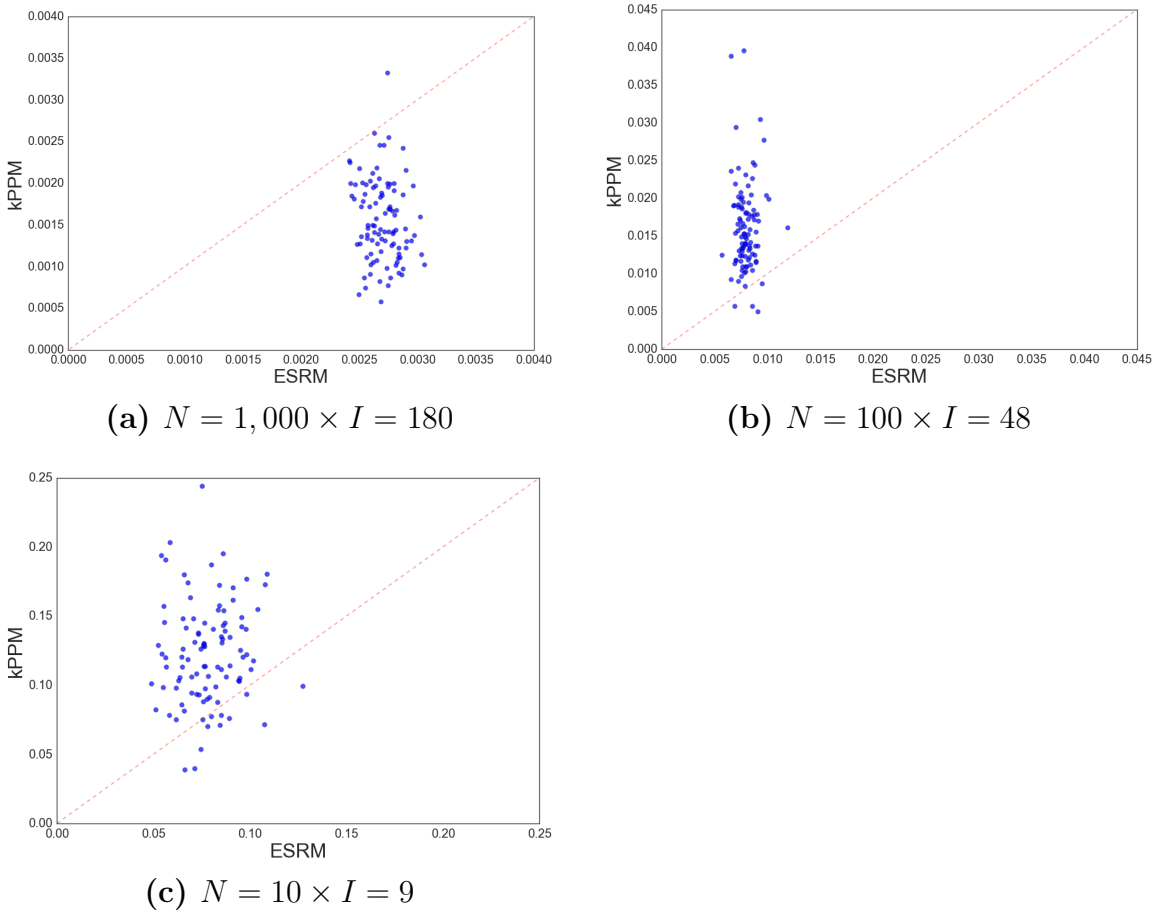
(e) EPPM: Truthful vs Collusion



(f) ESRM: Truthful vs Collusion

Figure 4.13: ESRM vs EPPM: Expected Payments of Truthful Strategy vs Other Strategies

Performance of ESRM in the GroupBelief Model



(a) $N = 1,000 \times I = 180$

(b) $N = 100 \times I = 48$

(c) $N = 10 \times I = 9$

Figure 4.14: ESRM vs kPPM: Absolute Difference in Expected Payment

ESRM is also applicable to the GroupBelief model. Here, we examine the accuracy of ESRM versus kPPM in the CommonBelief model.

First, we compare the accuracies of the expected payments of the two mechanisms for different number of data points. Similar to the earlier experiment in the CommonBelief model, we compare three different combination of N and I — (1000×180), (100×48), and (10×9). Recall from Chapter 3 Section 3.4.3 that the EM algorithm and the k-Means-Confusion algorithm can accurately estimate the common confusion matrices and the true states when $N = 1000$ and $I \geq 120$. In contrast, when $N = 10$ and $I = 9$ the two algorithms exhibited high errors in estimating

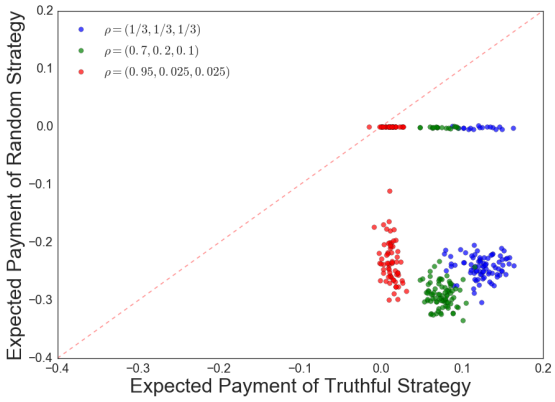
the confusion matrices, the group membership, and the true states.

The outcomes of the experiments are quite similar to those observed for the CommonBelief model. We briefly comment on the outcomes.

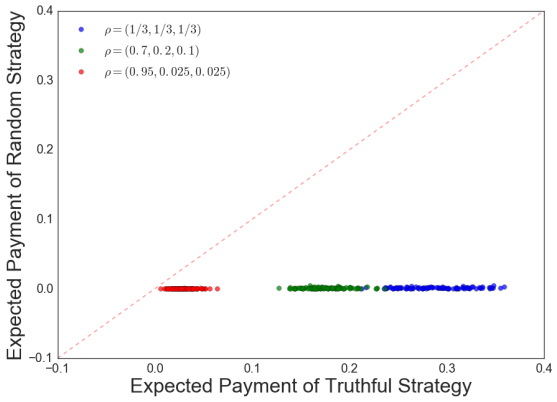
Figure 4.14 shows the absolute difference between empirical scores versus the theoretical scores to highlight the accuracy of the two mechanisms. We notice that both kPPM and ESRM are highly accurate when there is a large number of data points, (1000×180); however, their accuracies progressively deteriorates as the number of data points decreases.

In contrast to the experiment in the CommonBelief model, the outcome of this experiment does not necessarily show larger variance for ESRM as the number of data point decreases. Because kPPM also relies on the group membership estimate Γ, it too is vulnerable to a lower number of data points; therefore, we do not see greater variance of the absolute differences in expected payments for ESRM.

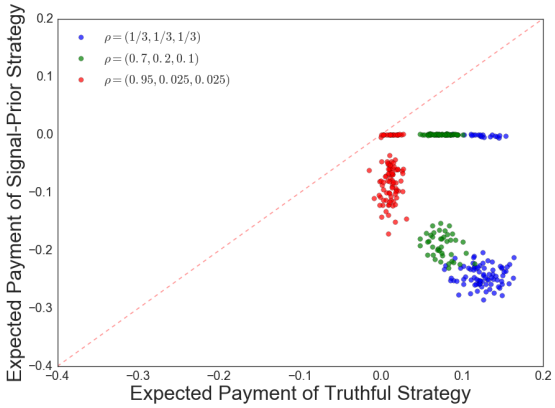
Finally, in Figure 4.15, we see the the expected payments for truthful reporting strategy versus other reporting strategies of an Intermediate agent. Similar to the experiment for the CommonBelief model, we see that the two mechanisms punish strategic reporting quite differently. Nevertheless, for both mechanisms, truthful reporting strategy appears to be the dominant strategy over the three other strategies. Thus, we establish that ESRM is applicable to the GroupBelief model and that ESRM aligns the incentives of the agents to truthful reporting just as well as kPPM in this model.



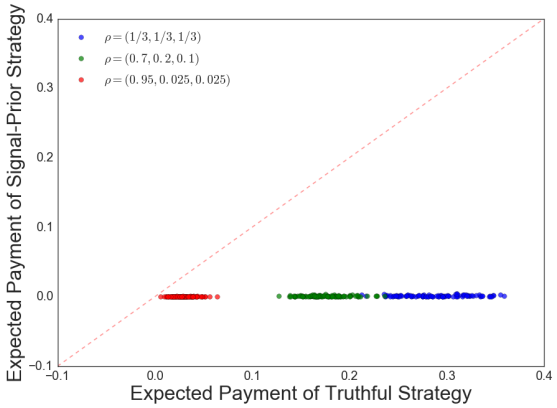
(a) kPPM: Truthful vs Random



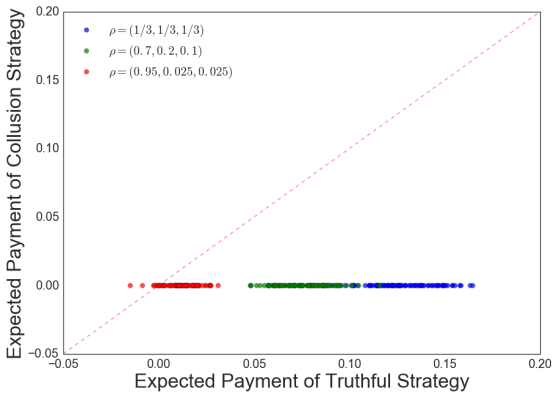
(b) ESRM: Truthful vs Random



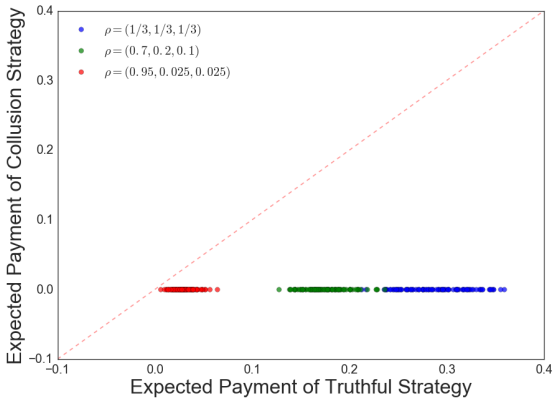
(c) kPPM: Truthful vs Signal-prior



(d) ESRM: Truthful vs Signal-prior



(e) kPPM: Truthful vs Collusion



(f) ESRM: Truthful vs Collusion

Figure 4.15: ESRM vs kPPM: Expected Payments of Truthful Strategy vs Other Strategies

Chapter 5: Summary and Conclusions

In this thesis, I explored existing peer prediction mechanisms from the information elicitation literature and models and algorithms for inferring error-rates of workers in human computation from the machine learning literature. I combined the concepts from these two different disciplines to present an unified approach to resolving the incentive-alignment problem and noisy-data problem in human computation systems.

I started with the CommonBelief model, which maintained the homogeneous agent assumption originally introduced in the base model of peer prediction. I progressively relaxed the homogeneity assumption as I introduced the GroupBelief model and the PrivateBelief model. For each model, I introduced an applicable empirical peer prediction mechanism and demonstrated through simulated data that the mechanism is robust against various reporting strategies. I also showed that the Empirical Scoring Rule Mechanism is applicable to not only the PrivateBelief model but also the CommonBelief model and the GroupBelief model. Through simulated data, I showed that ESRM is a robust mechanism in all three domains.

I believe there are several interesting research directions that one can take in the empirical methods in peer prediction.

In this thesis, I empirically demonstrated using simulated data the robustness of the empirical peer prediction mechanisms against strategic reporting. One interesting future direction would be to apply the *probably approximately correct learning* (PAC learning) framework to rigorously examine the robustness of the empirical peer prediction mechanism.

All three empirical peer prediction mechanisms require that they receive complete report for all items from every participating agents. While in theory a mechanism can indefinitely withhold payments, in real world human computation systems, workers typically demand timely payment for the completed tasks. Therefore, delaying payments until all the workers complete every task may not be applicable in practice.

An interesting future direction for the empirical methods in peer prediction would be to incorporate other machine learning techniques, besides the EM algorithm and the k-means algorithm, to quickly and accurately estimate the belief models of the agents so that the agents can be paid in timely fashion. Ideally, a mechanism should accurately compute necessary parameters of the model and pay an agent as soon as she completes a single task.

Also, the three empirical peer prediction mechanisms compute the payments based on the error-rates of the agents. They do not consider attributes of the tasks. In human computation, some tasks are harder than others, and these difficult tasks should carry higher reward for the workers. Exploring models that incorporate heterogeneity of the tasks is an interesting research direction. The JointConfusion model (Lakkaraju et al., 2015) is a promising candidate for exploration in this direction.

The marriage between machine learning algorithms and peer prediction mechanisms is still in its infancy. The three empirical peer prediction mechanisms in this thesis merely scratch the surface of many more interesting empirical methods in peer prediction to be introduced in the future. We should look forward to see how these innovative mechanisms improve human computation systems and be excited about what newly improved human computation systems can contribute to the broader scientific community.

References

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Chawla, S., & Gionis, A. (2013). k-means--: A unified approach to clustering and outlier detection. In *Proceedings of the 2013 SIAM International Conference on Data Mining*.

Dawid, A., & Skene, A. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 20–28.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B*, *39*(1), 1–38.

Ghahramani, Z., & Kim, H. c. (2003). *Bayesian Classifier Combination* (Tech. Rep.). University College London.

Grier, D. A. (2005). *When Computers Were Human*. Princeton, NJ: Princeton University Press.

Hanson, R. (2007). Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation. *Journal of Prediction Markets*, *1*(1), 3–15.

Ipeirotis, P. G., & Paritosh, P. K. (2011). *Managing Crowdsourced Human Computation: A Tutorial.* Retrieved from `http://www.slideshare.net/ipeirotis/managing-crowdsourced-human-computation`

Joglekar, M., Garcia-Molina, H., & Parameswaran, A. (2013). Evaluating the Crowd with Confidence. In *Proceedings of the 19th ACM SIGKDD international con-*

ference on Knowledge discovery and data mining.

Johnson, S., Miller, N., Pratt, J., & Zeckhauser, R. (2002). *Efficient Design with Interdependent Valuations and an Informed Center* (Tech. Rep. No. RWP02-025). Kennedy School Working Paper.

Jurca, R., & Faltings, B. (2005). Enforcing Truthful Strategies in Incentive Compatible Reputation Mechanisms. In *Proceedings of the 1st International Workshop on Internet and Network Economics.*

Jurca, R., & Faltings, B. (2009). Mechanisms for Making Crowds Truthful. *Journal of Artificial Intelligence Research*, *34*, 209–253.

Jurca, R., & Faltings, B. (2011). Incentives for Answering Hypothetical Questions. In *Proceedings of the 1st Workshop on Social Computing and User Generated Content.*

Lakkaraju, H., Leskovec, J., Kleinberg, J., & Mullainathan, S. (2015). A Bayesian Framework for Modeling Human Evaluations. In *Proceedings of the 2015 SIAM International Conference on Data Mining.*

Liu, C., & Wang, Y. (2012). TrueLabel+Confusion, A Spectrum of Probabilistic Models in Analyzing Multiple Ratings. In *Proceedings of the 29 th International Conference on Machine Learning.*

Miller, N., Resnick, P., & Zeckhauser, R. (2005). Eliciting Informative Feedback: The Peer-Prediction Method. *Management Science*, *51*(9), 1259–1373.

Parkes, D. C., & Seuken, S. (2017). *Economics and Computation: A Design Approach.* Cambridge, England: Cambridge University Press.

Prelec, D. (2004). A Bayesian Truth Serum for Subjective Data. *Science*, *306*(5695), 462–466.

Radanovic, G., & Faltings, B. (2013). A Robust Bayesian Truth Serum for Non-binary Signals. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence.*

Radanovic, G., & Faltings, B. (2015). Incentives for Subjective Evaluations with Private Beliefs. In *Proceedings of the 29th AAAI Conference on Artificial In-*

telligence.

Selten, R. (1998). Axiomatic Characterization of the Quadratic Scoring Rule. *Experimental Economics*, *1*, 43–62.

Shnayder, V., Agarwal, A., Frongillo, R., & Parkes, D. C. (2016). *Informed Truthfulness in Multi-Task Peer Prediction.* Retrieved from `http://arxiv.org/abs/1603.03151`

von Ahn, L., & Dabbish, L. (2004). Labeling Images with A Computer Game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.*

Waggoner, B., & Chen, Y. (2013). Information Elicitation Sans Verification. In *Proceedings of the 3rd Workshop on Social Computing and User Generated Content.*

Welinder, P., Brandson, S., Belongie, S., & Perona, P. (2010). The Multidimensional Wisdom of Crowds. In *Advances in Neural Information Processing Systems.*

Welinder, P., & Perona, P. (2010). Online crowdsourcing: rating annotators and obtaining cos-effective labels. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*

Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., & Movellan, J. (2009). Whose Vote Should Count More: Optimal Integrations of Labels from Labelers of Unknown Expertise. In *Advances in Neural Information Processing Systems.*

Witkowski, J. (2014). *Robust Peer Prediction Mechanisms* (Unpublished doctoral dissertation). Albert-Ludwigs-Universität Freiburg Institut Für Informatik, Freiburg im Breisgau, Germany.

Witkowski, J., & Parkes, D. C. (2012). Peer Prediction Without a Common Prior. In *Proceedings of the 13th ACM Conference on Electronic Commerce.*

Zhang, P., & Chen, Y. (2014). Elicitability and Knowledge-Free Elicitation with Peer Prediction. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems.*

Glossary

Mechanism Design Often called "inverse game theory," *mechanism design* is the study of approaches to engineering incentives in strategic settings where the agents behave rationally.

Nash Equilibrium A strategy profile (within game theory) $\boldsymbol{\Psi} = \{\Psi^i, ..., \Psi^I\}$ of I agents is a *Nash equilibrium* if, for every agent $i \in \mathcal{I}$, Ψ^i is the best response to Ψ^{-i}. Nash equilibrium is a stable strategy profile such that no agent would benefit, or increase his utility, by changing his strategy.

Correlated Equilibrium A solution concept in game theory that generalizes Nash equilibrium. In correlated equilibrium, each agent chooses her action according to her private observation of the same public signal. In peer prediction mechanism, this public signal is the true state of the world.

Human Computation A computational technique in which certain tasks, usually microtasks, in computation is outsourced to humans. Unlike the traditional computation scheme where human beings employ the machines to solve a problem, in human computation, the machine seeks input from human workers, serving as a step in an algorithm that collects, interprets, and integrates the inputs into the final solution.

Confusion Matrix Also known as contingency table, or error matrix, the confusion matrix allows visualization of the performance of classification tasks. For exam-

ple, given the confusion matrix generated from an image processing software,

	Predicted Class	
True Class	*Dog*	*Cat*
Dog	8	2
Cat	1	9

we observe that the software correctly labeled 8 images and mislabeled 2 images out of the total of 10 images of dogs.

Maximum Likelihood Estimation A widely used method of estimating the parameters of a statistical model given data (i.e. the likelihood $P(\boldsymbol{D}|\theta)$)

$$\theta_{MLE} = \arg\max_{\theta} P(\boldsymbol{D}|\theta)$$

.