Peer Prediction with Heterogeneous Users

ARPIT AGARWAL, University of Pennsylvania
DEBMALYA MANDAL, Harvard University
DAVID C. PARKES, Harvard University
NISARG SHAH, University of Toronto

Peer prediction mechanisms incentivize agents to truthfully report their signals, in the absence of a verification mechanism, by comparing their reports with those of their peers. Prior work in this area is essentially restricted to the case of homogeneous agents, whose signal distributions are identical. This is limiting in many domains, where we would expect agents to differ in taste, judgment and reliability. Although the Correlated Agreement (CA) mechanism [34] can be extended to handle heterogeneous agents, there is a new challenge of efficiently estimating agent signal types. We solve this problem by clustering agents based on their reporting behavior, proposing a mechanism that works with clusters of agents and designing algorithms that learn such a clustering. In this way, we also connect peer prediction with the Dawid and Skene [6] literature on latent types. We retain the robustness against coordinated misreports of the CA mechanism, achieving an approximate incentive guarantee of ε-informed truthfulness. We show on real data that this incentive approximation is reasonable in practice, even with a small number of clusters.

Additional Key Words and Phrases: Peer Prediction; Information Elicitation; Clustering; Tensor Decompsition

ACM Reference Format:
Arpit Agarwal, Debmalya Mandal, David C. Parkes, and Nisarg Shah. 2019. Peer Prediction with Heterogeneous Users. *ACM Transactions on Economics and Computation* 0, 0, Article 00 (2019), 28 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Peer prediction is the problem of information elicitation without verification. Peer prediction mechanisms incentivize users to provide honest reports when the reports cannot be verified, either because there is no objective ground truth or because it is costly to acquire the ground truth. Peer prediction mechanisms leverage correlation in the reports of peers in order to score contributions. The main challenge of peer prediction is to incentivize agents to put effort to obtain a signal or form an opinion and then honestly report to the system. In recent years, peer prediction has been studied in several domains, including peer assessment in massively open online courses (MOOCs) [Gao et al., 2016, Shnayder and Parkes, 2016], for feedback on local places in a city [Mandal et al., 2016], and in the context of collaborative sensing platforms [Radanovic and Faltings, 2015a].

We thank Adish Singla and Dengyong Zhou for providing us the Stanford Dogs (Dogs) and the Waterbird dataset (WB). We also acknowledge the useful feedback from anonymous reviewers.
This work is supported in part by Google, the SEAS TomKat fund, and NSF grant CCF-1301976.
Authors' email addresses: Arpit Agarwal: aarpit@seas.upenn.edu; Debmalya Mandal: dmandal@g.harvard.edu; David C. Parkes: parkes@eecs.harvard.edu; Nisarg Shah: nisarg@cs.toronto.edu.
Authors' addresses: Arpit Agarwal, University of Pennsylvania; Debmalya Mandal, Harvard University; David C. Parkes, Harvard University; Nisarg Shah, University of Toronto.

XXXX-XXXX/2019/00-ART00 $15.00
https://doi.org/0000001.0000001

The simplest peer prediction mechanism is *output agreement*, which pairs up two users and rewards them in the event that their reports agree (the ESP game [von Ahn and Dabbish, 2004] can be interpreted this way). However, output agreement is not incentive aligned for reports of *a priori* unlikely signals. As a result, there has been a lot of attention in recent years on finding methods that work more generally and provide robustness to coordinated misreports.

All existing, general methods are essentially restricted to settings with homogeneous participants, whose signal distributions are identical. This is a poor fit with many suggested applications of peer prediction. Consider for example, the problem of peer assessment in MOOCs. DeBoer et al. [2013] and Wilkowski et al. [2014] observe that students differ based on their geographical locations, educational backgrounds, and level of commitment, and indeed the heterogeneity of assessment is clear from a study of Coursera data [Kulkarni et al., 2015]. Simpson et al. [2013] observed that the users participating in a *citizen science* project can be categorized into five distinct groups based on their behavioral patterns in classifying an image as a Supernovae or not. A similar problem occurs in determining whether a news headline is offensive or not. Depending on which social community a user belongs to, we should expect to get different opinions [Zafar et al., 2016]. Moreover, Allcott and Gentzkow [2017] report that leading to the 2016 U.S. presidential election, people were more likely to believe the stories that favored their preferred candidate; Fourney et al. [2017] find that there is very low connectivity among Trump and Clinton supporters on social networks, which leads to confirmation bias among the two groups and clear heterogeneity about how they believe whether a news is “fake” or not.

One obstacle to designing peer prediction mechanisms for heterogeneous agents is an impossibility result. No mechanism can provide strict incentive for truth-telling to a population of heterogeneous agents without knowledge of their signal distributions [Radanovic and Faltings, 2015c]. This negative result holds for minimal mechanisms, which only elicit signals and not beliefs from agents. One way to alleviate this problem, without going to non-minimal mechanisms, is to use reports from the agents across multiple tasks to estimate their signal distributions. This is our goal: we want to design minimal peer prediction mechanisms for heterogeneous agents that use reports from the agents for both learning and scoring. We also want to provide robustness against coordinated misreports.

As a starting point, we consider the *correlated agreement* (CA) mechanism proposed by Shnayder et al. [2016]. If the agents are homogeneous and the designer has knowledge of their joint signal distribution, the CA mechanism is *informed truthful*, i.e. no strategy profile, even if coordinated, can provide more expected payment than truth-telling, and the expected payment under an uninformed strategy (where an agent’s report is independent of her signal) is strictly less than the expected payment under truth-telling. These two properties remove any incentive for coordinated deviations and strictly incentivize the agents to put effort in acquiring signals, respectively. In a detail-free variation, in which the designer learns the signal distribution from reports, approximate incentive alignment is provided (still maintaining the second property as a strict guarantee.) The detail-free CA mechanism can be extended to handle agent heterogeneity, but a naïve approach would require learning the joint signal distributions between every pair of agents, and the total number of reports that need to be collected would be prohibitive for many settings.

Our contributions: We design the first minimal and detail-free mechanism for peer prediction with heterogeneous agents, where the learning component has sample complexity that is only linear in the number of agents, while providing an incentive guarantee of approximate informed truthfulness. Like the CA mechanism, this is a multi-task mechanism in that each agent makes reports across multiple tasks. Since our mechanism has a learning component, the task assignments to agents should be such that both the goals of incentive alignment and learning are simultaneously achieved. We consider two assignment schemes under which these goals can be achieved and analyze the sample complexity of our methods for these schemes.

The mechanism clusters the agents based on their reported behavior and learns the pairwise correlations between these clusters. The clustering introduces one component of the incentive approximation, and could be problematic in the absence of a good clustering such that agents within a cluster behave similarly. Using eight real-world datasets, which contain reports of users on crowdsourcing platforms for multiple labeling tasks, we show that the clustering error is small in practice even when using a relatively small number of clusters. The second component of the incentive approximation stems from the need to learn the pairwise correlations between clusters; this component can be made arbitrarily small using a sufficient number of signal reports.

Another contribution of this work is to connect, we believe for the first time, the peer prediction literature with the extensive and influential literature on latent, confusion matrix models of label aggregation [Dawid and Skene, 1979b]. The Dawid-Skene model assumes that signals are generated independently, conditional on a latent attribute of a task and according to an agent's confusion matrix. We cluster the agents based on their confusion matrices and then estimate the average confusion matrices within clusters using recent developments in tensor decomposition algorithms [Anandkumar et al., 2014, Zhang et al., 2016]. These average confusion matrices are then used to learn the pairwise correlations between clusters and design reward schemes to achieve approximate informed truthfulness.

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

1.1 Related Work

We focus our discussion on related work about minimal mechanisms, but remark that we are not aware of any non-minimal mechanisms (following from the work of Prelec [2004]) that handle agent heterogeneity. Miller et al. [2005] introduce the peer prediction problem, and proposed an incentive-aligned mechanism for the single-task setting. However, their mechanism requires knowledge of the joint signal distribution and is vulnerable to coordinated misreports. In regard to coordinated misreports, Jurca et al. [2009] show how to eliminate uninformative, pure-strategy equilibria through a three-peer mechanism, and Kong et al. [2016] provide a method to design robust, single-task, binary signal mechanisms (but need knowledge of the joint signal distribution). Frongillo and Witkowski [2017] provide a characterization of minimal (single task) peer prediction mechanisms.

Witkowski and Parkes [2013] introduce the combination of learning and peer prediction, coupling the estimation of the signal prior together with the shadowing mechanism. Some results make use of reports from a large population. Radanovic and Faltings [2015b], for example, establish robust incentive properties in a large-market limit where both the number of tasks and the number of agents assigned to each task grow without bound. Radanovic et al. [2016] provide complementary theoretical results, giving a mechanism in which truthfulness is the equilibrium with the highest payoff in the asymptote of a large population and with a structural property on the signal distribution.

Dasgupta and Ghosh [2013] show that robustness to coordinated misreports can be achieved for binary signals in a small population by using a multi-task mechanism. The idea is to reward agents if they provide the same signal on the same task, but punish them if one agent's report on one task is the same as another's on a different task. The Correlated Agreement (CA) mechanism [Shnayder et al., 2016] generalizes this mechanism to handle multiple signals, and uses reports to estimate the correlation structure on pairs of signals without compromising incentives. In related work, Kong and Schoenebeck [2016] show that many peer prediction mechanisms can be derived within a single information-theoretic framework. Their results use different technical tools than those used by Shnayder et al. [2016], and also include a different multi-signal generalization of the Dasgupta-Ghosh mechanism that provides robustness against coordinated misreports in the limit of a large number of tasks. Shnayder et al. [2016] adopt replicator dynamics as a model of population learning in peer prediction, and confirm that these multi-task mechanisms (including the mechanism by Kamble et al. [2015]) are successful at avoiding uninformed equilibria.

There are very few results on handling agent heterogeneity in peer prediction. For binary signals, the method of Dasgupta and Ghosh [2013] is likely to be an effective solution because their assumption on correlation structure will tend to hold for most reasonable models of heterogeneity. But it will break down for more than two signals, as explained by Shnayder et al. [2016]. Moreover, although the CA mechanism can in principle be extended to handle heterogeneity, it is not clear how the required statistical information about joint signal distributions can be efficiently learned and coupled with an analysis of approximate incentives. For a setting with binary signals and where each task has one of a fixed number of latent types, Kamble et al. [2015] design a mechanism that provides strict incentive compatibility for a suitably large number of heterogeneous agents, and when the number of tasks grows without bound (while allowing each agent to only provide reports on a bounded number of tasks). Their result is restricted to binary signals, and requires a strong regularity assumption on the generative model of signals.

Finally, we consider only binary effort of a user, i.e. the agent either invests effort and receives an informed signal or does not invest effort and receives an uninformed signal. Shnayder et al. [2016] work with the binary effort setting and provide strict incentive for being truthful. Therefore, as long as the mechanism designer is aware of the cost of investing effort, the payments can be scaled to cover the cost of investing effort. The importance of motivating effort in the context of peer prediction has also been considered by Liu and Chen [2017b] and Witkowski et al. [2013].[1] See Mandal et al. [2016] for a setting with heterogeneous tasks but homogeneous agents. Liu and Chen [2017a] also designed single-task peer prediction mechanism for the same setting but only when each task is associated with a latent ground truth,

2 MODEL

Let notation $[t]$ denote $\{1, \ldots, t\}$ for $t \in \mathbb{N}$. We consider a population of agents $P = [\ell]$, and use indices such as p and q to refer to agents from this population. There is a set of tasks $M = [m]$. When an agent performs a task, she receives a signal from $N = [n]$. As mentioned before, we assume that the effort of an agent is binary. We also assume that the tasks are *ex ante* identical, that is, the signals of an agent for different tasks are sampled i.i.d. We use S_p to denote the random variable for the signal of agent p for a task.

We work in the setting where the agents are heterogeneous, i.e., the distribution of signals can be different for different agents. We say that agents vary by *signal type*. Let $D_{p,q}(i, j)$ denote the

[1] Cai et al. [2015] work in a different model, showing how to achieve optimal statistical estimation from data provided by rational agents. They only focus on the cost of effort. They do not consider possible misreports, and thus their mechanism is also vulnerable to coordinated misreports.

joint probability that agent p receives signal i while agent q receives signal j on a random task. Let $D_p(i)$ and $D_q(j)$ denote the corresponding marginal probabilities. We define the *Delta matrix* $\Delta_{p,q}$ between agents p and q as

$$\Delta_{p,q}(i,j) = D_{p,q}(i,j) - D_p(i) \cdot D_q(j), \ \forall i,j \in [n]. \tag{1}$$

Example 2.1. For two agents p and q, consider the following joint signal distribution $D_{p,q}$ is

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

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

$$\Delta_{p,q} = \begin{bmatrix} 0.4 & 0.1 \\ 0.1 & 0.4 \end{bmatrix} - \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \cdot \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}.$$

Shnayder et al. [2016] show that it is without loss of generality for the class of mechanisms we study in this paper to assume that an agent's strategy is uniform across different tasks. Given this, let R_p denote the random variable for the report of agent p for a task. The strategy of agent p, denoted F^p, defines the distribution of reports for each possible signal i, with $F^p_{ir} = \Pr(R_p = r | S_p = i)$. The collection of agent strategies, denoted $\{F^p\}_{p \in P}$, is the *strategy profile.* A strategy of agent p is *informed* if there exist distinct $i, j \in [n]$ and $r \in [n]$ such that $F^p_{ir} \neq F^p_{jr}$, i.e., if not all rows of F^p are identical. We say that a strategy is *uninformed* otherwise.

2.1 Multi-Task Peer Prediction

We consider *multi-task peer prediction* mechanisms. In particular, we adapt the *correlated agreement* (CA) mechanism [Shnayder et al., 2016] to our setting with heterogeneous agents. For every pair of agents $p, q \in P$, we define a *scoring matrix* $S_{p,q} : [n] \times [n] \to \mathbb{R}$ as a means of scoring agent reports.[2] We randomly divide (without the knowledge of an agent) the set of tasks performed by each agent into nonempty sets of *bonus tasks* and *penalty tasks.* For agent p, we denote the set of her bonus tasks by M^p_1 and the set of her penalty tasks by M^p_2.

To calculate the payment to an agent p for a bonus task $t \in M^p_1$, we do the following:

(1) Randomly select an agent $q \in P \setminus \{p\}$ such that $t \in M^q_1$, and the set $M^p_2 \cup M^q_2$ has at least 2 distinct tasks, and call q the *peer* of p.
(2) Pick tasks $t' \in M^p_2$ and $t'' \in M^q_2$ randomly such that $t' \neq t''$ (t' and t'' are the penalty tasks for agents p and q respectively)
(3) Let the reports of agent p on tasks t and t' be r^t_p and $r^{t'}_p$, respectively and the reports of agent q on tasks t and t'' be r^t_q and $r^{t''}_q$ respectively.
(4) The payment of agent p for task t is then $S_{p,q}(r^t_p, r^t_q) - S_{p,q}(r^{t'}_p, r^{t''}_q)$.

The total payment to an agent is the sum of payments for the agent's bonus tasks. We assume that from an agent's perspective, every other agent is equally likely to be her peer. This requires agents not to know each other's task assignments. In Section 4 we give two task assignment schemes which satisfy this property that from an agent's perspective all peers are equally likely.

[2]Multi-task peer prediction mechanisms for homogeneous agents need a single Delta and a single scoring matrix.

The expected payment to agent p for any bonus task performed by her, equal across all bonus tasks as the tasks are *ex ante* identical, is given as

$$\begin{aligned} u_p(F^p, \{F^q\}_{q\neq p}) &= \frac{1}{\ell-1}\sum_{q\neq p}\left\{\sum_{i,j} D_{p,q}(i,j)\sum_{r_p,r_q} F^p_{ir_p}F^q_{jr_q}S_{p,q}(r_p,r_q)\right.\\ &\qquad \left. - \sum_i D_p(i)\sum_{r_p} F^p_{ir_p}\sum_j D_q(j)\sum_{r_q} F^q_{jr_q}S_{p,q}(r_p,r_q)\right\}\\ &= \frac{1}{\ell-1}\sum_{q\neq p}\left\{\sum_{i,j}\left(D_{p,q}(i,j) - D_p(i)D_q(j)\right)\sum_{r_p,r_q} F^p_{ir_p}F^q_{jr_q}S_{p,q}(r_p,r_q)\right\}\\ &= \frac{1}{\ell-1}\sum_{q\neq p}\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q} F^p_{ir_p}F^q_{jr_q}S_{p,q}(r_p,r_q) \end{aligned}$$

Using the fact that there exist optimal solutions to linear functions that are extremal, it is easy to show that there always exists an optimal strategy for agent p that is deterministic (see also Shnayder et al. [2016]).

Lemma 2.2. *For every player p, and any strategies of others, there always exists an optimal strategy F^p maximizing u_p that is deterministic.*

Hereafter, we assume without loss of generality that agent strategies are deterministic. For a deterministic strategy F^p of agent p, we will slightly abuse notation and write F^p_i to denote the signal reported by agent p when she observes signal i. For a deterministic strategy profile $\{F^q\}_{q\in P}$, the expected payment to agent p is

$$u_p(F^p, \{F^q\}_{q\neq p}) = \frac{1}{\ell-1}\sum_{q\neq p}\sum_{i,j}\Delta_{p,q}(i,j)\cdot S_{p,q}(F^p_i, F^q_j). \tag{2}$$

2.2 Informed Truthfulness

Following Shnayder et al. [2016], we define the notion of approximate informed truthfulness for a multi-task peer prediction mechanism.

Definition 2.3. (ε-informed truthfulness) We say that a multi-task peer prediction mechanism is ε-informed truthful, for some $\varepsilon \geqslant 0$, if and only if for every strategy profile $\{F^q\}_{q\in P}$ and every agent $p \in P$, we have $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p(F^p, \{F^q\}_{q\neq p}) - \varepsilon$, where $\mathbb{I}$ is the truthful strategy, and $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > u_p(F^p_0, \{F^q\}_{q\neq p})$ where F^p_0 is an uninformed strategy.

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

[3] We do not model the cost of effort explicitly in this paper, but a binary cost model (effort → signal, no-effort → no signal) can be handled in a straightforward way. See Shnayder et al. [2016].

2.3 Agent Clustering

While the natural extension of the detail-free CA mechanism to our setting with heterogeneous agents preserves informed truthfulness, it would require learning the Delta matrix between every pair of agents. In order to learn this from data without additional assumptions, we would require $\Omega(\ell^2)$ samples, which will often be impractical. Rather, the number of reports in a practical mechanism should scale closer to linearly in the number of agents.

In response, we will assume that agents can be (approximately) clustered into a bounded number K of agent signal types, with all agents of the same type having nearly identical signal distributions. Let $G_1, \ldots, G_K$ denote a partitioning of agents into K clusters. With a slight abuse of notation, we also use $G(p)$ to denote the cluster to which agent p belongs.

Definition 2.4. We say that clustering $G_1, \ldots, G_K$ is ε_1*-accurate*, for some $\varepsilon_1 \geqslant 0$, if for every pair of agents $p, q \in P$,

$$\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leqslant \varepsilon_1, \tag{3}$$

where $\Delta_{G(p),G(q)}$ is the *cluster Delta matrix* between clusters $G(p)$ and $G(q)$, defined as the average of the Delta matrices between agents in $G(p)$ and agents in $G(q)$:

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

Example 2.5. Let there be 4 agents p, q, r and s. Let the pairwise Delta matrices be the following

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

$$\Delta_{q,r} = \begin{bmatrix} -0.05 & 0.05 \\ 0.05 & -0.05 \end{bmatrix}, \Delta_{q,s} = \begin{bmatrix} -0.15 & 0.15 \\ 0.15 & -0.15 \end{bmatrix}, \Delta_{r,s} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}$$

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

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

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

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

There is an inverse relationship between the number of clusters K and the clustering accuracy ε_1: the higher the K, the lower the ε_1. In the extreme, we can let every agent be a separate cluster ($K = \ell$), which results in $\varepsilon_1 = 0$. But a small number of clusters is essential for a reasonable sample complexity as we need to learn $O(K^2)$ cluster Delta matrices. For instance, in Example 2.5 we need to learn 3 Delta matrices with clustering, as opposed to 6 without clustering. In Section 4, we give a learning algorithm that can learn all the pairwise cluster Delta matrices with $\tilde{O}(K)$ samples given a clustering of the agents. In Section 5, we show using real-world data that a reasonably small clustering error can be achieved with relatively few clusters.

3 CORRELATED AGREEMENT FOR CLUSTERED, HETEROGENEOUS AGENTS

In this section we define mechanism CAHU, presented as Algorithm 1, which takes as input a clustering as well as estimates of the cluster Delta matrices. Specifically, CAHU takes as input a clustering $G_1, \ldots, G_K$, together with estimates of cluster Delta matrices $\{\overline{\Delta}_{G_s,G_t}\}_{s,t\in[K]}$.

Definition 3.1. We say that a clustering $\{G_s\}_{s\in[K]}$ and the estimates $\{\overline{\Delta}_{G_s,G_t}\}_{s,t\in[K]}$ are $(\varepsilon_1,\varepsilon_2)$-accurate if

- $\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leqslant \varepsilon_1$ for all agents $p,q \in P$, i.e., the clustering is ε_1-accurate, and
- $\|\Delta_{G_s,G_t} - \overline{\Delta}_{G_s,G_t}\|_1 \leqslant \varepsilon_2$ for all clusters $s,t \in [K]$, i.e., the cluster Delta matrix estimates are ε_2-accurate.

When we have a clustering and estimates of the delta matrices which are $(\varepsilon_1,\varepsilon_2)$-accurate, we prove that the CAHU mechanism is $(\varepsilon_1+\varepsilon_2)$-informed truthful. In Section 4, we present algorithms that can learn an ε_1-accurate clustering and ε_2-accurate estimates of cluster Delta matrices.

Throughout the rest of this section, we will use ε_1 to denote the clustering error and ε_2 to denote the learning error. We remark that the clustering error ε_1 is determined by the level of similarity present in agent signal-report behavior, as well as the number of clusters K used, whereas the learning error ε_2 depends on how many samples the learning algorithm sees.

Algorithm 1 Mechanism CAHU

Input:

A clustering $G_1,\ldots,G_K$ such that $\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leqslant \varepsilon_1$ for all $p,q \in P$;
estimates $\{\overline{\Delta}_{G_s,G_t}\}_{s,t\in[K]}$ such that $\|\overline{\Delta}_{G_s,G_t} - \Delta_{G_s,G_t}\|_1 \leqslant \varepsilon_2$ for all $s,t \in [K]$; and
for each agent $p \in P$, her bonus tasks M_1^p, penalty tasks M_2^p, and responses $\{r_b^p\}_{b\in M_1^p \cup M_2^p}$.

Method:

for every agent $p \in P$ **do**
  for every task $b \in M_1^p$ **do** ▹ Reward response r_b^p
    $q \leftarrow$ uniformly at random conditioned on $b \in M_1^q \cup M_2^q$ and (either $|M_2^q| \geqslant 2, |M_2^p| \geqslant 2$ or $M_2^q \neq M_2^p$) ▹ Peer agent
    Pick tasks $b' \in M_2^p$ and $b'' \in M_2^q$ randomly such that $b' \neq b''$ ▹ Penalty tasks
    $S_{G(p),G(q)} \leftarrow \text{Sign}(\overline{\Delta}_{G(p),G(q)})$ †
    Reward to agent p for task b is $S_{G(p),G(q)}\left(r_b^p, r_b^q\right) - S_{G(p),G(q)}\left(r_{b'}^p, r_{b''}^q\right)$
  end for
end for

†Sign$(x) = 1$ if $x > 0$, and 0 otherwise.

3.1 Analysis of CAHU

Because mechanism CAHU uses scoring matrix $S_{G(p),G(q)}$ to reward agent p given peer agent q in place of a separate scoring matrix $S_{p,q}$ for every pair of agents (p,q), the payoff equation (2) transforms to

$$u_p(F^p, \{F^q\}_{q\neq p}) = \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{G(p),G(q)}(F_i^p, F_j^q). \tag{4}$$

Next, we prove a sequence of results that show that mechanism CAHU is $(\varepsilon_1+\varepsilon_2)$-informed truthful when the clustering is ε_1-accurate and the cluster Delta matrix estimates are ε_2-accurate. We first show that the mechanism would be (exactly) informed truthful if it used the Delta matrices between agents instead of using estimates of cluster Delta matrices. In the following we use $u_p^*(\cdot)$ (resp. $u_p(\cdot)$) to denote the utility of agent p under the CA (resp. CAHU) mechanism.

LEMMA 3.2. *For a strategy profile $\{F^q\}_{q\in P}$ and an agent $p \in P$, define*

$$u_p^*(F^p, \{F^q\}_{q\neq p}) = \frac{1}{\ell - 1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(F_i^p, F_j^q),$$

where $S_{p,q}(i,j) = Sign(\Delta_{p,q}(i,j))$ for all $i, j \in [n]$. Then, $u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p^*(F^p, \{F^q\}_{q\neq p})$. Moreover, for any uninformed strategy $F^p = r$, $u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > u_p^*(r, \{F^q\}_{q\neq p})$.

PROOF. Note that

$$\begin{aligned}
u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) &= \frac{1}{\ell - 1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(i,j) \\
&= \frac{1}{\ell - 1} \sum_{q\in P\setminus\{p\}} \sum_{i,j:\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j) \\
&\geqslant \frac{1}{\ell - 1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(F_i^p, F_j^q) \\
&= u_p^*(F^p, \{F^q\}_{q\neq p}),
\end{aligned}$$

where the inequality follows because $S_{p,q}(i,j) \in \{0,1\}$.

For an uninformed strategy $F^p = r$ we have

$$\begin{aligned}
u_p^*(r, \{F^q\}_{q\neq p}) &= \frac{1}{\ell - 1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(r, F_j^q) \\
&= \frac{1}{\ell - 1} \sum_{q\in P\setminus\{p\}} \sum_{j} S_{p,q}(r, F_j^q) \left(\sum_{i} \Delta_{p,q}(i,j)\right) = 0
\end{aligned}$$

The last equality follows since the row / column sum of delta matrices is zero. On the other hand, $u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})$, being a sum of only positive entries, is strictly greater than 0. □

We now use this result to prove one of our main theorems, which is the desired informed truthfulness guarantee about mechanism CAHU.

THEOREM 3.3. *With $(\varepsilon_1, \varepsilon_2)$-accurate clustering and learning, mechanism CAHU is $(\varepsilon_1+\varepsilon_2)$-informed truthful if $\min_p u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \varepsilon_1$. In particular,*

1. *For every profile $\{F^q\}_{q\in P}$ and agent $p \in P$, we have $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p(F^p, \{F^q\}_{q\neq p}) - \varepsilon_1 - \varepsilon_2$.*
2. *For any uninformed strategy F_0^p, $u_p(F_0^p, \{F^q\}_{q\neq p}) < u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})$.*

PROOF. Fix a strategy profile $\{F^q\}_{q\in P}$. We first show that $u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p(F^p, \{F^q\}_{q\neq p})$, and then show that $|u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})| \leqslant \varepsilon_1 + \varepsilon_2$. These together imply that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p(F^p, \{F^q\}_{q\neq p}) - \varepsilon_1 - \varepsilon_2$. To prove the former, note that

$$\begin{aligned}
u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) &= \frac{1}{\ell - 1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(F_i^p, F_j^q) \\
&= \frac{1}{\ell - 1} \sum_{q\in P\setminus\{p\}} \sum_{i,j:\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j) \\
&\geqslant \frac{1}{\ell - 1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{G(p),G(q)}(F_i^p, F_j^q) \\
&= u_p(F^p, \{F^q\}_{q\neq p}),
\end{aligned}$$

where the inequality again follows because $S_{G(p),G(q)}(i,j) \in \{0,1\}$. For the latter, we have

$$
\begin{aligned}
|u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})| &= \left| \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j)\big(\mathrm{Sign}(\Delta_{p,q})_{i,j} - \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})_{i,j}\big) \right| \qquad (5)\\
&\leqslant \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} |\Delta_{p,q}(i,j)\big(\mathrm{Sign}(\Delta_{p,q})_{i,j} - \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})_{i,j}\big)| \\
&\leqslant \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} |\Delta_{p,q}(i,j) - \overline{\Delta}_{G(p),G(q)}(i,j)| \\
&= \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \|\Delta_{p,q} - \overline{\Delta}_{G(p),G(q)}\|_1 \\
&\leqslant \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 + \|\Delta_{G(p),G(q)} - \overline{\Delta}_{G(p),G(q)}\|_1 \\
&\leqslant \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \varepsilon_1 + \varepsilon_2 = \varepsilon_1 + \varepsilon_2.
\end{aligned}
$$

To show that the third transition holds, we show that $|a \cdot (\mathrm{Sign}(a) - \mathrm{Sign}(b))| \leqslant |a-b|$ for all real numbers $a, b \in \mathbb{R}$. When $\mathrm{Sign}(a) = \mathrm{Sign}(b)$, this holds trivially. When $\mathrm{Sign}(a) \neq \mathrm{Sign}(b)$, note that the RHS becomes $|a| + |b|$, which is an upper bound on the LHS, which becomes $|a|$. The penultimate transition holds by ε_1-accurate clustering and ε_2-accurate estimates of cluster Delta matrices. This proves the first part of the theorem.

Now, we prove the second part of the theorem. Consider an uninformed strategy $F^p = r$ of agent p. Then,

$$
\begin{aligned}
u_p(r, \{F^q\}_{q\neq p}) &= \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{G(p),G(q)}(r, F_j^q) \\
&= \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{j} S_{G(p),G(q)}(r, F_j^q) \left(\sum_i \Delta_{p,q}(i,j) \right) = 0,
\end{aligned}
$$

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

$$
u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - \varepsilon_1 - \varepsilon_2 \geqslant \tau - \varepsilon_2
$$

for some $\tau > 0$ as $u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \varepsilon_1$. Since the delta matrices can be learned within any required accuracy, ε_2 can be made smaller than τ to ensure that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > 0$. □

The CAHU mechanism always ensures that there is no strategy profile which gives an expected utility more than $\varepsilon_1 + \varepsilon_2$ above truthful reporting. The condition $\min_p u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \varepsilon_1$ is required to ensure that any uninformed strategy gives strictly less than the truth-telling equilibrium. This is important to promote effort in collecting and reporting an informative signal. Writing it out, this condition requires that for each agent p the following holds :

$$
\frac{1}{\ell-1} \sum_{q\neq p} \sum_{i,j:\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j) > \varepsilon_1. \qquad (6)
$$

In particular, a sufficient condition for this property is that for every pair of agents the expected reward on a bonus task in the CA mechanism when making truthful reports is at least ε_1, i.e. for every pair of agents p and q,

$$\sum_{i,j:\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j) > \varepsilon_1. \tag{7}$$

In turn, as pointed out by Shnayder et al. [2016], the LHS in (7) quantity can be interpreted as a measure of how much positive correlation there is in the joint distribution on signals between a pair of agents. Note that it is not important that this is same-signal correlation. For example, this quantity would be large between an accurate and an always-wrong agent in a binary-signal domain, since the positive correlation would be between one agent's report and the flipped report from the other agent.

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

Theorem 3.4. *Let $\widehat{S}$ be an arbitrary scoring function i.e. $\widehat{S}_{G_s,G_t}(i,j)$ is the reward when a user p from cluster G_s and user q from cluster G_t report signals i and j respectively. Then for every profile $\{F^q\}_{q\in P}$ and agent $p \in P$, we have*

(1) $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant \hat{u}_p(F^p, \{F^q\}_{q\neq p}) - \varepsilon_1 - \varepsilon_2$.
(2) *If* $\min_p u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \varepsilon$, *then for any uninformed strategy* F^p_0, $\hat{u}_p(F^p_0, \{F^q\}_{q\neq p}) < u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})$.

Proof.

$$\begin{aligned} u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) &= \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(i,j) \\ &= \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j:\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j) \\ &\geqslant \frac{1}{\ell-1} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \cdot \widehat{S}_{p,q}(F^p_i, F^q_j) \\ &= \hat{u}_p(F^p, \{F^q\}_{q\neq p}), \end{aligned}$$

Now the proof of theorem 3.3 shows that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - \varepsilon_1 - \varepsilon_2$. Using the result above we get $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant \hat{u}_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - \varepsilon_1 - \varepsilon_2$. It is also easy to see that $\hat{u}_p(F^p_0, \{F^q\}_{q\neq p}) = 0$ for any uninformed strategy F^p_0 and the proof of theorem 3.3 also shows that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})$ can be made positive whenever $\min_p u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \varepsilon$. □

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

4 LEARNING THE AGENT SIGNAL TYPES

In this section, we provide algorithms for learning a clustering of agent signal types from reports, and further, for learning the cluster pairwise Δ matrices. The estimates of the Δ matrices can then be used to give an approximate-informed truthful mechanism. Along the way, we couple our methods with the latent "confusion matrix" methods of Dawid and Skene [1979b].

Recall that m is the total number of tasks about which reports are collected. Reports on m_1 of these tasks will also be used for clustering, and reports on a further m_2 of these tasks will be used for learning the cluster pairwise Δ matrices. We consider two different schemes for assigning agents to tasks for the purpose of clustering and learning (see Figures 1 and 2):

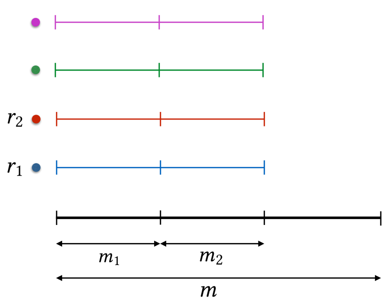

Fig. 1. Fixed Task Assignment

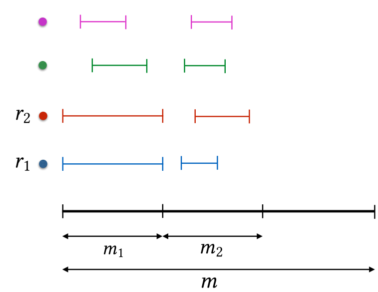

Fig. 2. Uniform Task Assignment

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

For each assignment scheme, the analysis establishes that there are enough agents who have done a sufficient number of joint tasks. Table 1 summarizes the sample complexity results, stating them under two different assumptions about the way in which signals are generated.

4.1 Clustering

We proceed by presenting and analyzing a simple clustering algorithm.

Definition 4.1. A clustering $G_1, \ldots, G_K$ is ε-good if for some $\gamma > 0$

$$G(q) = G(r) \Rightarrow \|\Delta_{pq} - \Delta_{pr}\|_1 \leqslant \varepsilon - 4\gamma \ \forall p \in [\ell] \setminus \{q, r\} \tag{8}$$

$$G(q) \neq G(r) \Rightarrow \|\Delta_{pq} - \Delta_{pr}\|_1 > \varepsilon \ \forall p \in [\ell] \setminus \{q, r\} \tag{9}$$

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

‡ In the no assumption approach (resp. Dawid-Skene Model), ε' is the error in the estimation of the joint probability distribution (resp. aggregate confusion matrix).

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

Table 1. Sample complexity for the CAHU mechanism. The rows indicate the assignment scheme and the columns indicate the modeling assumption. Here ℓ is the number of agents, n is the number of signals, ε' is a parameter that controls learning accuracy ‡ , γ is a clustering parameter, K is the number of clusters, and m_1 (resp. m_2) is the size of the set of tasks from which the tasks used for clustering (resp. learning) are sampled.

We first show that an ε-good clustering, if exists, must be unique.

Theorem 4.2. *Suppose there exist two clustering $\{G_j\}_{j\in[K]}$ and $\{T_i\}_{i\in[K']}$ that are ε-good. Then $K'=K$ and $G_j=T_{\pi(j)}$ for some permutation π over $[K]$.*

Proof. Suppose equations 8 and 9 hold with parameters γ_1 and γ_2 respectively for the clusterings $\{G_j\}_{j\in[K]}$ and $\{T_i\}_{i\in[K']}$. If possible, assume there exist T_i and G_j such that $T_i\setminus G_j\neq\emptyset$, $G_j\setminus T_i\neq\emptyset$ and $T_i\cap G_j\neq\emptyset$. Pick $s\in T_i\cap G_j$ and $r\in G_j\setminus T_i$. Then we must have, for any $p\notin\{q,s,r\}$,

(1) $\|\Delta_{pr}-\Delta_{ps}\|_1>\varepsilon$ (inter-cluster distance in $\{T_i\}_{i\in[K']}$)
(2) $\|\Delta_{pr}-\Delta_{ps}\|_1\leqslant\varepsilon-4\gamma_1$ (intra-cluster distance in $\{G_j\}_{j\in[K]}$)

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

(1) $\|\Delta_{pq}-\Delta_{pr}\|_1>\varepsilon$ (inter-cluster distance in $\{T_i\}_{i\in[K']}$)
(2) $\|\Delta_{pq}-\Delta_{pr}\|_1\leqslant\varepsilon-4\gamma_1$ (intra-cluster distance in $\{G_j\}_{j\in[K]}$)

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

Since there is a unique ε-good clustering (up to a permutation), we will refer to this clustering as the *correct clustering*. The assumption that there exists an ε-good clustering is stronger than Equation (3) introduced earlier. In particular, identifying the correct clustering needs to satisfy eq. (9), i.e. the Δ-matrices of two agents belonging to two different clusters are different with respect to every other agent. So, we need low inter-cluster similarities in addition to high intra-cluster similarities. The pseudo-code for the clustering algorithm is presented in Algorithm 2. This algorithm iterates over the agents, and forms clusters in a greedy manner. First, we prove that as long as we can find an agent p_t that has $\Omega\left(\frac{n^2\log(\ell/\delta)}{\gamma^2}\right)$ tasks in common with both q_t and i, then the clustering produced by Algorithm 2 is correct with probability at least $1-\delta$.

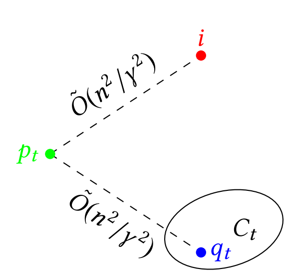

Fig. 3. Algorithm 2 checks whether i and q_t are in the same cluster by estimating Δ_{p_t,q_t} and $\Delta_{p_t,i}$.

Algorithm 2 Clustering

Input: ε, γ such that there exists an ε-good clustering with parameter γ.
Output: A clustering $\{\hat{G}_t\}_{t=1}^{\hat{K}}$

$\hat{G} \leftarrow \emptyset$, $\hat{K} \leftarrow 0$ ▷ $\hat{G}$ is the list of clusters, $\hat{K} = |\hat{G}|$
Make a new cluster $\hat{G}_1$ and add agent 1
Add $\hat{G}_1$ to $\hat{G}$, $\hat{K} \leftarrow \hat{K} + 1$
for $i = 2, \ldots, \ell$ **do**
  for $t \in [\hat{K}]$ **do**
    Pick an arbitrary agent $q_t \in \hat{G}_t$
    Pick $p_t \in [l] \setminus \{i, q_t\}$ (**Fixed**) or $p_t \in \{r_1, r_2\} \setminus \{i, q_t\}$ (**Uniform**), such that p_t has at least $\Omega(\frac{n^2 \log(K\ell/\delta)}{\gamma^2})$ tasks in common with both q_t and i
    Let $\bar{\Delta}_{p_t,q_t}$ be the empirical Delta matrix from reports of agents p_t and q_t
    Let $\bar{\Delta}_{p_t,i}$ be the empirical Delta matrix from reports of agents p_t and i
  end for
  if $\exists t \in [\hat{K}] : \|\bar{\Delta}_{p_t,q_t} - \bar{\Delta}_{p_t,i}\|_1 \leqslant \varepsilon - 2\gamma$ **then**
    add i to $\hat{G}_t$ (with ties broken arbitrarily for t)
  else
    Make a new cluster $\hat{G}_{\hat{K}+1}$ and add agent i to it
    Add $\hat{G}_{\hat{K}+1}$ to $\hat{G}$, $\hat{K} \leftarrow \hat{K} + 1$
  end if
end for

Theorem 4.3. *If for all $i \in P$ and $q_t \in G(i)$, there exists p_t which has $\Omega\left(\frac{n^2 \log(\ell/\delta)}{\gamma^2}\right)$ tasks in common with both q_t and i, then Algorithm 2 recovers the correct clustering i.e. $\hat{G}_t = G_t$ for $t = 1, \ldots, K$ with probability at least $1 - \delta$.*

We need two key technical lemmas to prove theorem 4.3. The first lemma shows that in order to estimate $\Delta_{p,q}$ with an L1 distance of at most γ, it is sufficient to estimate the joint probability distribution $D_{p,q}$ with an L1 distance of at most $\gamma/3$. With this, we can estimate the delta matrices of agent pairs from the joint empirical distributions of their reports.

Lemma 4.4. *For all $p, q \in P$, $\|\bar{D}_{p,q} - D_{p,q}\|_1 \leqslant \gamma/3 \Rightarrow \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 \leqslant \gamma$.*

Proof.

$$
\begin{aligned}
\|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 &= \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - \bar{D}_p(i)\bar{D}_q(j) - \left(D_{p,q}(i,j) - D_p(i)D_q(j)\right)\right| \\
&= \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - D_{p,q}(i,j)\right| + \sum_{i,j} \left|\bar{D}_p(i)\bar{D}_q(j) - \bar{D}_p(i)D_q(j) + \bar{D}_p(i)D_q(j) - D_p(i)D_q(j)\right| \\
&\leqslant \gamma/3 + \sum_i \bar{D}_p(i) \sum_j \left|\bar{D}_q(j) - D_q(j)\right| + \sum_j D_q(j) \sum_i \left|\bar{D}_p(i) - D_p(i)\right| \\
&\leqslant \gamma/3 + \sum_j \left|\bar{D}_q(j) - D_q(j)\right| + \sum_i \left|\bar{D}_p(i) - D_p(i)\right| \\
&\leqslant \gamma/3 + \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - D_{p,q}(i,j)\right| + \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - D_{p,q}(i,j)\right| \\
&\leqslant \gamma,
\end{aligned}
$$

as required. □

The second lemma is about learning the empirical distributions of reports of pairs of agents. This can be proved using Theorems 3.1 and 2.2 from the work of Devroye and Lugosi [2012].

Lemma 4.5. *Any distribution over a finite domain Ω is learnable within a L1 distance of d with probability at least $1-\delta$, by observing $O\left(\frac{|\Omega|}{d^2}\log(1/\delta)\right)$ samples from the distribution.*

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

Corollary 4.6. *For any agent pair $p, q \in P$, the joint distribution of their reports $D_{p,q}$ is learnable within a L1 distance of γ using $O\left(\frac{n^2}{\gamma^2}\log(K\ell/\delta)\right)$ reports on joint tasks with probability at least $1-\delta/K\ell$.*

We are now ready to prove Theorem 4.3.

Proof. (of Theorem 4.3) The proof is by induction on the number of agents ℓ. Suppose all the agents up to and including $i-1$ have been clustered correctly. Consider the i-th agent and suppose i belongs to the cluster G_t. Suppose $\hat{G}_t \neq \emptyset$. Then using the triangle inequality we have

$$\|\bar{\Delta}_{p_t,q_t} - \bar{\Delta}_{p_t,i}\|_1 \leqslant \|\bar{\Delta}_{p_t,q_t} - \Delta_{p_t,q_t}\|_1 + \|\Delta_{p_t,q_t} - \Delta_{p_t,i}\|_1 + \|\bar{\Delta}_{p_t,i} - \Delta_{p_t,i}\|_1$$

Since $q_t \in G_t$, we have $\|\Delta_{p_t,q_t} - \Delta_{p_t,i}\|_1 \leqslant \varepsilon/2 - 4\gamma$. Moreover, using lemma 4.4 and corollary 4.6 we have that, with probability at least $1-\delta/K\ell$, $\|\bar{\Delta}_{p_t,q_t} - \Delta_{p_t,q_t}\|_1 \leqslant \gamma$ and $\|\bar{\Delta}_{p_t,i} - \Delta_{p_t,i}\| \leqslant \gamma$. This ensures that $\|\bar{\Delta}_{p_t,q_t} - \bar{\Delta}_{p_t,i}\|_1 \leqslant \varepsilon/2 - 2\gamma$. On the other hand pick any cluster G_s such that $s \neq t$ and $\hat{G}_s \neq \emptyset$. Then

$$\|\bar{\Delta}_{p_s,q_s} - \bar{\Delta}_{p_s,i}\|_1 \geqslant \|\Delta_{p_s,q_s} - \Delta_{p_s,i}\| - \|\bar{\Delta}_{p_s,q_s} - \Delta_{p_s,q_s}\|_1 - \|\bar{\Delta}_{p_s,i} - \Delta_{p_s,i}\|_1$$

Since $i \notin G_s$ we have $\|\Delta_{p_s,q_s} - \Delta_{p_s,i}\|_1 > \varepsilon/2$. Again, with probability at least $1-\delta/K\ell$, we have $\|\bar{\Delta}_{p_s,q_s} - \Delta_{p_s,q_s}\|_1 \leqslant \gamma$ and $\|\bar{\Delta}_{p_s,i} - \Delta_{p_s,i}\|_1 \leqslant \gamma$. This ensures that $\|\bar{\Delta}_{p_s,q_s} - \bar{\Delta}_{p_s,i}\|_1 > \varepsilon/2 - 2\gamma$. This ensures that condition on line (11) is violated for all clusters $s \neq t$. If $\hat{G}_t \neq \emptyset$ this condition is satisfied and agent i added to cluster $\hat{G}_t$, otherwise the algorithm makes a new cluster with agent i. Now note that the algorithm makes a new cluster only when it sees an agent belonging to a new cluster. This implies that $\hat{K} = K$. Taking a union bound over the K choices of q_s for the K clusters, we see that agent i is assigned to its correct cluster with probability at least $1-\delta/\ell$. Finally, taking a union bound over all the ℓ agents we get the desired result. □

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

In the uniform assignment scheme, we select two agents r_1 and r_2 uniformly at random to be reference agents, and assign these agents to each of $m_1 = \Omega(\frac{n^2}{\gamma^2}\log(K\ell/\delta))$ tasks. For all other agents we then assign $s_1 = \Omega(\frac{n^2}{\gamma^2}\log(K\ell/\delta))$ tasks uniformly at random from this set of m_1 tasks. If $m_1 = s_1$, then the uniform task assignment is the same as fixed task assignment. However, in applications (e.g., [Karger et al., 2011]), where one wants the task assignments to be more uniform across tasks, it will make sense to use a larger value of m_1. The reference agent r_1 can act as p_t

for all agent pairs q_t and i other than r_1. Similarly, reference r_2 can act as p_t for all agent pairs q_t and i other than r_2. If $q_t = r_1$ and $i = r_2$ or $q_t = r_2$ and $i = r_1$, then any other agent can act as p_t. The total number of tasks performed is $\Omega(\frac{\ell n^2}{\gamma^2} \log(K\ell/\delta) + m_1)$, which is sufficient for the high probability result.

4.2 Learning the Cluster Pairwise Δ Matrices

We proceed now under the assumption that the agents are clustered into K groups, $G_1, \ldots, G_K$. Our goal is to estimate the cluster-pairwise delta matrices Δ_{G_s,G_t} as required by Algorithm 1. We estimate the Δ_{G_s,G_t} under two different settings: when we have no model of the signal distribution, and in the Dawid-Skene latent attribute model.

Algorithm 3 Learning-Δ-No-Assumption

for $t = 1, \ldots, K$ **do**
  Chose agent $q_t \in G_t$ arbitrarily.
end for
for each pair of clusters G_s, G_t **do**
  Let q_s and q_t be the chosen agents for G_s and G_t, respectively.
  Let $\bar{D}_{q_s,q_t}$ be the empirical estimate of D_{q_s,q_t} such that $\|\bar{D}_{q_s,q_t} - D_{q_s,q_t}\|_1 \leqslant \varepsilon'$ with probability at least $1 - \delta/K^2$
  Let $\bar{\Delta}_{q_s,q_t}$ be the empirical Delta matrix computed using $\bar{D}_{q_s,q_t}$
  Set $\bar{\Delta}_{G_s,G_t} = \bar{\Delta}_{q_s,q_t}$
end for

4.2.1 ***Learning the Δ-Matrices with No Assumption.*** We first characterize the sample complexity of learning the Δ-matrices in the absence of any modeling assumptions. In order to estimate $\bar{\Delta}_{G_s,G_t}$, Algorithm 3 first picks agent q_s from cluster G_s, estimates $\bar{\Delta}_{q_s,q_t}$ and use this estimate in place of $\bar{\Delta}_{G_s,G_t}$. For the fixed assignment scheme, we assign the agents q_s to the same set of tasks of size $O\left(\frac{n^2}{(\varepsilon')^2}\log(K/\delta)\right)$. For the uniform assignment scheme, we assign the agents to subsets of tasks of an appropriate size among the pool of m_2 tasks.

Theorem 4.7. *Given an ε-good clustering $\{G_s\}_{s=1}^K$, if the number of shared tasks between any pair of agents q_s, q_t is $O\left(\frac{n^2}{(\varepsilon')^2}\log(K/\delta)\right)$, then algorithm 3 guarantees that for all s, t, $\|\bar{\Delta}_{G_s,G_t} - \Delta_{G_s,G_t}\|_1 \leqslant 3\varepsilon' + 2\varepsilon$ with probability at least $1-\delta$. The total number of samples collected by the algorithm is $O\left(\frac{Kn^2}{(\varepsilon')^2}\log(K/\delta)\right)$ $\left(\text{resp. } O\left(Km_2^{7/8}\sqrt{\frac{n^2}{(\varepsilon')^2}\log(K/\delta)}\right) \text{ w.h.p.}\right)$ under the fixed (resp. uniform) assignment scheme.*

We first prove a sequence of lemmas that will be used to prove the result.

Lemma 4.8. *For every pair of agents p, q, we have*

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

Proof. Let $\Delta_{p,G(q)} = \frac{1}{|G(q)|}\sum_{r\in G(q)} \Delta_{p,r}$, then using the property of clusters we have

$$\begin{aligned}
\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 &= \left\|\Delta_{p,q} - \frac{1}{|G(p)|\,|G(q)|}\sum\nolimits_{u\in G(p),v\in G(q)} \Delta_{u,v}\right\|_1 \\
&= \left\|\frac{1}{|G(p)|\,|G(q)|}\sum\nolimits_{u\in G(p),v\in G(q)} \left(\Delta_{p,q} - \Delta_{u,v}\right)\right\|_1 \\
&\leqslant \frac{1}{|G(p)|\,|G(q)|}\sum_{u\in G(p),v\in G(q)} \|\Delta_{p,q} - \Delta_{u,v}\|_1 \\
&\leqslant \frac{1}{|G(p)|\,|G(q)|}\sum_{u\in G(p),v\in G(q)} \|\Delta_{p,q} - \Delta_{u,q}\|_1 + \|\Delta_{u,q} - \Delta_{u,v}\|_1 \\
&\leqslant \frac{1}{|G(p)|\,|G(q)|}\sum_{u\in G(p),v\in G(q)} 2 \max_{a,b,c\in P: G(a)=G(b)} \|\Delta_{a,c} - \Delta_{b,c}\|_1 \\
&= 2 \max_{a,b,c\in P: G(a)=G(b)} \|\Delta_{a,c} - \Delta_{b,c}\|_1,
\end{aligned}$$

as required. □

The next lemma characterizes the error made by Algorithm 3 in estimating the Δ_{G_s,G_t}-matrices.

Lemma 4.9. *For any two agents $p \in G_s$ and $q \in G_t$, $\|\bar{D}_{p,q} - D_{p,q}\|_1 \leqslant \varepsilon' \Rightarrow \|\bar{\Delta}_{p,q} - \Delta_{G_s,G_t}\|_1 \leqslant 3\varepsilon' + 2\varepsilon$.*

Proof. Lemma 4.4 shows that $\|\bar{D}_{p,q} - D_{p,q}\|_1 \leqslant \varepsilon' \Rightarrow \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 \leqslant 3\varepsilon'$.
Now,

$$\|\bar{\Delta}_{p,q} - \Delta_{G_s,G_t}\|_1 \leqslant \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 + \|\Delta_{p,q} - \Delta_{G_s,G_t}\|_1 \leqslant 3\varepsilon' + 2\varepsilon.$$

The last inequality uses Lemma 4.8 □

Proof. (Theorem 4.7) By Lemma 4.5, to estimate $D_{p,q}$ within a distance of ε' with probability at least $1-\delta/K^2$, we need $O\left(\frac{n^2}{(\varepsilon')^2}\log(K^2/\delta)\right)$. By a union bound over the K^2 pairs of clusters we see that with probability at least $1-\delta$, we have $\|\bar{D}_{q_s,q_t} - D_{q_s,q_t}\|_1 \leqslant \varepsilon'$. This proves the first part of the theorem. When the assignment scheme is fixed, we can assign all the same tasks to K agents $\{q_t\}_{t=1}^K$, and hence the total number of samples is multiplied by K.

On the other hand, under the uniform assignment scheme, suppose each agent $\{q_t\}_{t=1}^K$ is assigned to a subset of s_2 tasks selected uniformly at random from the pool of m_2 tasks. Now consider any two agents q_s and q_t. Let X_i be an indicator random variable which is 1 when $i \in [m_2]$ is included in tasks of q_s, and 0 otherwise. Also, let Y_i be a similar random variable for the tasks of q_t. Let $Z_i = X_i \times Y_i$. The probability that both agents are assigned to a particular task i, $\Pr(Z_i = 1) = (s_2/m_2)^2$. Therefore, the expected number of overlapping tasks among the two agents is $m_2 \cdot \left(\frac{s_2}{m_2}\right)^2 = \frac{s_2^2}{m_2}$, i.e. $\mathrm{E}\left[\sum_i Z_i\right] = \frac{s_2^2}{m_2}$. Now, we want to bound the deviations from this expectations. Let $R_j = \mathrm{E}\left[\sum_{i=1}^{m_2} Z_i | X_1,\cdots,X_j,Y_1,\cdots,Y_j\right]$, then R_j is a Doob martingale sequence for $\sum_{i=1}^{j} Z_i$. Also, it is easy to see that this martingale sequence is bounded by 1, i.e. $|R_{j+1} - R_j| \leqslant 1$. Therefore, we apply the Azuma-Hoeffding bound (Lemma 4.10)as

$$\Pr\left[\left|\sum_i Z_i\right| > \frac{s_2^2}{2m_2}\right] \leqslant 2\exp\left\{-\frac{s_2^4}{8m_2^3}\right\}.$$

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

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

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

Lemma 4.10. *Suppose $X_n, n \geqslant 1$ is a martingale such that $X_0 = 0$ and $|X_i - X_{i-1}| \leqslant 1$ for each $1 \leqslant i \leqslant n$. Then for every $t > 0$*

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

4.2.2 ***Learning the Δ-matrices Under the Dawid-Skene Model.*** In this section, we assume that the agents receive signals according to the Dawid and Skene [1979a] model. Here, each task has a latent attribute and each agent has a confusion matrix to parameterize its signal distribution conditioned on this latent value. More formally:

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

$$D_{p,q}(S_p = i, S_q = j) = \sum_{k=1}^n \pi_k C^p_{ki} C^q_{kj}, \tag{10}$$

and the marginal signal distribution for agent p is

$$D_p(S_p = i) = \sum_{k=1}^n \pi_k C^p_{ki}. \tag{11}$$

For cluster G_t, we write $C^t = \frac{1}{|G_t|}\sum_{p\in G_t} C^p$ to denote the aggregate confusion matrix of G_t. As before, we assume that we are given an ε-good clustering, $G_1, \ldots, G_K$, of the agents. Our goal is to provide an estimate of the Δ_{G_s,G_t}-matrices.

Lemma 4.11 proves that in order to estimate Δ_{G_s,G_t} within an L1 distance of ε', it is enough to estimate the aggregate confusion matrices within an L1 distance of $\varepsilon'/4$. So in order to learn the pairwise delta matrices between clusters, we first ensure that for each cluster G_t, we have $\|\bar{C}^t - C^t\|_1 \leqslant \varepsilon'/4$ with probability at least $1 - \delta/K$, and then use the following formula to compute the delta matrices:

$$\Delta_{G_s,G_t}(i,j) = \sum_{k=1}^n \pi_k \bar{C}^s_{ki}\bar{C}^t_{kj} - \sum_{k=1}^n \pi_k \bar{C}^s_{ki} \sum_{k=1}^n \pi_k \bar{C}^t_{kj} \tag{12}$$

Lemma 4.11. *Forall G_a, G_b, $\|\bar{C}^a - C^a\|_1 \leqslant \varepsilon'/4$ and $\|\bar{C}^b - C^b\|_1 \leqslant \varepsilon'/4 \Rightarrow \|\bar{\Delta}_{G_a,G_b} - \Delta_{G_a,G_b}\| \leqslant \varepsilon'$.*

Proof.

$$\Delta_{G_a,G_b}(i,j) = \frac{1}{|G_a|\,|G_b|}\sum_{p\in G_a, q\in G_b}\Delta_{p,q}(i,j) = \frac{1}{|G_a|\,|G_b|}\sum_{p\in G_a, q\in G_b} D_{p,q}(i,j) - D_p(i)D_q(j)$$
$$= \frac{1}{|G_a|\,|G_b|}\sum_{p\in G_a, q\in G_b}\sum_k \pi_k C^p_{ki}C^q_{kj} - \sum_k \pi_k C^p_{ki}\sum_k C^q_{kj}$$
$$= \sum_k \pi_k\left(\frac{1}{|G_a|}\sum_{p\in G_a} C^p_{ki}\right)\left(\frac{1}{|G_b|}\sum_{q\in G_b} C^q_{kj}\right)$$
$$- \sum_k \pi_k\left(\frac{1}{|G_a|}\sum_{p\in G_a} C^p_{ki}\right)\sum_k \pi_k\left(\frac{1}{|G_b|}\sum_{q\in G_b} C^q_{kj}\right)$$
$$= \sum_k \pi_k C^a_{ki}C^b_{kj} - \sum_k \pi_k C^a_{ki}\sum_k \pi_k C^b_{kj}$$

Now

$$\|\bar{\Delta}_{G_a,G_b} - \Delta_{G_a,G_b}\|_1 = \sum_{i,j}\left|\bar{\Delta}_{G_a,G_b}(i,j) - \Delta_{G_a,G_b}(i,j)\right|$$
$$= \sum_{i,j}\left|\sum_k \pi_k \bar{C}^a_{ki}\bar{C}^b_{kj} - \sum_k \pi_k \bar{C}^a_{ki}\sum_k \pi_k \bar{C}^b_{kj} - \left(\sum_k \pi_k C^a_{ki}C^b_{kj} - \sum_k \pi_k C^a_{ki}\sum_k \pi_k C^b_{kj}\right)\right|$$
$$\leqslant \sum_{i,j}\left|\sum_k \pi_k \bar{C}^a_{ki}\bar{C}^b_{kj} - \sum_k \pi_k C^a_{ki}C^b_{kj}\right| + \sum_{i,j}\left|\sum_k \pi_k \bar{C}^a_{ki}\sum_k \pi_k \bar{C}^b_{kj} - \sum_k \pi_k C^a_{ki}\sum_k \pi_k C^b_{kj}\right|$$
$$= \sum_{i,j}\left|\sum_k \pi_k \bar{C}^a_{ki}\bar{C}^b_{kj} - \sum_k \pi_k \bar{C}^a_{ki}C^b_{kj} + \sum_k \pi_k \bar{C}^a_{ki}C^b_{kj} - \sum_k \pi_k C^a_{ki}C^b_{kj}\right|$$
$$+ \sum_{i,j}\left|\sum_k \pi_k \bar{C}^a_{ki}\sum_k \pi_k \bar{C}^b_{kj} - \sum_k \pi_k \bar{C}^a_{ki}\sum_k \pi_k C^b_{kj} + \sum_k \pi_k \bar{C}^a_{ki}\sum_k \pi_k C^b_{kj} - \sum_k \pi_k C^a_{ki}\sum_k \pi_k C^b_{kj}\right|$$
$$= \sum_k \pi_k \sum_j \left|\bar{C}^b_{kj} - C^b_{kj}\right|\sum_i \bar{C}^a_{ki} + \sum_k \pi_k \sum_i \left|\bar{C}^a_{ki} - C^a_{ki}\right|\sum_j C^b_{kj}$$
$$+ \sum_k \pi_k \sum_i \bar{C}^a_{ki}\sum_{k'}\pi_{k'}\sum_j\left|\bar{C}^b_{k'j} - C^b_{k'j}\right| + \sum_k \pi_k \sum_j C^b_{kj}\sum_{k'}\pi_{k'}\sum_i\left|\bar{C}^a_{k'i} - C^a_{k'i}\right|$$
$$= 2\sum_k \pi_k \sum_j\left|\bar{C}^b_{kj} - C^b_{kj}\right| + 2\sum_k \pi_k \sum_i\left|\bar{C}^a_{ki} - C^b_{ki}\right|$$
$$\leqslant 2\|\bar{C}^a - C^a\|_1 + 2\|\bar{C}^b - C^b\|_1 \leqslant 4 \times \varepsilon'/4 = \varepsilon'$$

□

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

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

(2) Pr [task j is unlabeled by G_a] $= \left(1 - \frac{\log(m_2 K/\beta)}{|G_a|}\right)^{|G_a|} \leqslant \frac{\beta}{m_2 K}$. Taking a union bound over the m_2 tasks and K clusters, we get the probability that any task is unlabeled is at most β. Now if we choose $\beta = 1/\text{poly}(m_2)$, we observe that with probability at least $1 - 1/\text{poly}(m_2)$, each task j is labeled by some agent in each cluster when $B_a = \tilde{O}(\frac{m_2}{|G_a|})$.

All that is left to do is to provide an algorithm and sample complexity for learning the aggregate confusion matrices. For this, we will use n dimensional unit vectors to denote the reports of the agents (recall that there are n possible signals). In particular agent p's report on task j, $r_{pj} \in \{0,1\}^n$. If p's report on task j is c, then the c-th coordinate of r_{pj} is 1 and all the other coordinates are 0. The expected value of agent p's report on the jth task is $\mathrm{E}\left[r_{pj}\right] = \sum_{k=1}^n \pi_k C_k^p$ The aggregated report for a cluster G_t is given as $R_{tj} = \frac{1}{|G_t|} \sum_{p \in G_t} r_{pj}$.

Suppose we want to estimate the aggregate confusion matrix C^1 of some cluster G_1. To do so, we first pick three clusters G_1, G_2 and G_3 and write down the corresponding cross moments. Let (a, b, c) be a permutation of the set $\{1, 2, 3\}$. We have:

$$M_1 = E[R_{aj}] = \sum_k \pi_k C_k^a \tag{13}$$

$$M_2 = E[R_{aj} \otimes R_{bj}] = \sum_k \pi_k C_k^a \otimes C_k^b \tag{14}$$

$$M_3 = E[R_{aj} \otimes R_{bj} \otimes R_{cj}] = \sum_k \pi_k C_k^a \otimes C_k^b \otimes C_k^c \tag{15}$$

The cross moments are asymmetric, however using Theorem 3.6 in the work by Anandkumar et al. [2014], we can write the cross-moments in a symmetric form.

Lemma 4.12. *Assume that the vectors $\{C_1^t, \dots, C_n^t\}$ are linearly independent for each $t \in \{1, 2, 3\}$. For any permutation (a, b, c) of the set $\{1, 2, 3\}$ define*

$$R'_{aj} = \mathrm{E}\left[R_{cj} \otimes R_{bj}\right] \left(\mathrm{E}\left[R_{aj} \otimes R_{bj}\right]\right)^{-1} R_{aj}$$
$$R'_{bj} = \mathrm{E}\left[R_{cj} \otimes R_{aj}\right] \left(\mathrm{E}\left[R_{bj} \otimes R_{aj}\right]\right)^{-1} R_{bj}$$
$$M_2 = \mathrm{E}\left[R'_{aj} \otimes R'_{bj}\right] \text{ and } M_3 = \mathrm{E}\left[R'_{aj} \otimes R'_{bj} \otimes R_{cj}\right]$$

$$\text{Then } M_2 = \sum_{k=1}^n \pi_k C_k^c \otimes C_k^c \text{ and } M_3 = \sum_{k=1}^n \pi_k C_k^c \otimes C_k^c \otimes C_k^c$$

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

Lemma 4.13. (1) *For any group G_a, $\mathrm{E}\left[\tilde{R}_{aj}\right] = \mathrm{E}\left[R_{aj}\right]$*
(2) *For any pair of groups G_a and G_b, $\mathrm{E}\left[\tilde{R}_{aj} \otimes \tilde{R}_{bj}\right] = \mathrm{E}\left[R_{aj} \otimes R_{bj}\right]$*
(3) *For any three groups G_a, G_b and G_c, $\mathrm{E}\left[\tilde{R}_{aj} \otimes \tilde{R}_{bj} \otimes \tilde{R}_{cj}\right] = \mathrm{E}\left[R_{aj} \otimes R_{bj} \otimes R_{cj}\right]$*

Proof.

1. First moments of $\{\tilde{R}_{gj}\}_{g=1}^{K}$ and $\{R_{gj}\}_{g=1}^{K}$ are equal :

$$\mathrm{E}\left[\tilde{R}_{aj}\right] = \frac{1}{|G_a|}\sum_{p\in G_a}\mathrm{E}\left[r_{pj}\right] = E[R_{aj}]$$

2. Second order cross-moments of $\{\tilde{R}_{gj}\}_{g=1}^{K}$ and $\{R_{gj}\}_{g=1}^{K}$ are equal :

$$\mathrm{E}\left[\tilde{R}_{aj}\otimes\tilde{R}_{bj}\right] = \sum_k \pi_k \mathrm{E}\left[\tilde{R}_{aj}\otimes\tilde{R}_{bj}|y_j = k\right] = \sum_k \pi_k \mathrm{E}\left[\tilde{R}_{aj}|y_j = k\right]\otimes \mathrm{E}\left[\tilde{R}_{bj}|y_j = k\right]$$

$$= \sum_k \pi_k \left(\frac{1}{|G_a|}\sum_{p\in G_a} C_k^p\right)\otimes\left(\frac{1}{|G_b|}\sum_{q\in G_b} C_k^q\right) = \sum_k \pi_k C_k^a\otimes C_k^b = \mathrm{E}\left[R_{aj}\otimes R_{bj}\right]$$

3. Third order cross-moments of $\{\tilde{R}_{gj}\}_{g=1}^{K}$ and $\{R_{gj}\}_{g=1}^{K}$ are equal :

$$\mathrm{E}\left[\tilde{R}_{aj}\otimes\tilde{R}_{bj}\otimes\tilde{R}_{cj}\right] = \sum_k \pi_k \mathrm{E}\left[\tilde{R}_{aj}\otimes\tilde{R}_{bj}\otimes\tilde{R}_{cj}|y_j = k\right]$$

$$= \sum_k \pi_k \mathrm{E}\left[\tilde{R}_{aj}|y_j = k\right]\otimes \mathrm{E}\left[\tilde{R}_{bj}|y_j = k\right]\otimes \mathrm{E}\left[\tilde{R}_{cj}|y_j = k\right]$$

$$= \sum_k \pi_k \left(\frac{1}{|G_a|}\sum_{p\in G_a} C_k^p\right)\otimes\left(\frac{1}{|G_b|}\sum_{q\in G_b} C_k^q\right)\otimes\left(\frac{1}{|G_c|}\sum_{r\in G_c} C_k^r\right)$$

$$= \sum_k \pi_k C_k^a\otimes C_k^b\otimes C_k^c = \mathrm{E}\left[R_{aj}\otimes R_{bj}\otimes R_{cj}\right]$$

□

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

$$\hat{R}'_{aj} = \left(\frac{1}{m_2}\sum_{j'=1}^{m_2}\tilde{R}_{cj'}\otimes\tilde{R}_{bj'}\right)\left(\frac{1}{m_2}\sum_{j'=1}^{m_2}\tilde{R}_{aj'}\otimes\tilde{R}_{bj'}\right)^{-1}\tilde{R}_{aj} \tag{16}$$

$$\hat{R}'_{bj} = \left(\frac{1}{m_2}\sum_{j'=1}^{m_2}\tilde{R}_{cj'}\otimes\tilde{R}_{aj'}\right)\left(\frac{1}{m_2}\sum_{j'=1}^{m_2}\tilde{R}_{bj'}\otimes\tilde{R}_{aj'}\right)^{-1}\tilde{R}_{bj} \tag{17}$$

$$\hat{M}_2 = \frac{1}{m_2}\sum_{j'=1}^{m_2}\hat{R}'_{aj'}\otimes\hat{R}'_{bj'} \text{ and } \hat{M}_3 = \frac{1}{m_2}\sum_{j'=1}^{m_2}\hat{R}'_{aj'}\otimes\hat{R}'_{bj'}\otimes\tilde{R}_{cj'} \tag{18}$$

We use the tensor decomposition algorithm (4) on $\hat{M}_2$ and $\hat{M}_3$ to recover the aggregate confusion matrix $\bar{C}^c$ and $\bar{\Pi}$, where $\bar{\Pi}$ is a diagonal matrix whose k-th component is $\bar{\pi}_k$, an estimate of π_k. In order to analyze the sample complexity of Algorithm 4, we need to make some mild assumptions about the problem instance. For any two clusters G_a and G_b, define $S_{ab} = \mathrm{E}\left[R_{aj}\otimes R_{bj}\right] = \sum_{k=1}^{n}\pi_k C_k^a \otimes C_k^b$. We make the following assumptions:

(1) There exists $\sigma_L > 0$ such that $\sigma_n(S_{ab}) \geqslant \sigma_L$ for each pair of clusters a and b, where $\sigma_n(M)$ is the n-th smallest eigenvalue of M.
(2) $\kappa = \min_{t\in[k]}\min_{s\in[n]}\min_{r\neq s}\left\{C_{rr}^t - C_{rs}^t\right\} > 0$

The first assumption implies that the matrices S_{ab} are non-singular. The second assumption implies that within a group, the probability of assigning the correct label is always higher than the

Algorithm 4 Estimating Aggregate Confusion Matrix

Input: K clusters of agents $G_1, G_2, \dots, G_K$ and the reports $\tilde{R}_{gj} \in \{0,1\}^n$ for $j \in [m]$ and $g \in [K]$
Output: Estimate of the aggregate confusion matrices $\bar{C}^g$ for all $g \in [K]$

```
Partition the K clusters into groups of three
for Each group of three clusters {g_a, g_b, g_c} do
    for (a, b, c) ∈ {(g_b, g_c, g_a), (g_c, g_a, g_b), (g_a, g_b, g_c)} do
        Compute the second and the third order moments M̂_2 ∈ R^{n×n}, M̂_3 ∈ R^{n×n×n}.   ▷ Compute C̄^g and Π̄ by tensor decomposition
        Compute whitening matrix Q̂ ∈ R^{n×n} such that Q̂^T M̂_2 Q̂ = I
        Compute eigenvalue-eigenvector pairs (α̂_k, v̂_k)_{k=1}^n of the whitened tensor M̂_3(Q̂, Q̂, Q̂) by using the robust tensor power method
        Compute ŵ_k = α̂_k^{-2} and μ̂_k = (Q̂^T)^{-1} α̂ v̂_k
        For k = 1, ..., n set the k-th column of C̄^c by some μ̂_k whose k-th coordinate has the greatest component, then set the k-th diagonal entry of Π̄ by ŵ_k
    end for
end for
```

probability of assigning any incorrect label. The following theorem gives the number of tasks each agent needs to complete to get an ε'-estimate of the aggregate confusion matrices.

We will use the following two lemmas due to Zhang et al. [2016].

Lemma 4.14. *For any $\hat{\varepsilon} \leqslant \sigma_L/2$, the second and the third empirical moments are bounded as*

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

with probability at least $1-\delta$ where $\delta = 6\exp\left(-(\sqrt{m_2}\hat{\varepsilon} - 1)^2\right) + n\exp\left(-(\sqrt{m_2/n}\hat{\varepsilon} - 1)^2\right)$

Lemma 4.15. *For any $\hat{\varepsilon} \leqslant \kappa/2$, if the empirical moments satisfy*

$$\max\{\|\hat{M}_2 - M_2\|_{op}, \|\hat{M}_3 - M_3\|_{op}\} \leqslant \hat{\varepsilon}H$$

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

then $\|\bar{C}^c - C\|_{op} \leqslant \sqrt{n}\hat{\varepsilon}$, $\|\bar{\Pi} - \Pi\|_{op} \leqslant \hat{\varepsilon}$ with probability at least $1-\delta$ where δ is defined in Lemma 4.14

Zhang et al. [2016] prove Lemma 4.14 when $\hat{M}_2$ is defined using the aggregate labels R_{gj}. However, this lemma holds even if one uses the labels $\tilde{R}_{gj}$. The proof is similar if one uses Lemma 4.13. We now characterize the sample complexity of learning the aggregate confusion matrices.

Theorem 4.16. *For any $\varepsilon' \leqslant \min\left\{\frac{31}{\sigma_L^2}, \frac{\kappa}{2}\right\} n^2$ and $\delta > 0$, if the size of the universe of shared tasks m_2 is at least $O\left(\frac{n^7}{(\varepsilon')^2\sigma_L^{11}}\log\left(\frac{nK}{\delta}\right)\right)$, then we have $\|\bar{C}^t - C^t\|_1 \leqslant \varepsilon'$ for each cluster G_t. The total number of samples collected by Algorithm 4 is $\tilde{O}(Km_2)$ under the uniform assignment scheme.*

Proof. Substituting $\hat{\varepsilon} = \hat{\varepsilon}_1 H\sigma_L^3/31$ in lemma 4.14 we get

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

with probability at least $1-(6+n)\exp\left(-\left(\frac{m_2^{1/2}\hat{\varepsilon}_1 H\sigma_L^3}{31n^{1/2}}-1\right)^2\right)$. This substitution requires $\hat{\varepsilon}_1 H\sigma_L^3/31 \leqslant \sigma_L/2$. Since $H \leqslant 1/2$, it is sufficient to have

$$\hat{\varepsilon}_1 \leqslant 31/\sigma_L^2 \tag{19}$$

Now using Lemma 4.15 we see that $\|\bar{C}^c - C\|_{op} \leqslant \sqrt{n}\hat{\varepsilon}_1$ and $\|\bar{\Pi} - \Pi\|_{op} \leqslant \hat{\varepsilon}_1$ with the above probability. It can be checked that $H \geqslant \frac{\sigma_L^{5/2}}{230n}$. This implies that the bounds hold with probability at least $1-(6+n)\exp\left(-\left(\frac{m_2^{1/2}\sigma_L^{11/2}\hat{\varepsilon}_1}{7130n^{3/2}}-1\right)^2\right)$. The second substitution requires

$$\hat{\varepsilon}_1 \leqslant \kappa/2 \tag{20}$$

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

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

It is sufficient that

$$m_2 \geqslant \Omega\left(\frac{n^3}{\hat{\varepsilon}_1^2\sigma_L^{11}}\log\left(\frac{n}{\delta}\right)\right)$$

to ensure $\|\bar{C}^c - C\|_{op} \leqslant \sqrt{n}\hat{\varepsilon}_1$. For each k, $\|\bar{C}_k^c - C_k\|_1 \leqslant \sqrt{n}\|\bar{C}_k^c - C_k\|_2 \leqslant \sqrt{n}\|\bar{C}^c - C\|_{op} \leqslant n\hat{\varepsilon}_1$. Substituting $\hat{\varepsilon}_1 = \hat{\varepsilon}'/n^2$, we get $\|\bar{C}^c - C\|_1 = \sum_{k=1}^n \|\bar{C}_k^c - C_k\|_1 \leqslant n^2\hat{\varepsilon}_1 = \hat{\varepsilon}'$ when $m_2 = \Omega\left(\frac{n^7}{(\hat{\varepsilon}')^2\sigma_L^{11}}\log\left(\frac{n}{\delta}\right)\right)$. By a union bound the result holds for all the clusters simultaneously with probability at least $1-\delta K$. Substituting δ/K instead of δ gives the bound on the number of samples. Substituting $\hat{\varepsilon}' = \hat{\varepsilon}_1/n^2$ in equations 19 and 20, we get the desired bound on $\hat{\varepsilon}'$.

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

Discussion. If the algorithm chooses $m_2 = \tilde{O}\left(\frac{n^7}{(\varepsilon')^2\sigma_L^{11}}\right)$, then the total number of samples collected under the uniform assignment scheme is at most $\tilde{O}\left(\frac{n^7}{(\varepsilon')^2\sigma_L^{11}}\right)$. So far we have analyzed the Dawid-Skene model under the uniform assignment scheme. When the assignment scheme is fixed, the moments of R_{aj} and $\tilde{R}_{aj}$ need not be the same. In this case we will have to run Algorithm 4 with respect to the actual aggregate labels $\{R_{gj}\}_{g=1}^K$. This requires collecting samples from every member of a cluster, leading to a sample complexity of $O\left(\frac{\ell n^7}{(\varepsilon')^2\sigma_L^{11}}\log\left(\frac{nK}{\delta}\right)\right)$

In order to estimate the confusion matrices, Zhang et al. [2016] require each agent to provide at least $O\left(n^5\log((\ell+n)/\delta)/(\varepsilon')^2\right)$ samples. Our algorithm requires $O\left(n^7\log(nK/\delta)/(\varepsilon')^2\right)$ samples from each cluster. The increase of n^2 in the sample complexity comes about because we are estimating the aggregate confusion matrices in L1 norm instead of the infinity norm. Moreover when the number of clusters is small ($K << \ell$), the number of samples required from each cluster does not grow with ℓ. This improvement is due to the fact that, unlike Zhang et al. [2016], we do not have to recover individual confusion matrices from the aggregate confusion matrices.

Note that the Dawid and Skene [1979b] based approach, for the uniform assignment scheme, does not require all agents to provide reports on the same set of shared tasks. Rather, we need that for each group of three clusters (as partitioned by Algorithm 4 on line 1) and each task, there exists one agent from the three clusters who completes the same task. In particular the reports for different tasks can be acquired from different agents within the same cluster. The assignment scheme makes sure that this property holds with high probability.

We now briefly compare the learning algorithms under the no-assumptions and model-based approach. When it is difficult to assign agents to the same tasks, and when the number of signals is small (which is often true in practice), the Dawid-Skene method has a strong advantage. Another advantage of the Dawid-Skene method is that the learning error ε' can be made arbitrarily small since each aggregate confusion matrix can be learned with arbitrary accuracy, whereas the true learning error of the no-assumption approach is at least 2ε (see Theorem 4.7), and depends on the problem instance.

5 CLUSTERING EXPERIMENTS

In this section, we study the clustering error on eight real-world, crowdsourcing datasets. Six of these datasets are from the *SQUARE* benchmark [Sheshadri and Lease, 2013], selected to ensure a sufficient density of worker labels across different latent attributes as well as the availability of latent attributes for sufficiently many tasks. In addition, we also use the Stanford *Dogs* dataset [Khosla et al., 2011] and the *Expressions* dataset [Mozafari et al., 2012, 2014]. Below, we briefly describe the format of tasks, the number of agents ℓ, and the number of signals n for each dataset.[4]

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

Recall that the maximum incentive an agent has to use a non-truthful strategy in the CAHU mechanism can be upper-bounded in terms of two sources of error:

- The clustering error. This represents how "clusterable" the agents are. From theory, we have the upper bound $\varepsilon_1 = \max_{p,q\in[\ell]} \|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1$.
- The learning error. This represents how accurate our estimates for the cluster Delta matrices are. From theory, we have the upper bound $\varepsilon_2 = \max_{i,j\in[K]} \|\Delta_{G_i,G_j} - \overline{\Delta}_{G_i,G_j}\|_1$.

Based on this, the CAHU mechanism is $(\varepsilon_1 + \varepsilon_2)$-informed truthful (Theorem 3.3) where ε_1 is the clustering error and ε_2 is the learning error. Even with the best clustering, the clustering error ε_1 cannot be made arbitrarily small because it depends on how close the signal distributions of the agents are as well as the number of clusters. In contrast, the learning error ε_2 of the no-assumption approach is $3\varepsilon' + 2\varepsilon$, (theorem 4.7) where the error due to ε' can indeed be made arbitrarily small by simply acquiring more data about agents' behavior. Similarly, the learning error ε_2 in the Dawid-Skene approach can be made arbitrarily small by acquiring more agent reports (theorem 4.16). Hence the total error is dominated by the clustering error ε_1.

For this reason, we focus on the clustering error, and show that it can be small even with a relatively small number of clusters. Rather than the use the weak bound $\max_{p,q\in[\ell]} \|\Delta_{p,q} -$

[4]We filter each dataset to remove tasks for which the latent attribute is unknown, and remove workers who only perform such tasks. ℓ is the number of agents that remain after filtering.

$\Delta_{G(p),G(q)}\|_1$ on the clustering error (which is nevertheless helpful for our theoretical results), we use the following tighter bound from the proof of Theorem 3.3.

$$|u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})| = \left|\frac{1}{(\ell-1)} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j)\left(\mathrm{Sign}(\Delta_{p,q})_{i,j} - \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})_{i,j}\right)\right| \tag{21}$$

The datasets specify the latent value of each task. Because of this, we can adopt the Dawid-Skene model, and estimate the confusion matrices from the frequency which with each agent p reports each label j in the case of each latent attribute i.

Typical clustering algorithms take a distance metric over the space of data points and attempt to minimize the maximum cluster diameter, which is the maximum distance between any two points within a cluster. In contrast, our objective (the tighter bound on the incentive in Equation (5)) is a complex function of the underlying confusion matrices. We therefore compare two approaches:

1) We cluster the confusion matrices using the standard k-means++ algorithm with the $L2$ norm distance (available in Matlab), and hope that resulting clustering leads to a small incentive bound.[5]
2) In the following lemma, we derive a distance metric between confusion matrices for which the maximum cluster diameter is provably an upper bound on the incentive, and use k-means++ with this metric (implemented in Matlab).[6] Note that computing this metric requires knowledge of the prior over the latent attribute.

Lemma 5.1. *For all agents p, q, r, we have $\|\Delta_{p,q} - \Delta_{p,r}\|_1 \leqslant 2 \cdot \sum_k \pi_k \sum_j |C^q_{kj} - C^r_{kj}|$.*

Proof. We have

$$\begin{aligned}
\|\Delta_{p,q} - \Delta_{p,r}\|_1 &= \sum_{i,j} \left|\Delta_{p,q}(i,j) - \Delta_{p,r}(i,j)\right| \\
&= \sum_{i,j} \left|D_{p,q}(i,j) - D_p(i)D_q(j) - D_{p,r}(i,j) + D_p(i)D_r(j)\right| \\
&= \sum_{i,j} \left|D_{p,q}(i,j) - D_{p,r}(i,j) - D_p(i)(D_q(j) - D_r(j))\right| \\
&= \sum_{i,j} \left|\sum_k \pi_k C^p_{ki} C^q_{kj} - \sum_k \pi_k C^p_{ki} C^r_{kj} - \sum_k \pi_k C^p_{ki}\left(\sum_l \pi_l C^q_{lj} - \sum_l \pi_l C^r_{lj}\right)\right| \\
&= \sum_{i,j} \left|\sum_k \pi_k C^p_{ki}\left(C^q_{kj} - C^r_{kj}\right) - \sum_k \pi_k C^p_{ki}\left(\sum_l \pi_l \left(C^q_{lj} - C^r_{lj}\right)\right)\right|
\end{aligned}$$

[5] We use $L2$ norm rather than $L1$ norm because the standard k-means++ implementation uses as the centroid of a cluster the confusion matrix that minimizes the sum of distances from the confusion matrices of the agents in the cluster. For $L2$ norm, this amounts to averaging over the confusion matrices, which is precisely what we want. For $L1$ norm, this amounts to taking a pointwise median, which does not even result in a valid confusion matrix. Perhaps for this reason, we observe that using the $L1$ norm performs worse.

[6] For computing the centroid of a cluster, we still average over the confusion matrices of the agents in the cluster. Also, since the algorithm is no longer guaranteed to converge (indeed, we observe cycles), we restart the algorithm when a cycle is detected, at most 10 times.

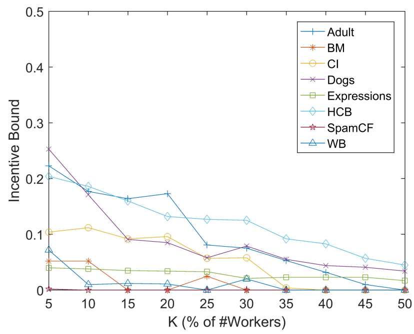

Fig. 4. Incentive bound on each of the 8 different data sets when using k-means++ with the $L2$ norm

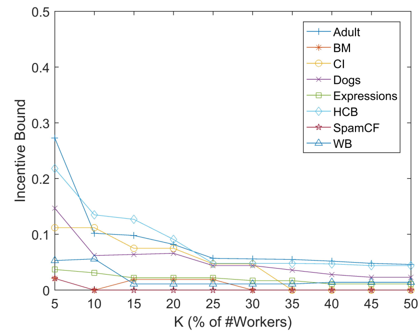

Fig. 5. Incentive bound on each of the 8 different data sets when using k-means++ with our custom metric

$$
\begin{aligned}
&\leqslant \sum_j \sum_k \pi_k \left|C^q_{kj} - C^r_{kj}\right| \sum_i C^p_{ki} + \sum_j \sum_k \pi_k \sum_l \pi_l \left|C^q_{lj} - C^r_{lj}\right| \sum_i C^p_{ki} \\
&= \sum_j \sum_k \pi_k \left|C^q_{kj} - C^r_{kj}\right| + \sum_j \sum_k \pi_k \sum_l \pi_l \left|C^q_{lj} - C^r_{lj}\right| \qquad [\text{Using } \sum_i C^p_{ki} = 1] \\
&\leqslant \sum_k \pi_k \sum_j \left|C^q_{kj} - C^r_{kj}\right| + \sum_k \pi_k \sum_l \pi_l \sum_j \left|C^q_{lj} - C^r_{lj}\right| \\
&= \sum_k \pi_k \sum_j \left|C^q_{kj} - C^r_{kj}\right| + \sum_l \pi_l \sum_j \left|C^q_{lj} - C^r_{lj}\right| \quad [\text{Using } \sum_k \pi_k = 1] \\
&= 2 \cdot \sum_k \pi_k \sum_j \left|C^q_{kj} - C^r_{kj}\right|,
\end{aligned}
$$

as required. □

Note that $\sum_k \pi_k \sum_j |C^q_{kj} - C^r_{kj}| \leqslant \|C^q - C^r\|_1$ because $\sum_j \left|C^q_{lj} - C^r_{lj}\right| \leqslant \|C^q - C^r\|_1$. Lemma 5.1, along with Lemma 4.8, shows that the incentive is upper bounded by four times the maximum cluster diameter under our metric. For each dataset, we vary the number of clusters K from 5% to 15% of the number of agents in the dataset. We repeat the experiment 20 times, and select the clustering that produces the smallest incentive bound. Figures 4 and 5 show the incentive bound achieved using the standard $L2$ metric and using our custom metric, respectively. We see that the incentive bound is small compared to the maximum payment of 1 by CAHU, even with the number of clusters K as small as 15% of the number of workers. The number of agents does not seem to affect this bound as long as the number of clusters is small relative to the number of agents. Using our custom metric leads to a clustering with a noticeably better incentive bound.

6 CONCLUSION

We have provided the first, general solution to the problem of peer prediction with heterogeneous agents. This is a compelling research direction, where new theory and algorithms can help to guide practice. In particular, heterogeneity is likely to be quite ubiquitous due to differences in taste, context, judgment, and reliability across users. Beyond testing these methods in a real-world application such as marketing surveys, there remain interesting directions for ongoing research. For example, is it possible to solve this problem with a similar sample complexity but without a clustering approach? Is it possible to couple methods of peer prediction with optimal methods for inference in crowdsourced classification [Ok et al., 2016], and with methods for task assignment in budgeted settings [Karger et al., 2014]? This should include attention to adaptive assignment schemes [Khetan and Oh, 2016] that leverage generalized Dawid-Skene models [Zhou et al., 2015], and could connect to the recent progress on task heterogeneity within peer prediction [Mandal et al., 2016].

REFERENCES

Hunt Allcott and Matthew Gentzkow. 2017. *Social Media and Fake News in the 2016 Election.* Technical Report. National Bureau of Economic Research.

Animashree Anandkumar, Rong Ge, Daniel J Hsu, Sham M Kakade, and Matus Telgarsky. 2014. Tensor Decompositions for Learning Latent Variable Models. *Journal of Machine Learning Research* 15, 1 (2014), 2773–2832.

Yang Cai, Constantinos Daskalakis, and Christos Papadimitriou. 2015. Optimum Statistical Estimation with Strategic Data Sources. In *Proceedings of The 28th Conference on Learning Theory.* 280–296.

Anirban Dasgupta and Arpita Ghosh. 2013. Crowdsourced Judgement Elicitation with Endogenous Proficiency. In *Proceedings of the 22nd international conference on World Wide Web.* ACM, 319–330.

Alexander Philip Dawid and Allan M Skene. 1979a. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied statistics* (1979), 20–28.

Philip A Dawid and Allan M Skene. 1979b. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied statistics* 28 (1979), 20–28.

Jennifer DeBoer, Glenda S Stump, Daniel Seaton, and Lori Breslow. 2013. Diversity in MOOC Students' Backgrounds and Behaviors in Relationship to Performance in 6.002 x. In *Proceedings of the Sixth Learning International Networks Consortium Conference*, Vol. 4.

Luc Devroye and Gábor Lugosi. 2012. *Combinatorial Methods in Density Estimation.* Springer Science & Business Media.

Adam Fourney, Miklos Z Racz, Gireeja Ranade, Markus Mobius, and Eric Horvitz. 2017. Geographic and Temporal Trends in Fake News Consumption During the 2016 US Presidential Election. (2017).

Rafael Frongillo and Jens Witkowski. 2017. A Geometric Perspective on Minimal Peer Prediction. *ACM Transactions on Economics and Computation (TEAC)* (2017).

Alice Gao, James R Wright, and Kevin Leyton-Brown. 2016. Incentivizing Evaluation via Limited Access to Ground Truth: Peer-Prediction Makes Things Worse. *EC 2016 Workshop on Algorithmic Game Theory and Data Science* (2016).

Radu Jurca, Boi Faltings, et al. 2009. Mechanisms for Making Crowds Truthful. *Journal of Artificial Intelligence Research* 34, 1 (2009), 209.

Vijay Kamble, David Marn, Nihar Shah, Abhay Parekh, and Kannan Ramachandran. 2015. Truth Serums for Massively Crowdsourced Evaluation Tasks. *The 5th Workshop on Social Computing and User-Generated Content* (2015).

David R. Karger, Sewoong Oh, and Devavrat Shah. 2011. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. *CoRR* abs/1110.3564 (2011). http://arxiv.org/abs/1110.3564

David R. Karger, Sewoong Oh, and Devavrat Shah. 2014. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. *Operations Research* 62, 1 (2014), 1–24. https://doi.org/10.1287/opre.2013.1235

Ashish Khetan and Sewoong Oh. 2016. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *Annual Conference on Neural Information Processing Systems.* 4844–4852.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. 2011. Novel Dataset for Fine-Grained Image Categorization. In *First CVPR Workshop on Fine-Grained Visual Categorization.*

Yuqing Kong, Katrina Ligett, and Grant Schoenebeck. 2016. Putting Peer Prediction Under the Micro (economic) scope and Making Truth-telling Focal. In *International Conference on Web and Internet Economics.* Springer, 251–264.

Yuqing Kong and Grant Schoenebeck. 2016. A Framework For Designing Information Elicitation Mechanism That Rewards Truth-telling. (2016). http://arxiv.org/abs/1605.01021

Chinmay Kulkarni, Koh Pang Wei, Huy Le, Daniel Chia, Kathryn Papadopoulos, Justin Cheng, Daphne Koller, and Scott R Klemmer. 2015. Peer and Self Assessment in Massive Online Classes. In *Design thinking research.* Springer, 131–168.

Yang Liu and Yiling Chen. 2017a. Machine-Learning Aided Peer Prediction. In *Proceedings of the 2017 ACM Conference on Economics and Computation.* ACM, 63–80.

Yang Liu and Yiling Chen. 2017b. Sequential Peer Prediction: Learning to Elicit Effort Using Posted Prices. In *Thirty-First AAAI Conference on Artificial Intelligence.* 607–613.

Debmalya Mandal, Matthew Leifer, David C Parkes, Galen Pickard, and Victor Shnayder. 2016. Peer Prediction with Heterogeneous Tasks. *NIPS 2016 Workshop on Crowdsourcing and Machine Learning* (2016).

Nolan Miller, Paul Resnick, and Richard Zeckhauser. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science* 51 (2005), 1359–1373.

Barzan Mozafari, Purnamrita Sarkar, Michael J. Franklin, Michael I. Jordan, and Samuel Madden. 2012. Active Learning for Crowd-Sourced Databases. *CoRR* abs/1209.3686 (2012).

Barzan Mozafari, Purnamrita Sarkar, Michael J. Franklin, Michael I. Jordan, and Samuel Madden. 2014. Scaling Up Crowd-Sourcing to Very Large Datasets: A Case for Active Learning. *PVLDB* 8, 2 (2014), 125–136.

Jungseul Ok, Sewoong Oh, Jinwoo Shin, and Yung Yi. 2016. Optimality of Belief Propagation for Crowdsourced Classification. In *Proc. 33nd Int. Conf. on Machine Learning (ICML)*. 535–544.

Drazen Prelec. 2004. A Bayesian Truth Serum For Subjective Data. *Science* 306, 5695 (2004), 462.

Goran Radanovic and Boi Faltings. 2015a. Incentive Schemes for Participatory Sensing. In *Proc. Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS*. 1081–1089.

Goran Radanovic and Boi Faltings. 2015b. Incentive Schemes for Participatory Sensing. In *AAMAS 2015*.

Goran Radanovic and Boi Faltings. 2015c. Incentives for Subjective Evaluations with Private Beliefs. In *Proc. 29th AAAI Conf. on Art. Intell. (AAAI'15)*. 1014–1020.

Goran Radanovic, Boi Faltings, and Radu Jurca. 2016. Incentives for effort in crowdsourcing using the peer truth serum. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7, 4 (2016), 48.

Aashish Sheshadri and Matthew Lease. 2013. SQUARE: A Benchmark for Research on Computing Crowd Consensus. In *Proc. 1st AAAI Conf. on Human Computation (HCOMP)*. 156–164.

Victor Shnayder, Arpit Agarwal, Rafael Frongillo, and David C Parkes. 2016. Informed Truthfulness in Multi-Task Peer Prediction. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. ACM, 179–196.

Victor Shnayder, Rafael Frongillo, and David C. Parkes. 2016. Measuring Performance Of Peer Prediction Mechanisms Using Replicator Dynamics. In *Proc. 25th Int. Joint Conf. on Art. Intell. (IJCAI'16)*. 2611–2617.

Victor Shnayder and David C Parkes. 2016. Practical Peer Prediction for Peer Assessment. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*.

Edwin Simpson, Stephen J Roberts, Ioannis Psorakis, and Arfon Smith. 2013. Dynamic Bayesian Combination of Multiple Imperfect Classifiers. *Decision Making and Imperfection* 474 (2013), 1–35.

Luis von Ahn and Laura Dabbish. 2004. Labeling Images with a Computer Game. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI'04)*. 319–326.

Julia Wilkowski, Amit Deutsch, and Daniel M Russell. 2014. Student Skill and Goal Achievement in the Mapping with Google MOOC. In *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 3–10.

Jens Witkowski, Yoram Bachrach, Peter Key, and David Parkes. 2013. Dwelling on the negative: Incentivizing effort in peer prediction. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Jens Witkowski and David C Parkes. 2013. Learning the Prior in Minimal Peer Prediction. In *Proceedings of the 3rd Workshop on Social Computing and User Generated Content at the ACM Conference on Electronic Commerce*. 14.

Muhammad Bilal Zafar, Krishna P Gummadi, and Cristian Danescu-Niculescu-Mizil. 2016. Message Impartiality in Social Media Discussions. In *ICWSM*. 466–475.

Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. 2016. Spectral Methods Meet EM: A Provably Optimal Algorithm for Crowdsourcing. *Journal of Machine Learning Research* 17, 102 (2016), 1–44.

Dengyong Zhou, Qiang Liu, John C. Platt, Christopher Meek, and Nihar B. Shah. 2015. Regularized Minimax Conditional Entropy for Crowdsourcing. *CoRR* abs/1503.07240 (2015). http://arxiv.org/abs/1503.07240